```
{Application 'RECODE1' logic file generated by CSPro}
{##########################################################################################
###################
2015-16 Tokelau HIES Data Editing CSPro Logic File

File Input: ..\Data\Tokelau HIES data (raw).dat

File Output: ..\Data\Tokelau HIES data (Run 0 to 1).dat

Last Compiled date: 29 September 2016

Pierre Wong
Data Processing Specialist
Statistics for Development Division, Pacific Community (SDD, SPC)
##########################################################################################
##################}
PROC GLOBAL
numeric day1day;
numeric incDay;
numeric currentHH;
numeric i, x, y, z;
numeric hh_ok;
numeric isValid;
numeric hhId;

array hd_20144_fr_tenure(TENURE_VS) save;
array hd_bMonth_fr_sex_age(hd_sex_vs, hd_age_vs) save;
array hd_bDay_fr_sex_age(hd_sex_vs, hd_age_vs) save;
array hd_birthPlace_fr_sex(hd_sex_vs) save;
array hd_citizenship_fr_sex(hd_sex_vs) save;
array hd_20356_fr_tenure_expCode(TENURE_VS, EXPCODE203_VS) save;
array hd_20906_fr_tenure_expCode(TENURE_VS, Q20903_VS1) save;
// we have to convert this to a standard HD because we are refrencing two differnet records
array hd_30602_fr_sex_relate_age(2, 10, 10) save;
array hd_30603_fr_sex_age_drink(2, 10, 10) save;
array hd_30604_fr_sex_age_drink(2, 10, 2) save;
array hd_30605_fr_sex_age_drink(2, 10, 2) save;
array hd_40106_fr_occ(ISCO_VS) save;
array hd_40307_fr_tenure(TENURE_VS) save;
array hd_21006_fr_tenure(Q21003_VS1) save;

array hd_40202_fr_tenure(TENURE2_VS) save;
array hd_40203_fr_tenure(TENURE2_VS) save;
array hd_40204_fr_tenure(TENURE2_VS) save;
array hd_40205_fr_tenure(TENURE2_VS) save;
array hd_40206_fr_tenure(TENURE2_VS) save;
array hd_40207_fr_tenure(TENURE2_VS) save;

array hd_40707_fr_tenure(TENURE2_VS) save;

array hd_10306_sex_age(SEX_VS, AGE_VS) save;
array hd_10307_sex_age(SEX_VS, AGE_VS) save;

array hd_10501_sex_age(SEX_VS, AGE_VS2) save;
array hd_10502_sex_age(SEX_VS, AGE_VS2) save;
array hd_10503_sex_age(SEX_VS, AGE_VS2) save;
array hd_10504_sex_age(SEX_VS, AGE_VS2) save;

array hd_20146(LIGHT, WALLS) save;


alias
    sex : Q10103,
    age : Q10104,
    relate : Q10105,
    rcdSex : psn_Q10103,
    rcdAge  : psn_Q10104,
    rcdRelate : psn_Q10105,
    IntYearMD1 : CV18C,
    IntYearMD2 : CV20C,
    IntYearMD3 : CV22C,
```

```
    IntYearMD4 : CV24C,
    hhSize : CV17;

// 1)sort_index 2)island    3)island_code   4)village   5)village_code  6) household_head
7)census_link   8)replacement   9)round 10) seq_num
array string sampleFrame(180,10) =
    "1" "Atafu" "1" "Atafu" "11"    "***************"    "AT4007"    "A" "1" "101"
    "2" "Atafu" "1" "Atafu" "11"    "***************"    "AT4044"    "A" "1" "102"
    "3" "Atafu" "1" "Atafu" "11"    "***************"    "AT4068"    "A" "1" "103"
    { rest removed for confidentiality}
    "177"   "Nukunonu"  "3" "Fale"  "33"    "***************"    "NU3019"    "B" "4" "357"
    "178"   "Nukunonu"  "3" "Alatai"    "31"    "***************"    "NU3049"    "B" "4" "358"
    "179"   "Nukunonu"  "3" "Motuhaga"  "34"    "***************"    "NU3014"    "B" "4" "359"
    "180"   "Nukunonu"  "3" "other village" "36"    "***************"    "NU3074"    "B" "4"
    "360"
;


// 1)sort_index 2)COICOP    3)Choice    4)1 diary   5)2 Diaries 6)module
array numeric Selection(316,6) =
// this is used for item selection e.g., item can only have one source
// eitehr from the diary or module.

1   21107602    3   17.38095238 26.07142857 0
2   21107604    3   17.38095238 26.07142857 0
3   21107605    3   17.38095238 26.07142857 0
4   21107608    3   17.38095238 26.07142857 0
5   21107609    3   17.38095238 26.07142857 0
6   21107611    3   17.38095238 26.07142857 0
7   21207801    3   17.38095238 26.07142857 0
8   21207803    3   17.38095238 26.07142857 0
9   21207806    3   17.38095238 26.07142857 0
10  21308001    3   17.38095238 26.07142857 0
11  21308003    3   17.38095238 26.07142857 0
12  21308004    3   17.38095238 26.07142857 0
13  22108201    3   17.38095238 26.07142857 0
14  22108202    3   17.38095238 26.07142857 0
15  22108301    3   17.38095238 26.07142857 0
16  31109202    2   0   0   4.010989011
17  31209301    2   0   0   4.010989011
18  31209302    2   0   0   4.010989011
19  31209303    2   0   0   4.010989011
20  31209304    2   0   0   4.010989011
21  31209305    2   0   0   4.010989011
22  31209306    2   0   0   1
23  31209307    2   0   0   4.010989011
24  31209309    2   0   0   4.010989011
25  31209310    2   0   0   4.010989011
26  31209401    2   0   0   4.010989011
27  31209402    2   0   0   4.010989011
28  31209404    2   0   0   4.010989011
29  31209406    2   0   0   4.010989011
30  31209407    2   0   0   1
31  31209408    2   0   0   4.010989011
32  31209410    2   0   0   4.010989011
33  31209411    2   0   0   4.010989011
34  31209412    2   0   0   4.010989011
35  31209501    2   0   0   4.010989011
36  31209610    2   0   0   4.010989011
37  31309605    2   0   0   4.010989011
38  31409901    2   0   0   4.010989011
39  32110201    3   3.476190476 3.724489796 0
40  32110203    2   0   0   4.010989011
41  32110301    2   0   0   4.010989011
42  32110302    2   0   0   4.010989011
43  32110304    2   0   0   4.010989011
44  32110401    2   0   0   4.010989011
45  42111201    2   0   0   12.00657895
46  42211401    2   0   0   12.00657895
47  43111501    2   0   0   1
48  43111504    2   0   0   1
```

49	43111505	2	0	0	1
50	43111601	2	0	0	1
51	43111603	2	0	0	1
52	43111702	2	0	0	1
53	43111703	2	0	0	1
54	43111704	2	0	0	1
55	43111801	2	0	0	1
56	43111802	2	0	0	1
57	43111804	2	0	0	1
58	43111805	2	0	0	1
59	43111806	2	0	0	1
60	43111807	2	0	0	1
61	43111808	2	0	0	1
62	43111809	2	0	0	1
63	43111810	2	0	0	1
64	43111811	2	0	0	1
65	43111812	2	0	0	1
66	43111813	2	0	0	1
67	43111815	2	0	0	1
68	43111817	2	0	0	1
69	43111821	2	0	0	1
70	43211901	2	0	0	1
71	45112501	2	0	0	12.00657895
72	45212702	2	0	0	1
73	45312801	2	0	0	12.00657895
74	45312802	2	0	0	12.00657895
75	45312803	2	0	0	12.00657895
76	45412901	2	0	0	12.00657895
77	45513001	1	26.07142857	52.14285714	0
78	51113101	2	0	0	1
79	51113102	2	0	0	1
80	51113201	2	0	0	1
81	51113203	2	0	0	1
82	51113301	2	0	0	1
83	51113302	2	0	0	1
84	51113303	2	0	0	1
85	51113305	2	0	0	1
86	51113501	2	0	0	1
87	51113802	2	0	0	1
88	51214001	2	0	0	1
89	51214003	2	0	0	1
90	52114202	2	0	0	1
91	52114304	2	0	0	1
92	52114401	2	0	0	1
93	52114403	2	0	0	1
94	52114404	2	0	0	1
95	52114407	2	0	0	1
96	52114408	2	0	0	1
97	52114501	2	0	0	1
98	52114602	2	0	0	1
99	52114603	2	0	0	1
100	52114604	2	0	0	1
101	52114701	1	26.07142857	52.14285714	0
102	52114708	2	0	0	1
103	53114901	2	0	0	1
104	53114904	2	0	0	1
105	53114905	2	0	0	1
106	53114906	2	0	0	1
107	53114908	2	0	0	1
108	53114909	2	0	0	1
109	53114910	2	0	0	1
110	53114911	2	0	0	1
111	53115002	2	0	0	1
112	53115106	2	0	0	1
113	53115109	2	0	0	1
114	53115111	2	0	0	1
115	53215201	2	0	0	1
116	53215202	2	0	0	1
116	53215204	2	0	0	1
117	53215205	2	0	0	1
118	53215206	2	0	0	1

119	53215213	2	0	0	1
120	53215215	2	0	0	1
121	54015702	3	0.963060686	0.981182796	0
122	54015718	3	0.963060686	0.981182796	0
123	54015719	2	0	0	1
124	55115902	2	0	0	1
125	55115907	2	0	0	1
126	55115909	2	0	0	1
127	55115911	2	0	0	1
128	55115912	2	0	0	1
129	55115914	2	0	0	1
130	55216121	2	0	0	1
131	55216124	2	0	0	1
132	55216128	3	0.963060686	0.981182796	0
133	55216201	2	0	0	1
134	55216211	2	0	0	1
135	55216401	2	0	0	1
136	55216408	2	0	0	1
137	55216411	2	0	0	1
138	55216605	2	0	0	1
139	55216607	2	0	0	1
140	55216701	1	26.07142857	52.14285714	0
141	55216704	1	26.07142857	52.14285714	0
142	55216705	1	26.07142857	52.14285714	0
143	55216710	2	0	0	1
144	55216711	2	0	0	1
145	55216713	2	0	0	1
146	55216722	2	0	0	1
147	55216723	3	0.963060686	0.981182796	0
148	55216724	2	0	0	1
149	55216725	2	0	0	1
150	56116902	1	26.07142857	52.14285714	0
151	56116903	1	26.07142857	52.14285714	0
152	56116904	1	26.07142857	52.14285714	0
153	56116905	1	26.07142857	52.14285714	0
154	56116906	1	26.07142857	52.14285714	0
155	56116907	1	26.07142857	52.14285714	0
156	56116911	1	26.07142857	52.14285714	0
157	56116912	1	26.07142857	52.14285714	0
158	56116914	1	26.07142857	52.14285714	0
159	56116915	1	26.07142857	52.14285714	0
160	56116917	1	26.07142857	52.14285714	0
161	56116918	1	26.07142857	52.14285714	0
162	56116919	1	26.07142857	52.14285714	0
163	56116920	1	26.07142857	52.14285714	0
164	56116921	1	26.07142857	52.14285714	0
165	56116923	1	26.07142857	52.14285714	0
166	56116924	1	26.07142857	52.14285714	0
167	56116925	1	26.07142857	52.14285714	0
168	56116926	1	26.07142857	52.14285714	0
169	56116928	1	26.07142857	52.14285714	0
170	56117001	1	26.07142857	52.14285714	0
171	56117003	1	26.07142857	52.14285714	0
172	56117010	1	26.07142857	52.14285714	0
173	56117101	1	26.07142857	52.14285714	0
174	56117102	1	26.07142857	52.14285714	0
175	56117103	1	26.07142857	52.14285714	0
176	56117105	1	26.07142857	52.14285714	0
177	56117106	1	26.07142857	52.14285714	0
178	56117108	3	0.963060686	0.981182796	0
179	56117202	1	26.07142857	52.14285714	0
180	56117215	1	26.07142857	52.14285714	0
181	56117224	2	0	0	1
182	56117225	1	26.07142857	52.14285714	0
183	56117226	1	26.07142857	52.14285714	0
184	56117227	1	26.07142857	52.14285714	0
185	56117228	1	26.07142857	52.14285714	0
186	56217304	2	0	0	1
187	56217306	2	0	0	1
188	56217309	2	0	0	1
189	56217398	2	0	0	1

190	61117801	2	0	0	1
191	61117901	3	0.963060686	0.981182796	0
192	62118401	2	0	0	1
193	62218501	2	0	0	1
194	62318601	2	0	0	1
195	62318603	2	0	0	1
196	63018901	2	0	0	1
197	71119101	2	0	0	1
198	71219202	2	0	0	1
199	71319301	2	0	0	1
200	72119503	2	0	0	1
201	72119507	2	0	0	1
202	72219601	3	8.220720721	9.759358289	0
203	72219602	3	8.220720721	9.759358289	0
204	72219603	3	8.220720721	9.759358289	0
205	72219705	2	0	0	1
206	72219708	3	0.963060686	0.981182796	0
207	72219709	3	0.963060686	0.981182796	0
208	72319801	2	0	0	1
209	72419902	2	0	0	1
210	72420001	3	0.963060686	0.981182796	0
211	73220102	2	0	0	1
212	73320201	2	0	0	1
213	73420301	2	0	0	1
214	73620501	2	0	0	1
215	82020701	2	0	0	1
216	83020902	2	0	0	12.00657895
217	83020904	2	0	0	1
218	83020905	2	0	0	12.00657895
219	83020906	2	0	0	12.00657895
220	91121005	2	0	0	1
221	91121006	2	0	0	1
222	91121008	2	0	0	1
223	91121010	2	0	0	1
224	91121011	2	0	0	1
225	91221101	2	0	0	1
226	91321505	2	0	0	1
227	91321507	2	0	0	1
228	91421604	2	0	0	1
229	92121801	2	0	0	1
230	92121803	2	0	0	1
231	93122204	2	0	0	1
232	93122205	2	0	0	1
233	93222804	1	26.07142857	52.14285714	0
234	93223004	2	0	0	1
235	94224701	2	0	0	12.00657895
236	95125008	2	0	0	1
237	95425414	3	0.963060686	0.981182796	0
238	95425416	2	0	0	1
239	95425601	2	0	0	1
240	101025801	2	0	0	1
241	101025802	2	0	0	1
242	102025901	2	0	0	1
243	104026101	2	0	0	1
244	105026203	2	0	0	1
245	111126301	2	0	0	1
246	111126302	2	0	0	1
247	111126402	1	26.07142857	52.14285714	0
248	111126403	2	0	0	1
249	112026801	2	0	0	1
250	112026802	2	0	0	1
251	112026901	2	0	0	1
252	121127101	2	0	0	4.010989011
253	121127208	2	0	0	1
254	121327504	1	26.07142857	52.14285714	0
255	121327601	1	26.07142857	52.14285714	0
256	121327602	1	26.07142857	52.14285714	0
257	121327603	1	26.07142857	52.14285714	0
258	121327604	1	26.07142857	52.14285714	0
259	121327606	1	26.07142857	52.14285714	0
260	121327607	1	26.07142857	52.14285714	0

```
261 121327608   1   26.07142857 52.14285714 0
262 121327610   1   26.07142857 52.14285714 0
263 121327611   1   26.07142857 52.14285714 0
264 121327612   1   26.07142857 52.14285714 0
265 121327613   1   26.07142857 52.14285714 0
266 121327615   1   26.07142857 52.14285714 0
267 121327701   1   26.07142857 52.14285714 0
268 121327702   1   26.07142857 52.14285714 0
269 121327703   1   26.07142857 52.14285714 0
270 121327704   1   26.07142857 52.14285714 0
271 121327705   1   26.07142857 52.14285714 0
272 121327707   1   26.07142857 52.14285714 0
273 121327708   1   26.07142857 52.14285714 0
274 121327709   1   26.07142857 52.14285714 0
275 121327710   1   26.07142857 52.14285714 0
276 121327711   1   26.07142857 52.14285714 0
277 121327714   1   26.07142857 52.14285714 0
278 121327716   2   0   0   4.010989011
279 121327801   1   26.07142857 52.14285714 0
280 121327804   1   26.07142857 52.14285714 0
281 121327805   1   26.07142857 52.14285714 0
282 121327806   1   26.07142857 52.14285714 0
283 121327807   1   26.07142857 52.14285714 0
284 121327808   1   26.07142857 52.14285714 0
285 121327809   1   26.07142857 52.14285714 0
286 121327810   1   26.07142857 52.14285714 0
287 121327811   1   26.07142857 52.14285714 0
288 121327813   3   0.963060686 0.981182796 0
289 123128101   2   0   0   4.010989011
290 123128202   2   0   0   4.010989011
291 123228601   1   26.07142857 52.14285714 0
292 125129201   2   0   0   1
293 125229301   2   0   0   1
294 125329401   2   0   0   1
295 126229801   2   0   0   1
296 126229802   2   0   0   1
297 126229803   2   0   0   1
298 127029901   2   0   0   1
299 127029905   2   0   0   1
300 901100101   2   0   0   1
301 901100102   2   0   0   1
302 901100103   2   0   0   1
303 901100104   2   0   0   1
304 901100105   2   0   0   1
305 901100106   2   0   0   1
306 901100108   2   0   0   1
307 901100109   2   0   0   1
308 901100110   2   0   0   1
309 901100301   2   0   0   1
310 901100302   2   0   0   1
311 901100601   2   0   0   1
312 901100602   2   0   0   1
313 901100603   2   0   0   1
314 901100704   2   0   0   1
315 901100706   2   0   0   1
;

// function to convert letters to day number
function getDay(string name)
    getDay = 0;
    if poschar("A",name) > 0 then getDay = 1; endif;
    if poschar("B",name) > 0 then getDay = 2; endif;
    if poschar("C",name) > 0 then getDay = 3; endif;
    if poschar("D",name) > 0 then getDay = 4; endif;
    if poschar("E",name) > 0 then getDay = 5; endif;
    if poschar("F",name) > 0 then getDay = 6; endif;
    if poschar("G",name) > 0 then getDay = 7; endif;
    if poschar("H",name) > 0 then getDay = 8; endif;
    if poschar("I",name) > 0 then getDay = 9; endif;
    if poschar("J",name) > 0 then getDay = 10; endif;
    if poschar("K",name) > 0 then getDay = 11; endif;
```

```
    if poschar("L",name) > 0 then getDay = 12; endif;
    if poschar("M",name) > 0 then getDay = 13; endif;
    if poschar("N",name) > 0 then getDay = 14; endif;
end;

// function to autocode COICOP based on the hies code
function toCOICOP(tmp)
    numeric t1;
    recode  tmp  => t1  ;
        402.501  => 904102301   ; //    Business - salary to staff
        402.502  => 904102302   ; //    Business - goods for resale
        402.503  => 904102303   ; //    Business - electricity
        402.504  => 904102304   ; //    Business - Water
        402.505  => 904102305   ; //    Business - Communication
        402.506  => 904102306   ; //    Business - Fuel
        402.507  => 904102307   ; //    Buiness - Raw materials
        402.508  => 904102308   ; //    Business - repair & maintenance
        402.509  => 904102309   ; //    Business - rental of building
        402.510  => 904102310   ; //    Business - rental of equipment
        402.511  => 904102311   ; //    Business - Registration license
        {I}402.512   => 904102312   ; //    Business - equipment (Investments)
        402.513  => 904102313   ; //    Business - other

        403.901  => 904103301   ; //    Agriculture - transport
        403.902  => 904103302   ; //    Agriculture - labor
        {I}403.903   => 904103303   ; //    Agriculture - Purchase of equipment (Investments)
        403.904  => 904103304   ; //    Agriculture - Rental of equipment
        403.905  => 904103305   ; //    Agriculture - Other (seeds, tools, fertilizer…)

        404.901  => 904104301   ; //    Handicraft&food - ingredient for making food
        404.902  => 904104302   ; //    Handicraft&food - material for making handicraft
        404.903  => 904104303   ; //    Handicraft&food - transport
        404.904  => 904104304   ; //    Handicraft&food - labor
        404.905  => 904104305   ; //    Handicraft&food - other

        405.901  => 904105301   ; //    Livestock - feed
        {I}405.902   => 904105302   ; //    Livestock - fencing and enclosure (Investments)
        405.903  => 904105303   ; //    Livestock - transport of animals
        {I}405.904   => 904105304   ; //    Livestock -Purcahse of animal  (Investments)
        405.905  => 904105305   ; //    Livestock - labor
        405.906  => 904105306   ; //    Livestock - veterinary
        405.907  => 904105307   ; //    Livestock - other

        406.901  => 904106301   ; //    Fishing - fuel and oil
        406.902  => 904106302   ; //    Fishing - maintenance and repair
        {I}406.903   => 904106303   ; //    Fishing - purchase of equipment  (Investments)
        406.904  => 904106304   ; //    Fishing - transport of catch
        406.905  => 904106305   ; //    Fishing - labor
        406.906  => 904106306   ; //    Fishing - ice
                 => notappl;
    endrecode;
    toCOICOP = t1;
end;

// This function is used to populate the imputation flags (numeric variables)
function setNumFlag(string name, occNum, impType, newVal)
    numeric tmp;
    numeric tmp2;
    tmp = noccurs(FLAGS)+1;
    insert(FLAGS_EDT(tmp));
    flag_quest_name(tmp) = name;
    flag_occ_num(tmp) = occNum;
    flag_imputation_type(tmp) = impType;
    if occNum = 0 then
        if getvaluenumeric(name) in notappl, default then
            flag_original_value(tmp) = "MISSING";
        else
            flag_original_value(tmp) = maketext("%d",getvaluenumeric(name));
        endif;
    else
        if getvaluenumeric(name,occNum) in notappl, default then
```

```
            flag_original_value(tmp) = "MISSING";
        else
            flag_original_value(tmp) = maketext("%d",getvaluenumeric(name,occNum));
        endif;
    endif;
    if newVal in notappl, default then
        flag_new_value(tmp) = "N.A.";
    else
        flag_new_value(tmp) = maketext("%d",newVal);
    endif;
    setNumFlag = newVal;
end;
// This function is used to populate the imputation flags (string variables)
function string setStrFlag(string name, occNum, impType, string newVal)
    numeric tmp;
    numeric tmp2;
    tmp = noccurs(FLAGS)+1;
    insert(FLAGS_EDT(tmp));
    flag_quest_name(tmp) = name;
    flag_occ_num(tmp) = occNum;
    flag_imputation_type(tmp) = impType;
    if length(strip(newVal)) = 0 then
        flag_new_value(tmp) = "EMPTY STRING / N.A.";
    else
        flag_new_value(tmp) = newVal;
    endif;
    if occNum = 0 then
        if length(strip(getvaluealpha(name))) = 0 then
            flag_original_value(tmp) = "EMPTY STRING";
        else
            flag_original_value(tmp) = getvaluealpha(name);
        endif;
    else
        if length(strip(getvaluealpha(name,occNum))) = 0 then
            flag_original_value(tmp) = "EMPTY STRING";
        else
            flag_original_value(tmp) = getvaluealpha(name,occNum);
        endif;
    endif;
    setStrFlag = newVal;
end;

PROC HIES_DICT_FF

// Process to check good households
// 1) Check if all the modules have at least

PROC HIES_QUEST


PROC ID07
postproc
    // set flag to make sure we only process the households that are part of the selection
    do varying currentHH = 0 until currentHH >= 180
        if tonumber(sampleFrame(currentHH,10)) = ID07 then
            hh_ok = 1;
            break;
        endif;
    enddo;

// This is duplicate code ready for deletion - the copy is in proc Q20225
// manual edits from Iapi's email sent on MAR 24 (house and tenure excel sheet)
//if ID07 = 117 then
//  impute (Q20225, 800);
//endif;

// manual edits from Iapi's email sent on MAR 29
if ID07 = 117 then
    impute (Q20223, 2);
    impute (Q20224, 2);
endif;
```

```
if ID07 = 302 then
    impute (Q20223, 4);
    impute (Q20224, 4);
endif;


// add missing home produced records
    if ID07 in 103, 327, 340 then
        Q40501_1 = setNumFlag({questName}'Q40501_1',{lineNo}curocc(),{imputeType}1,{newVal}1);
    endif;
// add missing agriculture records
    if ID07 in 306, 340 then
        Q40201_1 = setNumFlag({questName}'Q40201_1',{lineNo}curocc(),{imputeType}1,{newVal}1);
    endif;
PROC ID00
postproc
    $ = "TO";
PROC ID01
postproc
    // assign the island back into the dataset
    impute($,tonumber(sampleFrame(currentHH,3)));
PROC ID02
postproc
    // assign the village back into the dataset
    impute($,tonumber(sampleFrame(currentHH,5)));
PROC CV2
postproc
    // assign the replacement back into the dataset
    $ = sampleFrame(currentHH,8);

SAMPLE_SET = $;
PROC CV3
postproc
    // assign the round back into the dataset
    impute($,tonumber(sampleFrame(currentHH,9)));
PROC CV5B
postproc
    // assign the census code back into the dataset
    $ = sampleFrame(currentHH,7);
PROC CV6
postproc
    // let's remove all the telephone numbers - we don't need them
    $ = notappl;
PROC CV10
postproc
    // assign the island back into the dataset
    $ = sampleFrame(currentHH,2);
PROC CV11
    // assign the island back into the dataset
    impute($,tonumber(sampleFrame(currentHH,3)));
PROC CV12
    // assign the village back into the dataset
    $ = sampleFrame(currentHH,4);
PROC CV13
    // assign the village back into the dataset
    impute($,tonumber(sampleFrame(currentHH,5)));
PROC WEIGHT
// lets assign some weight for each island
if CV11 = 1 then
    $ = 2.33;
endif;
if CV11 = 2 then
    $ = 2.15;
endif;
if CV11 = 3 then
    $ = 1.85;
endif;

SAMPLE_WEIGHT = $;
PROC S1010_EDT
preproc
```

```
    // we need to delete blank lines on multiple occurring records
    if hh_ok then
        do varying i = noccurs(S1010_EDT) until i <= 0 by (-1)
            if age(i) in notappl and sex(i) in notappl then
                //errmsg("Deleting blank person lines") summary;
                delete(S1010_EDT(i));
            endif;
        enddo;
        if noccurs(S1020_EDT) = 0 then
            do varying i = 1 until i > noccurs(S1010_EDT)
                insert(S1020_EDT(i));
                //errmsg("1020 insert sucessful!");
            enddo;
        endif;
        if noccurs(S1030_EDT) = 0 then
            do varying i = 1 until i > noccurs(S1010_EDT)
                insert(S1030_EDT(i));
                //errmsg("1030 insert sucessful!");
            enddo;
        endif;
        if noccurs(S1040_EDT) = 0 then
            do varying i = 1 until i > noccurs(S1010_EDT)
                insert(S1040_EDT(i));
                //errmsg("1040 insert sucessful!");
            enddo;
        endif;
        if noccurs(S1050_EDT) = 0 then
            do varying i = 1 until i > noccurs(S1010_EDT)
                insert(S1050_EDT(i));
                //errmsg("1050 insert sucessful!");
            enddo;
        endif;
        if noccurs(S3061_EDT) = 0 then
            do varying i = 1 until i > maxocc(S3061_EDT)
                insert(S3061_EDT(i));
                //errmsg("3061 insert sucessful!");
            enddo;
        endif;
    endif;
    do varying i = 1 until i > noccurs(S1010_EDT)
        insert(PERSON_LEVEL_EDT(i));
    enddo;

PROC Q10101
postproc
    //reassign person line number
    if curocc() <> $ then
        impute($, curocc());
    endif;
PROC Q10105
// make sure only one head and if none assign the first person 15+ to be the head
if count(S1010_EDT where $ = 1) = 0 then
    if age >= 15 then
        impute($,setNumFlag(getsymbol(),curocc(),1,1));
    endif;
endif;
// check if the birthday is valid if not imput from hotdeck
PROC Q10106A
if $ in 1:31 then
    putdeck(hd_bDay_fr_sex_age,$);
else
    impute($, setNumFlag(getsymbol(),curocc(),3,getdeck(hd_bDay_fr_sex_age)));
endif;
// check if the birthmonth is valid if not imput from hotdeck
PROC Q10106B
if $ in 1:12 then
    putdeck(hd_bMonth_fr_sex_age,$);
else
    impute($,setNumFlag(getsymbol(),curocc(),3,getdeck(hd_bMonth_fr_sex_age))) ;
endif;
PROC Q10109
```

```
postproc
//make sure that person 15 and below are not married
if age in 0:14 then
    if $ <> 1 then
        impute($,setNumFlag(getsymbol(),curocc(),1,1));
    endif;
endif;
PROC S1020_EDT
// make sure that we put some space for persons in the labor force section
// just so that we know that all the data are aligned
preproc
    if hh_ok then
        numeric tmpOcc101;
        numeric tmpOcc102;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc102 = noccurs(S1020_EDT);
        // assign line numbers to matching records
        if tmpOcc101 = tmpOcc102 then
            do varying i = 1 until i > tmpOcc101
                impute(Q10200(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc102 = noccurs(S1020_EDT);
        if tmpOcc101 > tmpOcc102 then
            do varying i = 1 until i > tmpOcc102
                impute(Q10200(i), i);
            enddo;
            do varying i = tmpOcc102+1 until i > tmpOcc101
                insert(S1020_EDT(i));
                impute(Q10200(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc102 = noccurs(S1020_EDT);
        if tmpOcc101 < tmpOcc102 then
            do varying i = tmpOcc102 until i <= tmpOcc101 by (-1)
                delete(S1020_EDT(i));
            enddo;
            do varying i = 1 until i > tmpOcc102
                impute(Q10200(i), i);
            enddo;
        endif;

    endif;

PROC Q10201
// if the person is < 15 then we remove their informaiton
postproc
    if age(curocc()) < 15 then
        if $ <> notappl then
            impute ($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10202
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 9:14 then
        if length(strip($)) <> 0 then
            $ = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
    endif;
PROC Q10203
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 9:14 then
        if length(strip($)) <> 0 then
            $ = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
```

```
    endif;
PROC Q10204
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 9:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10205
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 9:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
    if Q10201 in 1:8 then
        if Q10204 >= 30 then
            if $ <> notappl then
                impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
            endif;
        endif;
    endif;
PROC Q10206
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10207
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if length(strip($)) <> 0 then
            $ = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
    endif;
    if Q10206 in 9:14 then
        if length(strip($)) <> 0 then
            $ = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
    endif;
PROC Q10207_C
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10208
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if length(strip($)) <> 0 then
            $ = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
    endif;
PROC Q10208_C
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
```

```
    if Q10201 in 12:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10209
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
    if Q10206 in 9:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
PROC Q10210
// making sure that skip pattern are followed and remove info for those that
// accidently added information.
postproc
    if Q10201 in 12:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
    if Q10206 in 9:14 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;
    if Q10206 in 1:8 then
        if Q10209 >= 30 then
            if $ <> notappl then
                impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
            endif;
        endif;
    endif;
PROC S1030_EDT
// align records
preproc
    if hh_ok then
        numeric tmpOcc101;
        numeric tmpOcc103;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc103 = noccurs(S1030_EDT);
        // assign line numbers to matching records
        if tmpOcc101 = tmpOcc103 then
            do varying i = 1 until i > tmpOcc101
                impute(Q10300(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc103 = noccurs(S1030_EDT);
        if tmpOcc101 > tmpOcc103 then
            do varying i = 1 until i > tmpOcc103
                impute(Q10300(i), i);
            enddo;
            do varying i = tmpOcc103+1 until i > tmpOcc101
                insert(S1030_EDT(i));
                impute(Q10300(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc103 = noccurs(S1030_EDT);
        if tmpOcc101 < tmpOcc103 then
            do varying i = tmpOcc103 until i <= tmpOcc101 by (-1)
                delete(S1030_EDT(i));
            enddo;
```

```
            do varying i = 1 until i > tmpOcc103
                impute(Q10300(i), i);
            enddo;
        endif;

    endif;

PROC Q10300
postproc
    if curocc() <> $ then
        impute($, curocc());
    endif;

    if ID07 in 159 and curocc() in 4 then
        impute(Q10301, setNumFlag({questName}'Q10301',{lineNo}4,{imputeType}1,{newVal}1));
        impute(Q10303, setNumFlag({questName}'Q10303',{lineNo}4,{imputeType}1,{newVal}1));
        impute(Q10306, setNumFlag({questName}'Q10306',{lineNo}4,{imputeType}1,{newVal}3));
        impute(Q10307, setNumFlag({questName}'Q10307',{lineNo}4,{imputeType}1,{newVal}1));
    endif;
PROC Q10303
postproc
    if Q10301 in 1 then
        // check if "now attending" is blank if they said that they have attended school
        if Q10303 = notappl then
            // if "now attending" is blank then we check to see if there is data in "why
            left school"
            // if there is valid data then we will assume that "now attending" is NO and
            impute
            if invalueset(Q10304) then
                impute(Q10303, setNumFlag("Q10303",curocc(),1,2));
            endif;
            // if there is valid data in "current grade level" lets impute YES for "now
            attendng"
            if invalueset(Q10306) then
                impute(Q10303, setNumFlag("Q10303",curocc(),1,1));
            endif;
        endif;
    endif;
    if age(curocc()) in 3:115 then
        if invalueset(Q10303) then
            if Q10301 <> 1 then
                impute (Q10301, setNumFlag("Q10301",curocc(),1,1));
            endif;
        endif;
    endif;
PROC Q10305
postproc
    if Q10303 in 1 then
        if Q10305 <> notappl then
            impute(Q10305, setNumFlag("Q10305",curocc(),1,notappl));
        endif;
    endif;
PROC S1050_EDT
preproc
    if hh_ok then
        numeric tmpOcc101;
        numeric tmpOcc105;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc105 = noccurs(S1050_EDT);
        // assign line numbers to matching records
        if tmpOcc101 = tmpOcc105 then
            do varying i = 1 until i > tmpOcc101
                impute(Q10500(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc105 = noccurs(S1050_EDT);
        if tmpOcc101 > tmpOcc105 then
            do varying i = 1 until i > tmpOcc105
                impute(Q10500(i), i);
            enddo;
```

```
            do varying i = tmpOcc105+1 until i > tmpOcc101
                insert(S1050_EDT(i));
                impute(Q10500(i), i);
            enddo;
        endif;
        tmpOcc101 = noccurs(S1010_EDT);
        tmpOcc105 = noccurs(S1050_EDT);
        if tmpOcc101 < tmpOcc105 then
            do varying i = tmpOcc105 until i <= tmpOcc101 by (-1)
                delete(S1050_EDT(i));
            enddo;
            do varying i = 1 until i > tmpOcc105
                impute(Q10500(i), i);
            enddo;
        endif;

    endif;

PROC Q10500
postproc
    if curocc() <> $ then
        impute($, curocc());
    endif;

//
// OK - this is where we will put all the person level manual edit corrections. We have to
put the corrections here, because we are making
// changes to the split records to align them - once they are aligned then we can make
changes in parallel.
//

//manual edits provided by Iapi on APR 12
hhId = ID07 * 100 + Q10101(curocc());
if hhId = 10303 then
    impute (age(curocc()), setNumFlag("Q10104",curocc(),1, 25));
endif;
if hhId = 15008 then
    impute (Q10106C(curocc()),setNumFlag("Q10106C",curocc(),1,2013));
endif;


// From Iapi's email sent on March 29
hhId = ID07 * 100 + Q10101(curocc());
if hhId in 12606 then
    impute (Q10105(curocc()), setNumFlag("Q10105",curocc(),1,3));
    impute (Q10106A(curocc()), setNumFlag("Q10106A",curocc(),1,8));
    impute (Q10106B(curocc()), setNumFlag("Q10106B",curocc(),1,11));
    impute (Q10106C(curocc()), setNumFlag("Q10106C",curocc(),1,2013));
    impute (Q10107(curocc()), setNumFlag("Q10107",curocc(),1,6));
    impute (Q10108(curocc()), setNumFlag("Q10108",curocc(),1,1));
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
endif;
if hhId in 33207 then
    impute (Q10105(curocc()), setNumFlag("Q10105",curocc(),1,9));
    impute (Q10106A(curocc()), setNumFlag("Q10106A",curocc(),1,8));
    impute (Q10106B(curocc()), setNumFlag("Q10106B",curocc(),1,11));
    impute (Q10106C(curocc()), setNumFlag("Q10106C",curocc(),1,2013));
    impute (Q10107(curocc()), setNumFlag("Q10107",curocc(),1,3));
    impute (Q10108(curocc()), setNumFlag("Q10108",curocc(),1,1));
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
endif;


// From Iapi List sent on March 24
if hhId in 10305 then
    impute (Q10211(curocc()), setNumFlag("Q10211",curocc(),1,2));
    impute (Q10212(curocc()), setNumFlag("Q10212",curocc(),1,1));
    impute (Q10213(curocc()), setNumFlag("Q10213",curocc(),1,2));
endif;
if hhId in 13802 then
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
```

```
endif;
if hhId in 13803 then
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
endif;
if hhId in 13804 then
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
endif;
if hhId in 15201 then
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
    impute (Q10206(curocc()), setNumFlag("Q10206",curocc(),1,13));
endif;
if hhId in 15202 then
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
    impute (Q10204(curocc()), setNumFlag("Q10204",curocc(),1,0));
    impute (Q10206(curocc()), setNumFlag("Q10206",curocc(),1,13));
endif;
{ from the list sent on March 24 - these 3 HHID need to be confirmed
  waiting for Iapi's response
if hhId in 15505 then
    impute (Q10106A(curocc()), );
    impute (Q10106B(curocc()), );
    impute (Q10106C(curocc()), );
endif;
if hhId in 15902 then
    impute (Q10106A(curocc()), );
    impute (Q10106B(curocc()), );
    impute (Q10106C(curocc()), );
endif;
if hhId in 15903 then
    impute (Q10106A(curocc()), );
    impute (Q10106B(curocc()), );
    impute (Q10106C(curocc()), );
endif;
}
if hhId = 20104 then
    impute (Q10105(curocc()), setNumFlag("Q10105",curocc(),1,7));
    impute (Q10106A(curocc()), setNumFlag("Q10106A",curocc(),1,11));
    impute (Q10106B(curocc()), setNumFlag("Q10106B",curocc(),1,06));
    impute (Q10106C(curocc()), setNumFlag("Q10106C",curocc(),1,2001));
    impute (Q10107(curocc()), setNumFlag("Q10107",curocc(),1,8));
    impute (Q10108(curocc()), setNumFlag("Q10108",curocc(),1,3));
    impute (Q10110(curocc()), setNumFlag("Q10110",curocc(),1,1));
    impute (Q10199(curocc()), setNumFlag("Q10199",curocc(),1,1));
endif;
if hhId in 23903 then
    impute (Q10204(curocc()), setNumFlag("Q10204",curocc(),1,40));
    impute (Q10205(curocc()), setNumFlag("Q10205",curocc(),1,1));
endif;
if hhId in 30403 then
    impute (Q10206(curocc()), setNumFlag("Q10206",curocc(),1,13));
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 31904 then
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 32205 then
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 32303 then
    impute (Q10301(curocc()), setNumFlag("Q10301",curocc(),1,1));
    impute (Q10302(curocc()), setNumFlag("Q10302",curocc(),1,notappl));
    impute (Q10303(curocc()), setNumFlag("Q10303",curocc(),1,1));
    impute (Q10304(curocc()), setNumFlag("Q10304",curocc(),1,notappl));
    impute (Q10305(curocc()), setNumFlag("Q10305",curocc(),1,notappl));
    impute (Q10306(curocc()), setNumFlag("Q10306",curocc(),1,7));
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 32403 then
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 32605 then
```

```
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;
if hhId in 32606 then
    impute (Q10307(curocc()), setNumFlag("Q10307",curocc(),1,1));
endif;

// catch left over birth year and impute based on int year - age
if Q10106C(curocc()) in notappl then
    impute($, setNumFlag("Q10106C",curocc(),1, IntYearMD1 - age(curocc())) );
endif;
// hot deck for birthplace for all the left overs
if invalueset(Q10107(curocc())) then
    putdeck(hd_BirthPlace_fr_sex,Q10107(curocc()));
else
    impute(Q10107(curocc()),setNumFlag("Q10107",curocc(),3,getdeck(hd_BirthPlace_fr_sex)));
endif;
// hotdeck for citizen for all the left overs
if invalueset(Q10108(curocc())) then
    putdeck(hd_Citizenship_fr_sex,Q10108(curocc()));
else
    impute(Q10108(curocc()),setNumFlag("Q10108",curocc(),3,getdeck(hd_Citizenship_fr_sex)));
endif;

PROC S1060_EDT
preproc
    if Q10600 <> 1 then
        if noccurs(S1060_EDT) > 0 then
            do varying i = noccurs(S1060_EDT) until i <= 0 by (-1)
                delete(S1060_EDT(i));
            enddo;
        endif;
    endif;

PROC Q20112C
postproc
    if $ in notappl then
        impute($, setNumFlag(getsymbol(),curocc(),1,5));
    endif;

PROC Q20122
postproc
    if $ in 2 then
        if Q20123 in 1:3 then
            impute($, setNumFlag(getsymbol(),curocc(),1,1));
        endif;
    endif;

PROC Q20123
postproc
    if Q20122 in 2 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
        if length(strip(Q20123_O)) <> 0 then
            Q20123_O = setStrFlag(getsymbol(),curocc(),1,"");
        endif;
    endif;

PROC Q20125
postproc
    if Q20124 = $ then
        impute($, setNumFlag(getsymbol(),curocc(),1,7));
    endif;

PROC Q20127
postproc
    if Q20126 = Q20127 then
        impute($, setNumFlag(getsymbol(),curocc(),1,9));
    endif;

PROC Q20131_1B
```

```
postproc
    if Q20131_1A = Q20131_1B then
        impute($, setNumFlag(getsymbol(),curocc(),1,8));
    endif;
PROC Q20131_2B
postproc
    if Q20131_2A = Q20131_2B then
        impute($, setNumFlag(getsymbol(),curocc(),1,8));
    endif;
PROC Q20131_3B
postproc
    if Q20131_3A = Q20131_3B then
        impute($, setNumFlag(getsymbol(),curocc(),1,8));
    endif;
PROC Q20133_B
postproc
    if Q20133_A = Q20133_B then
        impute($, setNumFlag(getsymbol(),curocc(),1,7));
    endif;
PROC Q20142
postproc
    if Q20141 in 2 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;

PROC Q20143
postproc
    if Q20141 in 2 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;

PROC Q20144
postproc
    if Q20141 in 2 or Q20143 in 2 then
        if $ <> notappl then
            impute($, setNumFlag(getsymbol(),curocc(),1,notappl));
        endif;
    endif;

    if Q20143 in 1 then
        if $ in 1:2 then
            putdeck(hd_20144_fr_tenure,$);
        else
            $ = setNumFlag(getsymbol(),curocc(),3,getdeck(hd_20144_fr_tenure));
        endif;
    endif;

PROC Q20145
postproc
    if $ in notappl then
        impute($,setNumFlag(getsymbol(),curocc(),1,2));
    endif;

PROC Q20212
if hh_ok then
    if $ > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.012); //change
        impute(M_COICOP(tmp),       041110901);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     1);
```

```
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][rent paid - household contribution for their
        own rent]";
    endif;
endif;
PROC Q20214
if hh_ok then
    if $ > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.014); //change
        impute(M_COICOP(tmp),       041110901);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     1);
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][rent paid (actual) - household contribution
        for their own rent]";
    endif;
endif;
PROC Q20217
//income
if hh_ok then
    if $ > 0 then
            // add to the income record as cash income from employer to pay for rent.
            numeric tmp = noccurs(INCOME_EDT)+1;
            insert(INCOME_EDT(tmp));
            impute(I_LINE_NUMBER(tmp),      tmp);
            impute(I_HIES_CODE(tmp),        202.017);
            impute(I_PACCOI(tmp),           1103);
            impute(I_COICOP(tmp),           notappl);
            impute(I_GROSS(tmp),            $);
            impute(I_MULTIPLIER(tmp),       12);
            impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
            impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
            impute(I_LOCATION(tmp),         1);
            impute(I_INCOME_TYPE(tmp),      1);// 1 = cash income 5= imputed rent; 7 =
            in-kind income
            impute(I_PROD_TYPE(tmp),        notappl);
            impute(I_QUANTITY(tmp),         notappl);
            impute(I_NUMBER(tmp),           notappl);
            impute(I_UNIT(tmp),             notappl);
            impute(I_BENEFICIARY(tmp),      51);
            I_DESCRIPTION(tmp) = "[tenure][cash income from employer for housing]";
    endif;
endif;
PROC Q20218
//expenditure
// this first recode checks 20211 if the house is free provided by the employer or by
another HH. If the house is provided by the
// employer then the income is income in-kind. if the house is provided by another household
then it is imputed rent - living for free
if hh_ok then
    // lets check free rent from another HH.
    if Q20211 in 3 then
        if $ > 0 then
            // add to the expenditure record as tenure
```

```
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.018); //change
        impute(M_COICOP(tmp),       042211401);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     5); // 5= imputed rent; 7 = in-kind income
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][imputed rent; living rent free from
        another HH]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.018);
        impute(I_PACCOI(tmp),           8200);
        impute(I_COICOP(tmp),           042211401);
        impute(I_GROSS(tmp),            $);
        impute(I_MULTIPLIER(tmp),       12);
        impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
        impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
        impute(I_LOCATION(tmp),         1);
        impute(I_INCOME_TYPE(tmp),      5);
        impute(I_PROD_TYPE(tmp),        notappl);
        impute(I_QUANTITY(tmp),         notappl);
        impute(I_NUMBER(tmp),           notappl);
        impute(I_UNIT(tmp),             notappl);
        impute(I_BENEFICIARY(tmp),      51);
        I_DESCRIPTION(tmp) = "[tenure][imputed rent; living rent free from another HH]";
    endif;
endif;

// now lets check free rent from employer.
if Q20211 in 2 then
    if $ > 0 then
        // add to the expenditure record as tenure
        tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.018); //change
        impute(M_COICOP(tmp),       042211401);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     5); // 5= imputed rent; 7 = in-kind income
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][imputed rent; living rent free from
        employer]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.018);
        impute(I_PACCOI(tmp),           8200);
        impute(I_COICOP(tmp),           042211401);
```

```
            impute(I_GROSS(tmp),            $);
            impute(I_MULTIPLIER(tmp),       12);
            impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
            impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
            impute(I_LOCATION(tmp),         1);
            impute(I_INCOME_TYPE(tmp),      5);// 5= imputed rent; 7 = in-kind income
            impute(I_PROD_TYPE(tmp),        notappl);
            impute(I_QUANTITY(tmp),         notappl);
            impute(I_NUMBER(tmp),           notappl);
            impute(I_UNIT(tmp),             notappl);
            impute(I_BENEFICIARY(tmp),      51);
            I_DESCRIPTION(tmp) =            "[tenure][imputed rent; living rent free from
            employer]";
            // add to the income record as income in-kind from employer
            //tmp = noccurs(INCOME_EDT)+1;
            //insert(INCOME_EDT(tmp));
            //impute(I_LINE_NUMBER(tmp),        tmp);
            //impute(I_HIES_CODE(tmp),      202.018);
            //impute(I_PACCOI(tmp),         notappl);
            //impute(I_COICOP(tmp),         notappl);
            //impute(I_GROSS(tmp),          $);
            //impute(I_MULTIPLIER(tmp),     12);
            //impute(I_AN_GROSS(tmp),           I_GROSS(tmp) * I_MULTIPLIER(tmp));
            //impute(I_WGHT_AN_GROSS(tmp),  I_AN_GROSS(tmp) * weight);
            //impute(I_LOCATION(tmp),           1);
            //impute(I_INCOME_TYPE(tmp),        7);// 5= imputed rent; 7 = in-kind income
            //impute(I_PROD_TYPE(tmp),      notappl);
            //impute(I_QUANTITY(tmp),           notappl);
            //impute(I_NUMBER(tmp),         notappl);
            //impute(I_UNIT(tmp),               notappl);
            //impute(I_BENEFICIARY(tmp),        51);
            //I_DESCRIPTION(tmp) = "[tenure][in-kind income; living rent free from employer]";
            //Commented out because it is already reported in the in-kind wages section in
            module 4
            //in-kind
        endif;
    endif;


endif;
PROC Q20219
postproc
    if Q20211 in 3 then
        if Q20219 <> notappl then
            impute(Q20219,
            setNumFlag({questName}'Q20219',{lineNo}curocc(),{imputeType}1,{newVal}notappl));
        endif;
    endif;
    if Q20211 in 4 then
        if Q20219 = notappl then
            if Q20218 >= 0 then
                impute(Q20219,
                setNumFlag({questName}'Q20219',{lineNo}curocc(),{imputeType}1,{newVal}Q20218))
                ;
                impute(Q20218,
                setNumFlag({questName}'Q20218',{lineNo}curocc(),{imputeType}1,{newVal}notappl)
                );
            endif;
        endif;
    endif;
    if ID07 in 304 then
        if Q20219 in 0,notappl then
            impute(Q20219,
            setNumFlag({questName}'Q20219',{lineNo}curocc(),{imputeType}1,{newVal}500));
        endif;
    endif;
    if ID07 in 332 then
        if Q20219 in 5200 then
            impute(Q20219,
            setNumFlag({questName}'Q20219',{lineNo}curocc(),{imputeType}1,{newVal}500));
        endif;
```

```
    endif;

if hh_ok then
    // lets check owner occipied housing
    if $ > 0 then
        // add to the expenditure record as tenure
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.019); //change
        impute(M_COICOP(tmp),       042111201);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     5); // 5= imputed rent
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][imputed rent; owner occupied without a
        mortgagte]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.019);
        impute(I_PACCOI(tmp),           8100);
        impute(I_COICOP(tmp),           042111201);
        impute(I_GROSS(tmp),            $);
        impute(I_MULTIPLIER(tmp),       12);
        impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
        impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
        impute(I_LOCATION(tmp),         1);
        impute(I_INCOME_TYPE(tmp),      5); // 5= imputed rent
        impute(I_PROD_TYPE(tmp),        notappl);
        impute(I_QUANTITY(tmp),         notappl);
        impute(I_NUMBER(tmp),           notappl);
        impute(I_UNIT(tmp),             notappl);
        impute(I_BENEFICIARY(tmp),      51);
        I_DESCRIPTION(tmp) = "[tenure][imputed rent; owner occupied without a mortgage]";
    endif;
endif;
PROC Q20221
if hh_ok then
    // owner occupied with mortgage.
    if Q20211 in 5 then
        if $ > 0 then
            // add to the expenditure record as tenure
            numeric tmp = noccurs(MODULES_EDT)+1;
            insert(MODULES_EDT(tmp));
            impute(M_LINE_NUMBER(tmp),  tmp);
            impute(M_HIES_CODE(tmp),    202.021); //change
            impute(M_COICOP(tmp),       902200903);
            impute(M_AMOUNT(tmp),       $);

            if Q20220 in 1 then
                i = 26;
            elseif Q20220 in 2 then
                i = 12;
            else
                i = 1;
            endif;
            impute(M_MULTIPLIER(tmp),   i); //change
            impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
            impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
            impute(M_LOCATION(tmp),     1);
            impute(M_EXP_TYPE(tmp),     1); // Cash expenditure
```

```
            impute(M_PROD_TYPE(tmp),    notappl);
            impute(M_QUANTITY(tmp),     notappl);
            impute(M_NUMBER(tmp),       notappl);
            impute(M_UNIT(tmp),         notappl);
            impute(M_LOCAL(tmp),        notappl);
            impute(M_BENEFICIARY(tmp),  51);
            M_DESCRIPTION(tmp) =        "[tenure][investment; mortgage payment]";
        endif;
    endif;
endif;

PROC Q20222
if hh_ok then
    // lets check owner occipied housing
    if $ > 0 then
        // add to the expenditure record as tenure
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.022); //change
        impute(M_COICOP(tmp),       042111201);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     5); // 5= imputed rent
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][imputed rent; owner occupied with a mortgage]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.022);
        impute(I_PACCOI(tmp),           8100);
        impute(I_COICOP(tmp),           042111201);
        impute(I_GROSS(tmp),            $);
        impute(I_MULTIPLIER(tmp),       12);
        impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
        impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
        impute(I_LOCATION(tmp),         1);
        impute(I_INCOME_TYPE(tmp),      5); // 5= imputed rent
        impute(I_PROD_TYPE(tmp),        notappl);
        impute(I_QUANTITY(tmp),         notappl);
        impute(I_NUMBER(tmp),           notappl);
        impute(I_UNIT(tmp),             notappl);
        impute(I_BENEFICIARY(tmp),      51);
        I_DESCRIPTION(tmp) =            "[tenure][imputed rent; owner occupied with a
        mortgage]";
    endif;
endif;
PROC Q20225
postproc
    if Q20224 in 1:3 then
        if Q20225 in notappl then
            if Q20232 >= 0 then
                impute(Q20225,
                setNumFlag({questName}"Q20225",{lineNo}curocc(),{imputeType}1,{newVal}Q20232))
                ;
                impute(Q20232,
                setNumFlag({questName}"Q20232",{lineNo}curocc(),{imputeType}1,{newVal}notappl)
                );
            endif;
        endif;
        if Q20225 in notappl then
            impute(Q20225,
```

```
            setNumFlag({questName}"Q20225",{lineNo}curocc(),{imputeType}1,{newVal}0));
        endif;
    endif;
    if Q20224 in 4 then
        if Q20225 <> notappl then
            impute(Q20225,
            setNumFlag({questName}"Q20225",{lineNo}curocc(),{imputeType}1,{newVal}notappl));
        endif;
    endif;

    // manual fix from Iapi on 24/3/16
    if ID07 = 141 then
        if Q20224 in 1:3 then
            if Q20225 in 0, notappl then
                impute(Q20225,
                setNumFlag({questName}"Q20225",{lineNo}curocc(),{imputeType}1,{newVal}800));
                // missing on entry; value refrenced from the original form
            endif;
        endif;
    endif;


{   this was removed because of a recommendation from the HIES TWG
if hh_ok then
    // lets check owner occipied housing
    if $ > 0 then
        // add to the expenditure record as tenure
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.025); //change
        impute(M_COICOP(tmp),       042211301);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     5); // 5= imputed rent
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][imputed rent; household's secondary dwelling]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.025);
        impute(I_PACCOI(tmp),           notappl);
        impute(I_COICOP(tmp),           042211301);
        impute(I_GROSS(tmp),            $);
        impute(I_MULTIPLIER(tmp),       12);
        impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
        impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
        impute(I_LOCATION(tmp),         1);
        impute(I_INCOME_TYPE(tmp),      5); // 5= imputed rent
        impute(I_PROD_TYPE(tmp),        notappl);
        impute(I_QUANTITY(tmp),         notappl);
        impute(I_NUMBER(tmp),           notappl);
        impute(I_UNIT(tmp),             notappl);
        impute(I_BENEFICIARY(tmp),      51);
        I_DESCRIPTION(tmp) =            "[tenure][imputed rent; household's secondary
        dwelling]";
    endif;
endif;
}
PROC Q20231
postproc
    if Q20231 in notappl then
```

```
        impute (Q20231,
        setNumFlag({questName}"Q20231",{lineNo}curocc(),{imputeType}1,{newVal}2));
    endif;

PROC Q20232
if hh_ok then
    // pay the rent for another HH
    if $ > 0 then
        // add to the expenditure record as tenure
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);
        impute(M_HIES_CODE(tmp),    202.032); //change
        impute(M_COICOP(tmp),       041110901);
        impute(M_AMOUNT(tmp),       $);
        impute(M_MULTIPLIER(tmp),   12); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        impute(M_LOCATION(tmp),     1);
        impute(M_EXP_TYPE(tmp),     3); // 3 = given away (bought goods)
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        "[tenure][rent paid; for the benefit of another
        household]";
        // add to the income record as tenure
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));
        impute(I_LINE_NUMBER(tmp),      tmp);
        impute(I_HIES_CODE(tmp),        202.032);
        impute(I_PACCOI(tmp),           7204);
        impute(I_COICOP(tmp),           041110901);
        impute(I_GROSS(tmp),            $);
        impute(I_MULTIPLIER(tmp),       12);
        impute(I_AN_GROSS(tmp),         I_GROSS(tmp) * I_MULTIPLIER(tmp));
        impute(I_WGHT_AN_GROSS(tmp),    I_AN_GROSS(tmp) * weight);
        impute(I_LOCATION(tmp),         1);
        impute(I_INCOME_TYPE(tmp),      3); // 3 = given away (bought goods)
        impute(I_PROD_TYPE(tmp),        notappl);
        impute(I_QUANTITY(tmp),         notappl);
        impute(I_NUMBER(tmp),           notappl);
        impute(I_UNIT(tmp),             notappl);
        impute(I_BENEFICIARY(tmp),      51);
        I_DESCRIPTION(tmp) =            "[tenure][rent paid; for the benefit of another
        household]";
    endif;
endif;
PROC S2032_EDT
preproc
    numeric tmpNum;

    // these transactions have a landline but no phonebill so we will impute them
    if ID07 in 255,206,209,222,208,217,111,120,126,159,204,210,224,319,333,351 then
        tmpNum = noccurs(S2032_EDT)+1;
        insert(S2032_EDT(tmpNum));
        impute(Q20351(tmpNum),
        setNumFlag({questName}'Q20351',{lineNo}tmpNum,{imputeType}1,{newVal}tmpNum));
        impute(Q20352(tmpNum),
        setNumFlag({questName}'Q20352',{lineNo}tmpNum,{imputeType}1,{newVal}51));
        Q20353(tmpNum) =
        setStrFlag({questName}'Q20353',{lineNo}tmpNum,{imputeType}1,{newVal}"LANDLINE PHONE
        BILL");
        impute(Q20354(tmpNum),
        setNumFlag({questName}'Q20354',{lineNo}tmpNum,{imputeType}1,{newVal}1));
        impute(Q20355(tmpNum),
        setNumFlag({questName}'Q20355',{lineNo}tmpNum,{imputeType}4,{newVal}51));
        impute(Q20356(tmpNum),
        setNumFlag({questName}'Q20356',{lineNo}tmpNum,{imputeType}1,{newVal}1));
```

```
        impute(Q203_COICOP(tmpNum),
        setNumFlag({questName}'Q203_COICOP',{lineNo}tmpNum,{imputeType}1,{newVal}083020904));
    endif;
    if noccurs(S2032_EDT) > 0 then
        do varying i = noccurs(S2032_EDT) until i <= 0 by (-1)
            if length(strip(Q20353(i))) < 2 then
                delete(S2032_EDT(i));
                //errmsg("deleting blank records in section S2032");
            endif;
        enddo;
    endif;

postproc
    numeric j;
    do varying j = 1 until j > noccurs($)
        // let's start transfering the data to the aggregated data record
        if Q20355(j) > 0 then
            numeric tmp = noccurs(MODULES_EDT)+1;
            insert(MODULES_EDT(tmp));
            impute(M_LINE_NUMBER(tmp),  tmp);

            if Q20352(j) in 11:57 then
                i = 203 + (Q20352(j)/1000);
            else
                i = 203.999;
            endif;
            impute(M_HIES_CODE(tmp),    i); //change
            impute(M_COICOP(tmp),       Q203_COICOP(j));
            impute(M_AMOUNT(tmp),       Q20355(j));
            i = 12;
            if Q20352(j) in 52,57 then
                i = 1;
            endif;
            impute(M_MULTIPLIER(tmp),   i); //change
            impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
            impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
            if Q20356(j) in 1 then
                i = 1;
            else
                i = 2;
            endif;
            impute(M_LOCATION(tmp),     i);
            impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
            impute(M_QUANTITY(tmp),     notappl);
            impute(M_NUMBER(tmp),       notappl);
            impute(M_UNIT(tmp),         notappl);
            impute(M_LOCAL(tmp),        notappl);

            impute(M_BENEFICIARY(tmp),  Q20354(j) + 50);
            if M_BENEFICIARY(tmp) = 51 then
                i = 1;
            else
                i = 3;
            endif;
            impute(M_EXP_TYPE(tmp),     i);
            M_DESCRIPTION(tmp) =        concat("[utility exp]",tolower(strip(Q20353(j))));
        endif;
    enddo;

PROC Q20351
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
    // manual correction based on existing data 24/3/16
    // "32" was written in the description and imputed for Kerosene and with a location of
    // provider is "on island"
    if ID07 = 151 and $ = 2 then
        if Q20352 in notappl then

            impute(Q20352,setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVa
```

```
            l}32));
            Q20353 =
            setStrFlag({questName}'Q20353',{lineNo}curocc(),{imputeType}1,{newVal}"KEROSENE");
            impute(Q20356,
            setNumFlag({questName}'Q20356',{lineNo}curocc(),{imputeType}1,{newVal}1));
        endif;
    endif;


PROC Q20352
if hh_ok then

    // these values that are being corrected are from Iapi's email on April 7.
    if ID07 in 232 and Q20351 in 02 and Q20352 in 32 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,30)); endif;
    if ID07 in 322 and Q20351 in 06 and Q20352 in 56 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,100)); endif;
    if ID07 in 327 and Q20351 in 02 and Q20352 in 31 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,100)); endif;
    if ID07 in 342 and Q20351 in 01 and Q20352 in 11 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,42)); endif;
    if ID07 in 309 and Q20351 in 06 and Q20352 in 52 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,10)); endif;
    if ID07 in 332 and Q20351 in 05 and Q20352 in 57 then impute(Q20355,
    setNumFlag("Q20355",curocc(),1,42)); endif;

    if invalueset(Q20356) and Q20356 <> notappl then
        putdeck(hd_20356_fr_tenure_expCode,Q20356);
    else
        Q20356 = setNumFlag("Q20356",curocc(),1,getdeck(hd_20356_fr_tenure_expCode));
    endif;

    // We are assuming that Satellite TV and Online subscriptions are always provided
    outside the country.
    if $ in 55,56 then
        if Q20356 <> 3 then
            impute(Q20356, setNumFlag("Q20356",curocc(),1,3)); // 3 = outside Tokelau and
            Samoa
        endif;
    endif;

    // these three transactions had a wrong expenditure code
    if ID07 in 332 then
        if Q20351 in 5  then
            impute(Q20352,
            setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVal}53));
        endif;
    endif;

    if ID07 in 320 then
        if Q20351 in 2  then
            impute(Q20352,
            setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVal}53));
        endif;
    endif;

    if ID07 in 322 then
        if Q20351 in 1  then
            impute(Q20352,
            setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVal}53));
        endif;
    endif;

    if ID07 in 326 then
        if Q20351 in 1  then
            impute(Q20352,
            setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVal}53));
        endif;
    endif;

    if ID07 in 307 then
```

```
        if Q20351 in 3  then
            impute(Q20352,
            setNumFlag({questName}'Q20352',{lineNo}curocc(),{imputeType}1,{newVal}53));
        endif;
    endif;

    // this transaction bought petrol for a scooter - wrong section so let's delete it
    if ID07 in 305 then
        if Q20351 in 2  then
            impute(Q20355,
            setNumFlag({questName}'Q20355',{lineNo}curocc(),{imputeType}1,{newVal}0));
        endif;
    endif;



    recode Q20352 => Q203_COICOP ;
        11 => 045112501; // prep-paid elect
        12 => 045112502; // other elect
        21 => 045312803; // diesel for gen
        22 => 045312802; // gasoline for gen
        23 => 045312899; // other generator fuels
        31 => 045212702; // LPG or Butane
        32 => 045312801; // kerosene
        33 => 045312899; // other liquid cooking fuels
        41 => 045412901; // firewood / coconut husk
        42 => 045412902; // charcoal
        51 => 083020904; // landline telephone
        52 => 083020906; // other landline costs
        53 => 083020905; // home internet via dialup
        54 => 083020905; // home internet via WiFi
        55 => 094224703; // online subscriptions
        56 => 094224702; // satellite TV suibscriptions
        57 => 999999999; // other
    endrecode;

endif;
PROC S2042_EDT
preproc
    if noccurs(S2042_EDT) > 0 then
        do varying i = noccurs(S2042_EDT) until i <= 0 by (-1)
            if length(strip(Q20453(i))) < 2 then
                delete(S2042_EDT(i));
                //errmsg("deleting blank records in section S2042") summary;
            endif;
        enddo;
    endif;

PROC Q20451
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;

PROC Q20452
if hh_ok then

    // these values have been imputed from marketvalue on April 18 from Iapi
    if ID07 =   232 and Q20451 =    1   and Q20452 =    351 then Q20456 =
    setNumFlag("Q20456",curocc(),1,98)  ;   endif;
    if ID07 =   233 and Q20451 =    1   and Q20452 =    204 then Q20456 =
    setNumFlag("Q20456",curocc(),1,77)  ;   endif;
    if ID07 =   233 and Q20451 =    3   and Q20452 =    331 then Q20456 =
    setNumFlag("Q20456",curocc(),1,878) ;   endif;
    if ID07 =   233 and Q20451 =    4   and Q20452 =    341 then Q20456 =
    setNumFlag("Q20456",curocc(),1,25)  ;   endif;
    if ID07 =   241 and Q20451 =    1   and Q20452 =    311 then Q20456 =
    setNumFlag("Q20456",curocc(),1,389) ;   endif;
    if ID07 =   241 and Q20451 =    2   and Q20452 =    351 then Q20456 =
    setNumFlag("Q20456",curocc(),1,196) ;   endif;
```

```
    // let's start transfering the data to the aggregated data record
    if length(strip(Q20453)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if Q20452 in 101:422 then
            i = 204 + ( Q20452/1000);
        else
            i = 204.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q204_COICOP);
        impute(M_AMOUNT(tmp),       Q20456);
        i = 1; // annual multiplier
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20457 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  Q20455 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[land exp]",tolower(strip(Q20453)));
    endif;
endif;
PROC S2052_EDT
preproc
    if noccurs(S2052_EDT) > 0 then
        do varying i = noccurs(S2052_EDT) until i <= 0 by (-1)
            if length(strip(Q20553(i))) < 2 then
                delete(S2052_EDT(i));
                //errmsg("deleting blank records in section S2052") summary;
            endif;
        enddo;
    endif;

PROC Q20551
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;

PROC Q20552
if hh_ok then
    // these values have been imputed from marketvalue on April 18 from Iapi
    if ID07 =   106 and Q20551 =    2   and Q20552 =    107 then Q20556 =
    setNumFlag("Q20556",curocc(),1,210) ;   endif;
    if ID07 =   106 and Q20551 =    3   and Q20552 =    202 then Q20556 =
    setNumFlag("Q20556",curocc(),1,225) ;   endif;
    if ID07 =   106 and Q20551 =    6   and Q20552 =    304 then Q20556 =
    setNumFlag("Q20556",curocc(),1,525) ;   endif;
    if ID07 =   106 and Q20551 =    7   and Q20552 =    305 then Q20556 =
    setNumFlag("Q20556",curocc(),1,150) ;   endif;
    if ID07 =   106 and Q20551 =    9   and Q20552 =    307 then Q20556 =
    setNumFlag("Q20556",curocc(),1,270) ;   endif;
    if ID07 =   106 and Q20551 =    10  and Q20552 =    308 then Q20556 =
```

```
    setNumFlag("Q20556",curocc(),1,350) ;   endif;
    if ID07 =   106 and Q20551 =    12  and Q20552 =    401 then Q20556 =
    setNumFlag("Q20556",curocc(),1,350) ;   endif;

    // let's start transfering the data to the aggregated data record
    if length(strip(Q20553)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 205 + ( $/1000);
        else
            i = 205.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q205_COICOP);
        impute(M_AMOUNT(tmp),       Q20556);
        i = 1; // annual multiplier
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20557 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  Q20555 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[hh-goods exp]",tolower(strip(Q20553)));
    endif;
endif;
PROC Q20602
if curocc() in 1:11 then
    if ! $ in 0:9 then
        impute ($, 0);
    endif;
    if ID07 in
        102,
        103,
        104,
        105,
        106,
        108,
        109,
        111,
        113,
        114,
        116,
        117,
        118,
        119,
        120,
        121,
        124,
        125,
        126,
        127,
        150,
        154,
```

```
        158,
        227,
        308,
        310,
        324,
        327,
        328,
        343,
        356
    then
        if $ <> 1 then
            impute ($, 0);
        endif;
    endif;
else
    if $ <> notappl then
        impute ($, notappl);
    endif;
endif;
PROC S2062_EDT
preproc
    if noccurs(S2062_EDT) > 0 then
        do varying i = noccurs(S2062_EDT) until i <= 0 by (-1)
            if length(strip(Q20653(i))) < 2 then
                delete(S2062_EDT(i));
                //errmsg("deleting blank records in section S2062") summary;
            endif;
        enddo;
    endif;

PROC Q20651
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;

PROC Q20652
if hh_ok then
    // these values have been imputed from Iapi's email on 7 april
    if ID07 =   148 and Q20651 =    1   and Q20652 =    502 then Q20657 =
    setNumFlag("Q20657",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   134 and Q20651 =    1   and Q20652 =    502 then Q20656 =
    setNumFlag("Q20656",curocc(),1,180) ;   endif;
    if ID07 =   150 and Q20651 =    1   and Q20652 =    502 then Q20657 =
    setNumFlag("Q20657",curocc(),1,1)   ;   endif;
    if ID07 =   159 and Q20651 =    13  and Q20652 =    502 then Q20657 =
    setNumFlag("Q20657",curocc(),1,1)   ;   endif;
    if ID07 =   159 and Q20651 =    14  and Q20652 =    504 then Q20657 =
    setNumFlag("Q20657",curocc(),1,1)   ;   endif;
    if ID07 =   138 and Q20651 =    1   and Q20652 =    4   then Q20652 =
    setNumFlag("Q20652",curocc(),1,502) ;   endif;

    // let's start transfering the data to the aggregated data record
    if length(strip(Q20653)) > 0 then
        if Q20652 in 101:501, 504:505 then
            numeric tmp = noccurs(MODULES_EDT)+1;
            insert(MODULES_EDT(tmp));
            impute(M_LINE_NUMBER(tmp),  tmp);

            if invalueset($) and $ <> notappl then
                i = 206 + ( $/1000);
            else
                i = 206.999;
            endif;
            impute(M_HIES_CODE(tmp),    i); //change
            impute(M_COICOP(tmp),       Q206_COICOP);
            impute(M_AMOUNT(tmp),       Q20656);
            i = 1; // assume all expenditures are reported as annual amounts and adjust
            according to the expenditure code
            if $ in 501:505 then
                i = 12 // montly expenditure - for fuels and other lubricants
```

```
        endif;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20657 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  Q20655 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[vehicle exp]",tolower(strip(Q20653)));
    else
        numeric tmpCOICOP = notappl; // coded below
        numeric tmpPACCOI = notappl; // coded below
        numeric tmpHIEScode = notappl; //coded below
        numeric tmpMulti = 12;
        numeric tmpISCO = notappl; // coded below
        numeric tmpISIC = notappl; // coded below
        numeric tmpValue = Q20656;
        numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind
        home  [5]rent  [6]intermediate  [7]in-kind income
        numeric tmpLocation = 1;
        numeric tmpBeneficiary = Q20655 + 50;
        string tmpDescription = ""; // coded below
        numeric tmpIncomeCategory = notappl;
        numeric tmpProductType = notappl;
        numeric tmpQuantity = notappl;
        numeric tmpNumber = notappl;
        numeric tmpUnit = notappl;
        numeric tmpCategory = notappl;
        numeric tmpDurability = notappl;
        if Q20652 in 502 then
            tmpCOICOP = 904106301;
            tmpPACCOI = 9305;
            tmpHIEScode = 206.502;
            tmpISCO = 634;
            tmpISIC = 31;
            tmpDescription ="[vehicle exp][intermediate][fuel for boat fishing]" +
            tolower(strip(Q20653));
        endif;
        if Q20652 in 503 then
            tmpCOICOP = 904102306;
            tmpPACCOI = 9106;
            tmpHIEScode = 206.503;
            tmpISCO = 835;
            tmpISIC = 501;
            tmpDescription ="[vehicles][intermediate][fuel for boat transport]" +
            tolower(strip(Q20653));
        endif;

        numeric tmp1 = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp1));

        I_LINE_NUMBER(tmp1) = tmp1;
        I_HIES_CODE(tmp1) = tmpHIEScode;
        I_PACCOI(tmp1) = tmpPACCOI;
        I_COICOP(tmp1) = tmpCOICOP;
        I_GROSS(tmp1) = tmpValue;
        I_AN_GROSS(tmp1) = I_GROSS(tmp1) * tmpMulti;
```

```
            I_WGHT_AN_GROSS(tmp1) = I_AN_GROSS(tmp1) * WEIGHT;
            I_MULTIPLIER(tmp1) = tmpMulti;
            I_LOCATION(tmp1) = tmpLocation;
            I_INCOME_TYPE(tmp1) = tmpIncomeType;
            I_PROD_TYPE(tmp1) = tmpProductType;
            I_QUANTITY(tmp1) = tmpQuantity;
            I_NUMBER(tmp1) = tmpNumber;
            I_UNIT(tmp1) = tmpUnit;
            I_CATEGORY(tmp1) = tmpIncomeCategory;
            I_BENEFICIARY(tmp1) = tmpBeneficiary;
            I_DURABILITY(tmp1) = tmpDurability;
            I_INDUSTRY(tmp1) = tmpISIC;
            I_OCCUPATION(tmp1) = tmpISCO;
            I_DESCRIPTION(tmp1) = tmpDescription;
        endif;
    endif;
endif;

PROC S2072_EDT
preproc
    if noccurs(S2072_EDT) > 0 then
        do varying i = noccurs(S2072_EDT) until i <= 0 by (-1)
            if length(strip(Q20724(i))) < 2 then
                delete(S2072_EDT(i));
                //errmsg("deleting blank records in section S2072") summary;
            endif;
        enddo;
    endif;
PROC Q20721
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;

PROC Q20723
if hh_ok then
    // these values have been imputed from Iapi's email on 7 april
    if ID07 =   135 and Q20721 =    1   and Q20723 =    5   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   141 and Q20721 =    1   and Q20723 =    2   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   160 and Q20721 =    1   and Q20723 =    1   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   160 and Q20721 =    2   and Q20723 =    2   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   160 and Q20721 =    3   and Q20723 =    3   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   160 and Q20721 =    4   and Q20723 =    6   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   329 and Q20721 =    2   and Q20723 =    2   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   329 and Q20721 =    3   and Q20723 =    3   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed
    if ID07 =   329 and Q20721 =    4   and Q20723 =    4   then Q20722 =
    setNumFlag("Q20722",curocc(),1,1)   ;   endif; //manually imputed

    if ID07 =   135 and Q20721 =    1   and Q20723 =    5   then Q20727 =
    setNumFlag("Q20727",curocc(),1,2)   ;   endif; //manually imputed

    // let's start transfering the data to the aggregated data record
    if length(strip(Q20724)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 207 + 0.2 + ( $/1000);
        else
            i = 207.299;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
```

```
        impute(M_COICOP(tmp),       Q207A_COICOP);
        impute(M_AMOUNT(tmp),       Q20726);
        // multiplier
        i = 1; // assume all expenditures are reported as annual amounts and adjust
        according to the expenditure code
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20727 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  Q20725 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[int travel exp]",tolower(strip(Q20724)));
    endif;
endif;
PROC S2074_EDT
preproc
    if noccurs(S2074_EDT) > 0 then
        do varying i = noccurs(S2074_EDT) until i <= 0 by (-1)
            // some data cleaning
            if ID07 in 208 and Q20771(i) in 1 and Q20773(i) in 3 then  impute(Q20773(i),
            setNumFlag({questName}"Q20773",{lineNo}i,{imputeType}1,{newVal}notappl)); endif;
            // delete blank transactions
            if length(strip(Q20774(i))) < 2 and !Q20773(i) in 1:3 then
                delete(S2074_EDT(i));
                //errmsg("deleting blank records in section S2074") summary;
            endif;
        enddo;
    endif;
PROC Q20771
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;

PROC Q20773
if hh_ok then
    // tansfer this data to the aggrigated expenditure record
    if length(strip(Q20774)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 207 + 0.4 + ( $/1000);
        else
            i = 207.499;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q207B_COICOP);
        impute(M_AMOUNT(tmp),       Q20776);
        // multiplier
        i = 1; // assume all expenditures are reported as annual amounts and adjust
        according to the expenditure code
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
```

```
        if Q20777 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),      i);
        impute(M_PROD_TYPE(tmp),     notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),      notappl);
        impute(M_NUMBER(tmp),        notappl);
        impute(M_UNIT(tmp),          notappl);
        impute(M_LOCAL(tmp),         notappl);
        impute(M_BENEFICIARY(tmp),   Q20775 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),      i);
        M_DESCRIPTION(tmp) =         concat("[dom travel exp]",tolower(strip(Q20774)));
    endif;
endif;
PROC S2080_EDT
postproc
    if noccurs(S2080_EDT) > 0 then
        do varying i = noccurs(S2080_EDT) until i <= 0 by (-1)
            if Q20802(i)= 2 then
                delete(S2080_EDT(i));
                //errmsg("deleting non-expense records in section S2080") summary;
            endif;
        enddo;
    endif;

PROC Q20802
postproc
    if not $ in 1:2 then
        impute($, setNumFlag(getsymbol(),curocc(),1,2));
    endif;

PROC Q20803
numeric rcd_coicop;
if hh_ok then
    // some data cleaning from Iapi on 7 april
    if ID07 in 141 and Q20801 in 8 and Q20803 in 203 then impute(Q20807,
    setNumFlag("Q20807",curocc(),1,1)); endif;
    if ID07 in 141 and Q20801 in 9 and Q20803 in 204 then impute(Q20807,
    setNumFlag("Q20807",curocc(),1,2)); endif;
    if ID07 in 160 and Q20801 in 9 and Q20803 in 204 then impute(Q20805,
    setNumFlag("Q20805",curocc(),1,1)); endif;
    if ID07 in 320 and Q20801 in 12 and Q20803 in 207 then impute(Q20807,
    setNumFlag("Q20807",curocc(),1,1)); endif;
    if ID07 in 155 and Q20801 in 10 and Q20803 in 205 then impute(Q20806,
    setNumFlag("Q20806",curocc(),1,260)); endif;

    // this next line has to be swapped because it was entered in the wrong area
    // record the new line of data in its proper place first
    if ID07 in 326 and Q20801 in 6 and Q20803 in 201 then impute(Q20802,
    setNumFlag("Q20802",curocc(),1,1)); endif;
    if ID07 in 326 and Q20801 in 6 and Q20803 in 201 then impute(Q20805,
    setNumFlag("Q20805",curocc(),1,1)); endif;
    if ID07 in 326 and Q20801 in 6 and Q20803 in 201 then impute(Q20806,
    setNumFlag("Q20806",curocc(),1,80)); endif;
    if ID07 in 326 and Q20801 in 6 and Q20803 in 201 then impute(Q20807,
    setNumFlag("Q20807",curocc(),1,1)); endif;
    // delete the data that was entered on the wrong line
    if ID07 in 326 and Q20801 in 10 and Q20803 in 205 then impute(Q20802,
    setNumFlag("Q20802",curocc(),1,2)); endif;
    if ID07 in 326 and Q20801 in 10 and Q20803 in 205 then impute(Q20805,
    setNumFlag("Q20805",curocc(),1,notappl)); endif;
    if ID07 in 326 and Q20801 in 10 and Q20803 in 205 then impute(Q20806,
    setNumFlag("Q20806",curocc(),1,notappl)); endif;
    if ID07 in 326 and Q20801 in 10 and Q20803 in 205 then impute(Q20807,
```

```
    setNumFlag("Q20807",curocc(),1,notappl)); endif;


    // tansfer this data to the aggrigated expenditure record
    // we found out that some households have a very high reported western union fees
    // for these households we imputed the mean of only the households that reported
    // fees in exp_code 206 between $2 and $120 which equates to $43.875 = $44
    if Q20802 in 1 then
        if $ in 206 then
            if Q20806 > 120 then
                impute(Q20806, setNumFlag("Q20806",curocc(),4,44));
            endif;
        endif;
    endif;
    // Lets add some default COICOP codes to the data
    if Q20802 in 1 then
        recode $ =>  Q208_COICOP;
            101 => 056217304; // gardener/landscaping
            102 => 056217306; // maid
            103 => 056217309; // baby sitting
            104 => 056217310; // elderly care
            105 => 056217398; // other

            201 => 127029901; // passport
            202 => 127029905; // legal service
            203 => 073620501; // freight destination - Tokelau
            204 => 073620501; // freight destination - Samoa
            205 => 126229803; // credit card fees
            206 => 126229802; // international money transfer
            207 => 126229801; // financial or banking fees
                => 999999999; // default for missng
        endrecode;
    endif;

    if Q20802 in 1 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 208 + ( $/1000);
        else
            i = 208.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q208_COICOP);
        impute(M_AMOUNT(tmp),       Q20806);
        // multiplier
        i = 1; // assume all expenditures are reported as annual amounts and adjust
        according to the expenditure code
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20807 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  Q20805 + 50);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
```

```
        M_DESCRIPTION(tmp) =        concat("[hh services exp]",tolower(strip(getlabel($,$))));
    endif;
endif;
PROC S2090_EDT
postproc
    if noccurs(S2090_EDT) > 0 then
        do varying i = noccurs(S2090_EDT) until i <= 0 by (-1)
            if Q20902(i)= 2 then
                delete(S2090_EDT(i));
                //errmsg("deleting non-expense records in section S2090") summary;
            endif;
        enddo;
    endif;

PROC Q20902
postproc
    if not $ in 1:2 then
        impute($, setNumFlag(getsymbol(),curocc(),4,2));
    endif;

PROC Q20903
if hh_ok then
    if Q20902 in 1 then
        if invalueset(Q20906) then
            putdeck(hd_20906_fr_tenure_expCode,Q20906);
        else
            impute(Q20906,
            setNumFlag("Q20906",curocc(),1,getdeck(hd_20906_fr_tenure_expCode)));
        endif;
    endif;
    recode ID07 : Q20901 : Q20903 => Q20905;
        141 : 8 : 108 => setNumFlag("Q20905",curocc(),1,1600);
    endrecode;
    // Lets add some default COICOP codes to the data
    if Q20902 in 1 then
        recode $ =>  Q209_COICOP;
            101 => 901100101; // birthday
            102 => 901100102; // funeral, weddings
            103 => 901100103; // fundraising
            104 => 901100104; // graduations
            105 => 901100105; // housewarming
            106 => 901100106; // traditional
            107 => 901100108; // special occations
            108 => 901100109; // other events
                => 999999999; // default for missng
        endrecode;
    endif;
    // transfer the data from the individual records to the agrigated record
    if Q20902 in 1 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 209 + ( $/1000);
        else
            i = 209.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q209_COICOP);
        impute(M_AMOUNT(tmp),       Q20905);
        // multiplier
        i = 1;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q20906 in 1 then
            i = 1;
        else
            i = 2;
        endif;
```

```
        impute(M_LOCATION(tmp),     i);
        impute(M_EXP_TYPE(tmp),     1);
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  51);
        M_DESCRIPTION(tmp) =        concat("[special occ exp]",tolower(strip(getlabel($,$))));
    endif;
endif;
PROC S2100_EDT
postproc
    if noccurs(S2100_EDT) > 0 then
        do varying i = noccurs(S2100_EDT) until i <= 0 by (-1)
            if Q21002(i)= 2 then
                delete(S2100_EDT(i));
                //errmsg("deleting non-expense records in section S2100") summary;
            endif;
        enddo;
    endif;

PROC Q21002
postproc
    if not $ in 1:2 then
        impute($, 2);
    endif;

PROC Q21003
if hh_ok then

    if Q21002 in 1 then
        if Q21006 in 1:3 then
            putdeck(hd_21006_fr_tenure,Q21006);
        else
            Q21006 = setNumFlag("Q21006",curocc(),3,getdeck(hd_21006_fr_tenure));
        endif;
    endif;



    // Lets add some default COICOP codes to the data
    if Q21002 in 1 then
        recode $ =>  Q210_COICOP;
            101 => 901100110; // donations to another HH
            102 => 901100301; // chruch: weekly
            103 => 901100302; // church: irregular
            104 => 901100601; // community
            105 => 901100602; // school
            106 => 901100603; // other groups
                => 999999999; // default for missng
        endrecode;
    endif;
    // transfer the data from the individual records to the main
    // recoded dataset.
    if Q21002 in 1 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 210 + ( $/1000);
        else
            i = 210.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q210_COICOP);
        impute(M_AMOUNT(tmp),       Q21005);
        // multiplier
        i = 1;
        impute(M_MULTIPLIER(tmp),   i); //change
```

```
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q21006 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_EXP_TYPE(tmp),     1);
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        recode Q21003 => i;
            101 => 51;
            102 => 53;
            103 => 53;
            104 => 54;
            105 => 55;
            106 => 56;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        M_DESCRIPTION(tmp) =        concat("[cash support
        exp]",tolower(strip(getlabel($,$))));
    endif;
endif;
PROC S2110_EDT
preproc
    if noccurs(S2110_EDT) > 0 then
        do varying i = noccurs(S2110_EDT) until i <= 0 by (-1)
            if Q21105(i) in notappl then
                delete(S2110_EDT(i));
                //errmsg("deleting blank records in section S2110") summary;
            endif;
        enddo;
    endif;
PROC S2120_EDT
postproc
    if noccurs(S2120_EDT) > 0 then
        do varying i = noccurs(S2120_EDT) until i <= 0 by (-1)
            if Q21202(i)= 2 then
                delete(S2120_EDT(i));
                //errmsg("deleting non-expense records in section S2120") summary;
            endif;
        enddo;
    endif;

PROC Q21202
postproc
    if not $ in 1:2 then
        impute($, 2);
    endif;

PROC Q21203
if hh_ok then
    // Lets add some default COICOP codes to the data
    if Q21202 in 1 then
        recode $ =>  Q212_COICOP;
            101 => 125229301; // Home insurance
            102 => 125429501; // Vehicle insurance
            103 => 125429502; // Boat insurance
            104 => 125429503; // Other insurance
            201 => 901100706; // Road use tax
            202 => 901100704; // Fines gor governmnet violations
            203 => 901100705; // Import tax
            204 => 901100706; // other government related fee, fines and/or taxes
                => 999999999; // default for missng
        endrecode;
    endif;
    // transfer the data from the individual records to the main
```

```
    // recoded dataset.
    if Q21202 in 1 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 212 + ( $/1000);
        else
            i = 212.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q212_COICOP);
        impute(M_AMOUNT(tmp),       Q21206);
        // multiplier
        i = 1;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q21207 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl);
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        impute(M_LOCAL(tmp),        notappl);
        impute(M_BENEFICIARY(tmp),  50 + Q21205);
        if M_BENEFICIARY(tmp) = 51 then
            i = 1;
        else
            i = 3;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[HH ins exp]",tolower(strip(getlabel($,$))));
    endif;
endif;
PROC S2130_EDT
preproc
    if noccurs(S2130_EDT) > 0 then
        do varying i = noccurs(S2130_EDT) until i <= 0 by (-1)
            if !Q21304(i)in 1:5 then
                delete(S2130_EDT(i));
                //errmsg("deleting non-expense records in section S2130") summary;
            endif;
        enddo;
    endif;

PROC Q21301
postproc
    impute($, curocc());
PROC Q21302
if curocc() in 1 then
    impute(Q21303, 101);
    Q21302 = getlabel(Q21304,Q21304);
endif;
if curocc() in 2 then
    impute(Q21303, 102);
    Q21302 = getlabel(Q21304,Q21304);
endif;
if curocc() in 3 then
    impute(Q21303, 103);
    Q21302 = getlabel(Q21304,Q21304);
endif;
if curocc() in 4 then
    impute(Q21303, 104);
    Q21302 = getlabel(Q21304,Q21304);
endif;
```

```
if curocc() in 5 then
    impute(Q21303, 105);
    Q21302 = getlabel(Q21304,Q21304);
endif;
PROC Q21303
if hh_ok then
    // Lets add some default COICOP codes to the data
    if Q21304 in 1:5 then
        recode Q21304 =>  Q213_COICOP;
            1 => 125329401; // Basic medical insurance
            2 => 125329401; // Supplementary medical insurance
            3 => 125329401; // Other medical insurance
            4 => 125129201; // Life insurance
            5 => 901100706; // Other personal insurance
                => 999999999; // default for missng
        endrecode;
    endif;
    // transfer the data from the individual records to the main
    // recoded dataset.
    if Q21304 in 1:5 then
        numeric tmpRatio = 0;
        // insurance is a bit different becasue we have to allocate a ratio of the amount to
        this HH and other HHs
        // so lets get the ratios first
        if Q21307 in 0, notappl then
            tmpRatio = 1;
        else
            tmpRatio = Q21306/(Q21306 + Q21307);
        endif;
        //this first set is for the main HH
            numeric tmp = noccurs(MODULES_EDT)+1;
            insert(MODULES_EDT(tmp));
            impute(M_LINE_NUMBER(tmp),  tmp);
            // HIES CODE
            i = 213 + ( Q21304/1000);
            impute(M_HIES_CODE(tmp),    i); //change
            impute(M_COICOP(tmp),       Q213_COICOP);
            impute(M_AMOUNT(tmp),       Q21305 * tmpRatio);
            // multiplier
            i = 1;
            impute(M_MULTIPLIER(tmp),   i); //change
            impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
            impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
            if Q21308 in 1 then
                i = 1;
            else
                i = 2;
            endif;
            impute(M_LOCATION(tmp),     i);
            impute(M_PROD_TYPE(tmp),    notappl);
            impute(M_QUANTITY(tmp),     notappl);
            impute(M_NUMBER(tmp),       notappl);
            impute(M_UNIT(tmp),         notappl);
            impute(M_LOCAL(tmp),        notappl);
            impute(M_BENEFICIARY(tmp),  51);
            impute(M_EXP_TYPE(tmp),     1);
            M_DESCRIPTION(tmp) =        concat("[Person ins
            exp]",tolower(strip(getlabel(Q21304,Q21304))));
        // now lets add the other outside members
        if tmpRatio < 1 then
            tmp = noccurs(MODULES_EDT)+1;
            insert(MODULES_EDT(tmp));
            impute(M_LINE_NUMBER(tmp),  tmp);
            // HIES CODE
            i = 213 + ( Q21304/1000);
            impute(M_HIES_CODE(tmp),    i); //change
            impute(M_COICOP(tmp),       Q213_COICOP);
            impute(M_AMOUNT(tmp),       Q21305 * (1 - tmpRatio));
            // multiplier
            i = 1;
            impute(M_MULTIPLIER(tmp),   i); //change
```

```
            impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
            impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
            if Q21308 in 1 then
                i = 1;
            else
                i = 2;
            endif;
            impute(M_LOCATION(tmp),     i);
            impute(M_PROD_TYPE(tmp),    notappl);
            impute(M_QUANTITY(tmp),     notappl);
            impute(M_NUMBER(tmp),       notappl);
            impute(M_UNIT(tmp),         notappl);
            impute(M_LOCAL(tmp),        notappl);
            impute(M_BENEFICIARY(tmp),  52);
            impute(M_EXP_TYPE(tmp),     3);
            M_DESCRIPTION(tmp) =        concat("[Person ins
            exp]",tolower(strip(getlabel(Q21304,Q21304))));
        endif;
    endif;
endif;
PROC Q21401_1
postproc
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_1;
    q21402(tmp) = Q21402_1;
    q21403(tmp) = Q21403_1;
    q21404(tmp) = Q21404_1;
    q21403_O(tmp) = Q21403_O_1;
    q21403_N(tmp) = Q21403_N_1;
    q21403_U(tmp) = Q21403_U_1;
    q21405(tmp) = Q21405_1;
    q21406(tmp) = Q21406_1;
    q21406_U(tmp) = Q21406_U_1;
    q21407(tmp) = Q21407_1;
    q21408(tmp) = Q21408_1;

PROC Q21401_2
postproc
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_2;
    q21402(tmp) = Q21402_2;
    q21403(tmp) = Q21403_2;
    q21404(tmp) = Q21404_2;
    q21403_O(tmp) = Q21403_O_2;
    q21403_N(tmp) = Q21403_N_2;
    q21403_U(tmp) = Q21403_U_2;
    q21405(tmp) = Q21405_2;
    q21406(tmp) = Q21406_2;
    q21406_U(tmp) = Q21406_U_2;
    q21407(tmp) = Q21407_2;
    q21408(tmp) = Q21408_2;
PROC Q21401_3A
postproc
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_3A;
    q21402(tmp) = Q21402_3A;
    q21403(tmp) = Q21403_3A;
    q21404(tmp) = Q21404_3A;
    q21403_O(tmp) = Q21403_O_3A;
    q21403_N(tmp) = Q21403_N_3A;
    q21403_U(tmp) = Q21403_U_3A;
    q21405(tmp) = Q21405_3A;
    q21406(tmp) = Q21406_3A;
    q21406_U(tmp) = Q21406_U_3A;
    q21407(tmp) = Q21407_3A;
    q21408(tmp) = Q21408_3A;
PROC Q21401_3B
postproc
```

```
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_3B;
    q21402(tmp) = Q21402_3B;
    q21403(tmp) = Q21403_3B;
    q21404(tmp) = Q21404_3B;
    q21403_O(tmp) = Q21403_O_3B;
    q21403_N(tmp) = Q21403_N_3B;
    q21403_U(tmp) = Q21403_U_3B;
    q21405(tmp) = Q21405_3B;
    q21406(tmp) = Q21406_3B;
    q21406_U(tmp) = Q21406_U_3B;
    q21407(tmp) = Q21407_3B;
    q21408(tmp) = Q21408_3B;
PROC Q21401_4
postproc
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_4;
    q21402(tmp) = Q21402_4;
    q21403(tmp) = Q21403_4;
    q21404(tmp) = Q21404_4;
    q21403_O(tmp) = Q21403_O_4;
    q21403_N(tmp) = Q21403_N_4;
    q21403_U(tmp) = Q21403_U_4;
    q21405(tmp) = Q21405_4;
    q21406(tmp) = Q21406_4;
    q21406_U(tmp) = Q21406_U_4;
    q21407(tmp) = Q21407_4;
    q21408(tmp) = Q21408_4;
PROC Q21401_5
postproc
    numeric tmp = noccurs(POWER_EDT)+1;
    insert(POWER_EDT(tmp));
    q21401(tmp) = Q21401_5;
    q21402(tmp) = Q21402_5;
    q21403(tmp) = Q21403_5;
    q21404(tmp) = Q21404_5;
    q21403_O(tmp) = Q21403_O_5;
    q21403_N(tmp) = Q21403_N_5;
    if tonumber(Q21403_U_5) in 1:6 then
        q21403_U(tmp) = tonumber(Q21403_U_5);
    else
        q21403_U(tmp) = notappl;
    endif;
    q21405(tmp) = Q21405_5;
    q21406(tmp) = Q21406_5;
    q21406_U(tmp) = Q21406_U_5;
    q21407(tmp) = Q21407_5;
    q21408(tmp) = Q21408_5;
PROC S3012_EDT
preproc
    if noccurs(S3012_EDT) > 0 then
        do varying i = noccurs(S3012_EDT) until i <= 0 by (-1)
            if length(strip(Q30153(i))) < 2 then
                delete(S3012_EDT(i));
                //errmsg("deleting blank records in section S3012") summary;
            endif;
        enddo;
    endif;

PROC Q30151
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30152
if hh_ok then
    if length(strip(Q30153)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
```

```
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 301 + ((Q30157*100 + $)/1000);
        else
            i = 301.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q301_COICOP);
        impute(M_AMOUNT(tmp),       Q30155);
        i = 1; // annual multiplier
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30156 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30154 => i;
            1-39 => Q30154;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[edu exp]",tolower(strip(Q30153)));
    endif;
endif;
PROC S3022_EDT
preproc
    if noccurs(S3022_EDT) > 0 then
        do varying i = noccurs(S3022_EDT) until i <= 0 by (-1)
            if length(strip(Q30253(i))) < 2 then
                delete(S3022_EDT(i));
                //errmsg("deleting blank records in section S3022") summary;
            endif;
        enddo;
    endif;
PROC Q30251
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30252
if hh_ok then
    if length(strip(Q30253)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 302 + ($/1000);
        else
            i = 302.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q302_COICOP);
        impute(M_AMOUNT(tmp),       Q30255);
```

```
        i = 1; // annual multiplier
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30256 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30254 => i;
            1-39 => Q30254;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[health exp]",tolower(strip(Q30253)));
    endif;
endif;
PROC S3032_EDT
preproc
    if noccurs(S3032_EDT) > 0 then
        do varying i = noccurs(S3032_EDT) until i <= 0 by (-1)
            if length(strip(Q30353(i))) < 1 then
                delete(S3032_EDT(i));
                //errmsg("deleting blank records in section S3032") summary;
            endif;
        enddo;
    endif;
PROC Q30351
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30352
if hh_ok then

    recode ID07 : Q30351 => Q30355;
        327 : 4 => setNumFlag({questName}"Q30355",{lineNo}curocc(),{imputeType}1,{newVal}8);
    endrecode;
    recode ID07 : Q30351 => Q30354;
        143 : 1 => setNumFlag({questName}"Q30354",{lineNo}curocc(),{imputeType}1,{newVal}1);
        143 : 2 => setNumFlag({questName}"Q30354",{lineNo}curocc(),{imputeType}1,{newVal}2);
    endrecode;


    if Q30355 > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 303 + ($/1000);
        else
            i = 303.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q303_COICOP);
        impute(M_AMOUNT(tmp),       Q30355);
```

```
        i = 4; // annual multiplier
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30356 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30354 => i;
            1-39 => Q30354;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[clothing exp]",tolower(strip(Q30353)));
    endif;
endif;
PROC S3042_EDT
preproc
    if noccurs(S3042_EDT) > 0 then
        do varying i = noccurs(S3042_EDT) until i <= 0 by (-1)
            if length(strip(Q30453(i))) < 1 then
                delete(S3042_EDT(i));
                //errmsg("deleting blank records in section S3042") summary;
            endif;
        enddo;
    endif;
PROC Q30451
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30452
if hh_ok then

    if length(strip(Q30453)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 304 + ($/1000);
        else
            i = 304.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q304_COICOP);
        impute(M_AMOUNT(tmp),       Q30455);
        if $ in 1:3 then
            i = 12;
        else
            i = 1;
        endif;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30456 in 1 then
```

```
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30454 => i;
            1-39 => Q30454;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[com exp]",tolower(strip(Q30453)));
    endif;
endif;
PROC S3052_EDT
preproc
    if noccurs(S3052_EDT) > 0 then
        do varying i = noccurs(S3052_EDT) until i <= 0 by (-1)
            if length(strip(Q30553(i))) < 2 then
                delete(S3052_EDT(i));
                //errmsg("deleting blank records in section S3052") summary;
            endif;
        enddo;
    endif;
PROC Q30551
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30552
if hh_ok then
    if length(strip(Q30553)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 305 + ($/1000);
        else
            i = 305.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q305_COICOP);
        impute(M_AMOUNT(tmp),       Q30555);
        if $ in 1:3 then
            i = 4;
        else
            i = 1;
        endif;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30556 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
```

```
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30554 => i;
            1-39 => Q30554;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[luxury exp]",tolower(strip(Q30553)));
    endif;
endif;
PROC S3061_EDT
preproc
    if noccurs($) < 17 then
        do varying i = noccurs($)+1 until i >= 17
            insert($(i));
        enddo;
    endif;

    do varying i = 1 until i > totocc($)
        if i in 1:15 then
            Q30601(i) = i;
        endif;
        if i in 16 then
            Q30601(i) = 60;
        endif;
        if i in 17 then
            Q30601(i) = 90;
        endif;
    enddo;


    if noccurs($) = 0 then
        errmsg(" THIS IS A BLANK ALC and TOBAC RECORD");
    endif;

PROC Q30602
numeric sexRcd,ageRcd, relateRcd;
if curocc() in 1:noccurs(S1010_EDT) then
    sexRcd = sex(curocc());
    relateRcd = relate(curocc());
    recode age(curocc()) => ageRcd ;
        0-9 => 1;
        10-19 => 2;
        20-29 => 3;
        30-39 => 4;
        40-49 => 5;
        50-59 => 6;
        60-69 => 7;
        70-79 => 8;
        80-89 => 9;
        90-99 => 10;
    endrecode;
endif;
if curocc() in 1:noccurs(S1010_EDT) then
    if age(curocc()) in 10:99 and sex(curocc()) in 1:2 then
        if $ in 1:2 then
            hd_30602_fr_sex_relate_age(sexRcd, ageRcd, relateRcd) = $;
        else
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}3,{newVal}hd_30602
            _fr_sex_relate_age(sexRcd, ageRcd, relateRcd));
```

```
            //errmsg("%d", $);
        endif;
        if special($) then
            $ = setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
    else
        if $ <> notappl then
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl)
            ;
        endif;
    endif;
else
    if $ <> notappl then
        $ =
        setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl);
    endif;
endif;

PROC Q30603
numeric sexRcd,ageRcd;
if curocc() in 1:noccurs(S1010_EDT) then
    sexRcd = sex(curocc());
    recode age(curocc()) => ageRcd ;
        0-9 => 1;
        10-19 => 2;
        20-29 => 3;
        30-39 => 4;
        40-49 => 5;
        50-59 => 6;
        60-69 => 7;
        70-79 => 8;
        80-89 => 9;
        90-99 => 10;
    endrecode;
endif;
if curocc() in 1:noccurs(S1010_EDT) then
    if age(curocc()) in 10:99 and sex(curocc()) in 1:2 then
        if $ in 1:2 then
            hd_30603_fr_sex_age_drink(sexRcd, ageRcd, Q30602) = $;
        else
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}3,{newVal}hd_30603
            _fr_sex_age_drink(sexRcd, ageRcd, Q30602));
            //errmsg("%d", $);
        endif;
        if special($) then
            $ = setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
    else
        if $ <> notappl then
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl)
            ;
        endif;
    endif;
else
    if $ <> notappl then
        $ =
        setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl);
    endif;
endif;

PROC Q30604
// this is a mistake - they put non-smoker but expenditure data shows otherwise
if ID07 = 234 and Q30601 = 1 and Q30604 = 2 then
    $ = setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}1);
endif;
numeric sexRcd,ageRcd;
if curocc() in 1:noccurs(S1010_EDT) then
    sexRcd = sex(curocc());
```

```
    recode age(curocc()) => ageRcd ;
        0-9 => 1;
        10-19 => 2;
        20-29 => 3;
        30-39 => 4;
        40-49 => 5;
        50-59 => 6;
        60-69 => 7;
        70-79 => 8;
        80-89 => 9;
        90-99 => 10;
    endrecode;
endif;
if curocc() in 1:noccurs(S1010_EDT) then
    if age(curocc()) in 10:99 and sex(curocc()) in 1:2 then
        if Q30605 in 1:500 then
            if $ <> 1 then
                $ =
                setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}1);
            endif;
        endif;
        if $ in 1:2 then
            hd_30604_fr_sex_age_drink(sexRcd, ageRcd, Q30602) =  $;
        else
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}3,{newVal}hd_30604
            _fr_sex_age_drink(sexRcd, ageRcd, Q30602));
            //errmsg("%d", $);
        endif;
        if special($) then
            $ = setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
    else
        if $ <> notappl then
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl)
            ;
        endif;
    endif;
else
    if $ <> notappl then
        $ =
        setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl);
    endif;
endif;
PROC Q30605
numeric sexRcd,ageRcd;
if curocc() in 1:noccurs(S1010_EDT) then
    sexRcd = sex(curocc());
    recode age(curocc()) => ageRcd ;
        0-9 => 1;
        10-19 => 2;
        20-29 => 3;
        30-39 => 4;
        40-49 => 5;
        50-59 => 6;
        60-69 => 7;
        70-79 => 8;
        80-89 => 9;
        90-99 => 10;
    endrecode;
endif;
if curocc() in 1:noccurs(S1010_EDT) then
    if Q30604 in 1 then
        if $ in 1:500 then
            hd_30605_fr_sex_age_drink(sexRcd, ageRcd, Q30602) = $
        else
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}3,{newVal}hd_30605
            _fr_sex_age_drink(sexRcd, ageRcd, Q30602));
        endif;
```

```
        if special($) then
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}4,{newVal}40);
            /// 40 median number of cigarettes
        endif;
    else
        if $ <> notappl then
            $ =
            setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}4,{newVal}notappl)
            ;
        endif;
    endif;
else
    if $ <> notappl then
        $ =
        setNumFlag({questName}getsymbol($),{lineNo}curocc(),{imputeType}1,{newVal}notappl);
    endif;
endif;
PROC S3062_EDT
preproc
    if noccurs(S3062_EDT) > 0 then
        do varying i = noccurs(S3062_EDT) until i <= 0 by (-1)
            if length(strip(Q30653(i))) < 2 then
                delete(S3062_EDT(i));
                //errmsg("deleting blank records in section S3062") summary;
            endif;
        enddo;
    endif;


PROC Q30651
postproc
    if $ <> curocc() then
        impute($, curocc());
    endif;
PROC Q30652
if hh_ok then
    // lets start with some maual fixed from Iapi on 11 apr 2016
    recode ID07 : Q30651 => Q30655;
        106 : 1 => setNumFlag({questName}"Q30655",{lineNo}curocc(),{imputeType}1,{newVal}48);
        342 : 1 => setNumFlag({questName}"Q30655",{lineNo}curocc(),{imputeType}1,{newVal}72);
        342 : 2 => setNumFlag({questName}"Q30655",{lineNo}curocc(),{imputeType}1,{newVal}35);
    endrecode;

    recode ID07 : Q30651 => Q30654;
        143 : 1 => setNumFlag({questName}"Q30654",{lineNo}curocc(),{imputeType}1,{newVal}1);
        143 : 2 => setNumFlag({questName}"Q30654",{lineNo}curocc(),{imputeType}1,{newVal}2);
        143 : 3 => setNumFlag({questName}"Q30654",{lineNo}curocc(),{imputeType}1,{newVal}2);
    endrecode;

    recode ID07 : Q30651 => Q30657;
        138 :   1   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        138 :   2   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        138 :   3   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        138 :   4   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        150 :   7   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        159 :   2   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
        160 :   3   =>
        setNumFlag({questName}"Q30657",{lineNo}curocc(),{imputeType}1,{newVal}1);
    endrecode;


    if length(strip(Q30653)) > 0 then
        numeric tmp = noccurs(MODULES_EDT)+1;
        insert(MODULES_EDT(tmp));
```

```
        impute(M_LINE_NUMBER(tmp),  tmp);

        if invalueset($) and $ <> notappl then
            i = 306 + ($/1000);
        else
            i = 306.999;
        endif;
        impute(M_HIES_CODE(tmp),    i); //change
        impute(M_COICOP(tmp),       Q306_COICOP);
        impute(M_AMOUNT(tmp),       Q30655);
        if $ in 1:6 then
            i = 52;
        else
            i = 1;
        endif;
        impute(M_MULTIPLIER(tmp),   i); //change
        impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
        impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
        if Q30657 in 1 then
            i = 1;
        else
            i = 2;
        endif;
        impute(M_LOCATION(tmp),     i);
        impute(M_PROD_TYPE(tmp),    notappl); //tmp code for purchase
        impute(M_QUANTITY(tmp),     notappl);
        impute(M_NUMBER(tmp),       notappl);
        impute(M_UNIT(tmp),         notappl);
        recode Q30654 => i;
            1-39 => Q30654;
            40-51 => 51;
            60-69 => 57;
            90-98 => 52;
                  => notappl;
        endrecode;
        impute(M_BENEFICIARY(tmp),  i);
        if M_BENEFICIARY(tmp) in 52 then
            i = 3;
        else
            i = 1;
        endif;
        impute(M_EXP_TYPE(tmp),     i);
        M_DESCRIPTION(tmp) =        concat("[alc & tobac exp]",tolower(strip(Q30653)));
    endif;
endif;
PROC S4011_EDT
preproc
    if noccurs(S4011_EDT) > 0 then
        do varying i = noccurs(S4011_EDT) until i <= 0 by (-1)
            if length(strip(Q40103(i))) < 2 then
                delete(S4011_EDT(i));
                //errmsg("deleting blank records in section S4011") summary;
            endif;
        enddo;
    endif;

    if noccurs(S4011_EDT) >= 1 then Q40101(1) = "1"; endif;
    if noccurs(S4011_EDT) >= 2 then Q40101(2) = "2"; endif;
    if noccurs(S4011_EDT) >= 3 then Q40101(3) = "3"; endif;
    if noccurs(S4011_EDT) >= 4 then Q40101(4) = "4"; endif;
    if noccurs(S4011_EDT) >= 5 then Q40101(5) = "5"; endif;
    if noccurs(S4011_EDT) >= 6 then Q40101(6) = "6"; endif;
    if noccurs(S4011_EDT) >= 7 then Q40101(7) = "7"; endif;
    if noccurs(S4011_EDT) >= 8 then Q40101(8) = "8"; endif;

    // deleting volunteer and domestic duties from income - they make no money
    string c;
    if ID07 in 328 and Q40102(3) = 4 then
        c = setStrFlag({questName}"Q40103",{lineNo}3,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}3,{imputeType}8,{newVal}"");
        delete($(3));
```

```
    endif;
    if ID07 in 350 and Q40102(4) = 2 then
        c = setStrFlag({questName}"Q40103",{lineNo}4,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}4,{imputeType}8,{newVal}"");
        delete($(4));
    endif;
    if ID07 in 350 and Q40102(3) = 1 then
        c = setStrFlag({questName}"Q40103",{lineNo}3,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}3,{imputeType}8,{newVal}"");
        delete($(3));
    endif;
    if ID07 in 351 and Q40102(2) = 1 then
        c = setStrFlag({questName}"Q40103",{lineNo}2,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}2,{imputeType}8,{newVal}"");
        delete($(2));
    endif;
    if ID07 in 356 and Q40102(2) = 1 then
        c = setStrFlag({questName}"Q40103",{lineNo}2,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}2,{imputeType}8,{newVal}"");
        delete($(2));
    endif;
    if ID07 in 104 and Q40102(1) = 3 then
        c = setStrFlag({questName}"Q40103",{lineNo}1,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40104",{lineNo}1,{imputeType}8,{newVal}"");
        delete($(1));
    endif;

PROC Q40101
recode  ID07    :   tonumber(Q40101)    =>  Q401_A_ISIC ;
    116 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    118 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    121 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    127 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    135 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    150 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    155 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    217 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    249 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    253 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    255 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    319 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    320 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    322 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    324 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    342 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    224 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    322 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    354 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    127 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    117 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
```

```
332 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
105 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
342 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
103 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
347 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
127 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
119 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
346 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}551) ;
342 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}551) ;
141 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
148 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
148 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
111 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
241 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
341 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
154 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
125 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
116 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
233 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
109 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
109 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
134 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
201 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
206 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
210 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
233 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
236 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
236 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
319 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
324 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
329 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
113 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
127 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
136 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
139 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
208 :   2   =>
```

```
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
209 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
241 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
253 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
307 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
352 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
160 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
160 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
239 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
224 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
343 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
219 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
151 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}852) ;
340 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}949) ;
223 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
125 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
126 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
320 :   5   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
349 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
219 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
323 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
343 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
350 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
150 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}861) ;
339 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
221 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
250 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
327 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
332 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
347 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
203 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
224 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
154 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
217 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
158 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
239 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
```

```
354 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
147 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
328 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
134 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
326 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
221 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
102 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
322 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
134 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
356 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
340 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
318 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
343 :   7   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
151 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
327 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
339 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
135 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
141 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
108 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
108 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
118 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
121 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
210 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
240 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
249 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
306 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
310 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
132 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}842) ;
237 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
159 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
301 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
126 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
124 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
204 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
226 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
239 :   1   =>
```

```
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
241 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
304 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
327 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
327 :   5   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
349 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
351 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
353 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}410) ;
124 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
258 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
160 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
227 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
252 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
343 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
343 :   5   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
343 :   6   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
353 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
301 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
304 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
305 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
306 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
309 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
309 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
320 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
322 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
326 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
326 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
327 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
332 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
217 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
217 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
218 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
218 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
218 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
218 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
219 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
```

```
220 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
223 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
226 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
227 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
235 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
246 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
250 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
258 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}131) ;
155 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
249 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
203 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
204 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
253 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
346 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
119 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
139 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
206 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}551) ;
258 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}551) ;
336 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
326 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
318 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
340 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}949) ;
338 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}970) ;
140 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}970) ;
223 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}551) ;
338 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
346 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
121 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
121 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
124 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
142 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
209 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
219 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
234 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
239 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
256 :   1   =>
```

```
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
346 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
213 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471) ;
234 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
234 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
238 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
240 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
348 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
213 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
207 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
208 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
208 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
211 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
227 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
232 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
234 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
236 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
237 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
238 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
254 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
319 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
323 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
150 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
211 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
132 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
302 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
306 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
306 :   4   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
308 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
309 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
309 :   3   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
311 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
311 :   2   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
328 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
333 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
339 :   1   =>
setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
```

```
    343 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    320 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    320 :   4   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    119 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    102 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    103 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    106 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    106 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    108 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    108 :   4   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    109 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    114 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    218 :   5   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    218 :   6   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    222 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    235 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    238 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    238 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    252 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    254 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    337 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    340 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    350 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    352 :   2   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    234 :   5   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    309 :   5   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    135 :   3   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    138 :   1   =>
    setNumFlag({questName}'Q401_A_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
endrecode;
recode  ID07    :   tonumber(Q40101)    =>  Q401_A_ISCO ;
    102 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}412) ;
    102 :   2   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
    103 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}132) ;
    103 :   2   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
    105 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}121) ;
    106 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
    106 :   2   =>
```

```
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
108 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
108 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
108 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
108 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
109 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
109 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
109 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
111 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}221) ;
113 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
114 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
116 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}226) ;
116 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
117 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
118 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
118 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
119 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}134) ;
119 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}833) ;
119 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
121 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
121 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
121 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
121 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}723) ;
124 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
124 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}712) ;
124 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
125 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
125 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}226) ;
126 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
126 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
127 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
127 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
127 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
127 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}134) ;
132 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
132 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
```

```
134 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}422) ;
134 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}351) ;
134 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
135 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}531) ;
135 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
135 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
136 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
138 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
139 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
139 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}833) ;
140 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
141 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}214) ;
141 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}531) ;
142 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
147 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}341) ;
148 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}214) ;
148 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}214) ;
150 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
150 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
150 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
151 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
151 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}262) ;
154 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}222) ;
154 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
155 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
155 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
158 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
159 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
160 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}235) ;
160 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}235) ;
160 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}723) ;
201 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
203 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
203 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
204 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
204 :   2   =>
```

```
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
206 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
206 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
207 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
208 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
208 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
208 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
209 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
209 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
210 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
210 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
211 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
211 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
213 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
213 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
217 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
217 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
217 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
217 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
218 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
218 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
218 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
218 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
218 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
218 :   6   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
219 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
219 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}261) ;
219 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
219 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
220 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
221 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
221 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}412) ;
222 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
223 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}315) ;
223 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
223 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
```

```
224 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}242) ;
224 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
224 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
226 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
226 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
227 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
227 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
227 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}723) ;
232 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
233 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
233 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}231) ;
234 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
234 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
234 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
234 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
234 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
235 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
235 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
236 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
236 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
236 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
237 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
237 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
238 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
238 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
238 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
238 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
239 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
239 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}241) ;
239 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
239 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
240 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
240 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
241 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
241 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
241 :   3   =>
```

```
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}222) ;
246 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
249 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
249 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
249 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
250 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
250 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
252 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
252 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
253 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
253 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
253 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
254 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
254 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
255 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
256 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
258 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}712) ;
258 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
258 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
301 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
301 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
302 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
304 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
304 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
305 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
306 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
306 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
306 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
306 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
307 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
308 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
309 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
309 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
309 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
309 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
309 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
```

```
310 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}541) ;
311 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
311 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
318 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
318 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}512) ;
319 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
319 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
319 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
320 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
320 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
320 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
320 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
320 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
322 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
322 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
322 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
322 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}422) ;
323 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
323 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
324 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
324 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
326 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
326 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
326 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
326 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}352) ;
327 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}531) ;
327 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
327 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
327 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
327 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
328 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
328 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}341) ;
329 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}233) ;
332 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
332 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
332 :   3   =>
```

```
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
333 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
336 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
337 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
338 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}921) ;
338 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
339 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
339 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
339 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}531) ;
340 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}263) ;
340 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}911) ;
340 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
340 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}512) ;
341 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}222) ;
342 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}122) ;
342 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111) ;
342 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}141) ;
343 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
343 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
343 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
343 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}252) ;
343 :   5   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
343 :   6   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
343 :   7   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}512) ;
346 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}741) ;
346 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}921) ;
346 :   3   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522) ;
346 :   4   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}141) ;
347 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}335) ;
347 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}132) ;
348 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
349 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
349 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
350 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
350 :   2   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}322) ;
351 :   1   =>
setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
```

```
    352 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}234) ;
    352 :   2   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}962) ;
    353 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}711) ;
    353 :   2   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}731) ;
    354 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
    354 :   2   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}341) ;
    356 :   1   =>
    setNumFlag({questName}'Q401_A_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}422) ;
endrecode;
if id07 in 114 and Q40102 in 1 then
    Q40102 = setNumFlag({questName}'Q40102',{lineNo}curocc(),{imputeType}1,{newVal}3);
endif;
PROC Q40102
if ID07 in 118 then
    if $ in notappl then
        impute ($,1);
    endif;
endif;
PROC Q40105
recode  ID07    :   tonumber(Q40101)    =>  Q40105  ;
    104 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}3)    ;
    147 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    148 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    148 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    150 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    150 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    150 :   3   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    151 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    151 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    154 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    154 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    155 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    155 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    158 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    159 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    160 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    160 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    160 :   3   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    301 :   1   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    301 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}1)    ;
    351 :   2   =>
    setNumFlag({questName}'Q40105',{lineNo}curocc(),{imputeType}1,{newVal}2)    ;
endrecode;
PROC Q40106
if Q40106 in 0 then Q40106 = notappl; endif;
```

```
if Q40107 in 0 then Q40107 = notappl; endif;
if Q40108 in 0 then Q40108 = notappl; endif;
if Q40109 in 0 then Q40109 = notappl; endif;
if Q40110 in 0 then Q40110 = notappl; endif;
if Q40106 in notappl, default then
    if Q40110 >= 1 then
        Q40106 = setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}1,{newVal}
        Q40110);
    endif;
endif;

set behavior() specialvalues(zero) on;
    if Q40106 + Q40107 + Q40108  + Q40109 <> Q40110 then
        if Q40107 + Q40108 + Q40109 = 0 then
            // find the biggest value between 40106 and 40110 then imput that value for both
            if Q40106 > 0 and Q40110 > 0 then
                if Q40110 > Q40106 then
                    Q40106 =
                    setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}2,{newVal}
                    Q40110);
                endif;
                if Q40110 < Q40106 then
                    Q40110 =
                    setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}2,{newVal}
                    Q40106);
                endif;
            endif;
        endif;
        // some of the values reported for wages is reported as forthnightly lets annualize.
        // imputation type is manual
        if  Q40110 > 0 and Q40106 > 0 then
            if Q40106 * 26 + Q40107 + Q40108 + Q40109 = Q40110 then
                Q40106 =
                setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}1,{newVal}
                Q40110-(Q40107+Q40108+Q40109));
            endif;
        endif;
        //lets fix some the reported wages to reflect the total but these are ones that have
        other incomes as well
        // impuation type is subjective
        if Q40106 + Q40107 + Q40108  + Q40109 <> Q40110 then
            if Q40107 + Q40108 + Q40109 > 0 then
                if Q40110 > 0 then
                    Q40106 =
                    setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}2,{newVal}
                    Q40110-(Q40107+Q40108+Q40109));
                endif;
            endif;
        endif;
        // this is the residual error where 40106 has a value and the rest have no value.
        if Q40106 + Q40107 + Q40108  + Q40109 <> Q40110 then
            if Q40107 + Q40108 + Q40109 + Q40110 = 0 then
                Q40106 =
                setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}2,{newVal}
                Q40106 * 26);
                Q40110 =
                setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}2,{newVal}
                Q40106);
            endif;
        endif;
        //this is for null values across all four fields
        if ID07 = 160 and tonumber(Q40101) = 1 and Q40106 = 0 then
            Q40106 = setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}1,{newVal}
            234 * 26); // this values was provided by Iapi
            Q40110 = setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}2,{newVal}
            Q40106);
        endif;
        if ID07 = 160 and tonumber(Q40101) = 2 and Q40106 = 0 then
            Q40106 = setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}1,{newVal}
            234 * 26); // this values was provided by Iapi
            Q40110 = setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}2,{newVal}
```

```
            Q40106);
        endif;
    endif;
    if Q40106 >= 1 then
        putdeck(hd_40106_fr_occ,Q40106);
    else
        Q40106 =
        setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}3,{newVal}getdeck(hd_40106
        _fr_occ));
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}3,{newVal}Q40106);

    endif;
    // there is only one libarian that was sampled so no wage match was possible on the
    hotdecks - so we used the median of teacher wages to impute
    if ID07 in 151 and tonumber(Q40101) = 2 then
        Q40106 =
        setNumFlag({questName}"Q40106",{lineNo}curocc(),{imputeType}4,{newVal}9800); // this
        is derived from the mean of teachers wages
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}4,{newVal}9800); // ISCO
        233,234 and 235
    endif;

    // Let fix some households that failed to disclose some in kind housing
    if ID07 in 111 and tonumber(Q40101) = 1 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}5400); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}40400); //
        adjusted teh total from the imputation
    endif;
    if ID07 in 117 and tonumber(Q40101) = 1 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}2400); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}42400); //
        adjusted teh total from the imputation
    endif;
    if ID07 in 210 and tonumber(Q40101) = 2 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}6720); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}16158); //
        adjusted teh total from the imputation
    endif;
    if ID07 in 233 and tonumber(Q40101) = 1 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}9300); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}32700); //
        adjusted teh total from the imputation
    endif;
    if ID07 in 307 and tonumber(Q40101) = 1 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}8400); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}21920); //
        adjusted teh total from the imputation
    endif;
    if ID07 in 341 and tonumber(Q40101) = 1 then
        Q40109 =
        setNumFlag({questName}"Q40109",{lineNo}curocc(),{imputeType}1,{newVal}9600); //
        imputed this value from reported tenure Q20218
        Q40110 =
        setNumFlag({questName}"Q40110",{lineNo}curocc(),{imputeType}1,{newVal}27600); //
```

```
        adjusted teh total from the imputation
    endif;


set behavior() specialvalues(zero) off;

// set paccoi
if $ > 0 then
    Q401_A_1_PACCOI = 1101;
endif;


PROC Q40107
// set paccoi
if $ > 0 then
    Q401_A_2_PACCOI = 1102;
endif;
PROC Q40108
// set paccoi
if $ > 0 then
    Q401_A_3_PACCOI = 1103;
endif;
PROC Q40109
// set paccoi
if $ > 0 then
    Q401_A_4_PACCOI = 1298;
endif;
PROC Q401_A_1_PACCOI
numeric tmp1 = 0;
if Q40106 > 0 then
    recode int(Q401_A_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
        10-33 => 5;
        35-35 => 6;
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_A_1_PACCOI, 11100+tmp1);
        impute (Q401_A_1_HIES_CODE, 401 + (100+int(Q401_A_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40104));
    endif;
endif;

if Q40107 > 0 then
    recode int(Q401_A_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
        10-33 => 5;
```

```
        35-35 => 6;
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_A_2_PACCOI, 11200+tmp1);
        impute (Q401_A_2_HIES_CODE, 401 + (200+int(Q401_A_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40104));
    endif;
endif;

if Q40108 > 0 then
    recode int(Q401_A_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
        10-33 => 5;
        35-35 => 6;
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_A_3_PACCOI, 11300+tmp1);
        impute (Q401_A_3_HIES_CODE, 401 + (300+int(Q401_A_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40104));
    endif;
endif;

if Q40109 > 0 then
    recode int(Q401_A_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
        10-33 => 5;
        35-35 => 6;
```

```
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_A_4_PACCOI, 11400+tmp1);
        impute (Q401_A_4_HIES_CODE, 401 + (400+int(Q401_A_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40104));
    endif;
endif;

PROC Q401_A_MULTI
if Q40106 in 1:99000 then
    $ = 1;
    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_A_1_HIES_CODE;
    I_PACCOI(tmp) = Q401_A_1_PACCOI;
    I_GROSS(tmp) = Q40106;
    I_MULTIPLIER(tmp) = Q401_A_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 1;
    I_INCOME_TYPE(tmp) = 1;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_A_ISIC;
    I_OCCUPATION(tmp) = Q401_A_ISCO;
    impute(I_BENEFICIARY(tmp), Q40102);
    I_DESCRIPTION(tmp) = concat("[primary cash
    wage]",tolower(strip(Q40103)),":",tolower(strip(Q40104)));
endif;
if Q40107 in 1:99000 then
    $ = 1;
    tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_A_2_HIES_CODE;
    I_PACCOI(tmp) = Q401_A_2_PACCOI;
    I_GROSS(tmp) = Q40107;
    I_MULTIPLIER(tmp) = Q401_A_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 1;
    I_INCOME_TYPE(tmp) = 1;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_A_ISIC;
    I_OCCUPATION(tmp) = Q401_A_ISCO;
    impute(I_BENEFICIARY(tmp), Q40102);
```

```
    I_DESCRIPTION(tmp) = concat("[primary cash
    overtime]",tolower(strip(Q40103)),":",tolower(strip(Q40104)));
endif;
if Q40108 in 1:99000 then
    $ = 1;
    tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_A_3_HIES_CODE;
    I_PACCOI(tmp) = Q401_A_3_PACCOI;
    I_GROSS(tmp) = Q40108;
    I_MULTIPLIER(tmp) = Q401_A_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 1;
    I_INCOME_TYPE(tmp) = 1;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_A_ISIC;
    I_OCCUPATION(tmp) = Q401_A_ISCO;
    impute(I_BENEFICIARY(tmp), Q40102);
    I_DESCRIPTION(tmp) = concat("[primary cash
    allowance]",tolower(strip(Q40103)),":",tolower(strip(Q40104)));
endif;

//becasue this is in-kind income we have to replicate this both in the income and
expenditure record
// income side
if Q40109 in 1:99000 then
    $ = 1;
    tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_A_4_HIES_CODE;
    I_PACCOI(tmp) = Q401_A_4_PACCOI;
    I_GROSS(tmp) = Q40109;
    I_MULTIPLIER(tmp) = Q401_A_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 1;
    I_INCOME_TYPE(tmp) = 7;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_A_ISIC;
    I_OCCUPATION(tmp) = Q401_A_ISCO;
    impute(I_BENEFICIARY(tmp), Q40102);
    I_DESCRIPTION(tmp) = concat("[primary in-kind
    wage]",tolower(strip(Q40103)),":",tolower(strip(Q40104)));
// expenditure side
    tmp = noccurs(MODULES_EDT)+1;
    insert(MODULES_EDT(tmp));
    impute(M_LINE_NUMBER(tmp),  tmp);
    impute(M_HIES_CODE(tmp),    Q401_A_4_HIES_CODE); //change
    impute(M_COICOP(tmp),       908108001);
    impute(M_AMOUNT(tmp),       Q40109);
    impute(M_MULTIPLIER(tmp),   1); //change
    impute(M_AN_AMT(tmp),       M_AMOUNT(tmp) * M_MULTIPLIER(tmp));
    impute(M_WGHT_AN_AMT(tmp),M_AN_AMT(tmp) * weight);
    impute(M_LOCATION(tmp),     1);
    impute(M_EXP_TYPE(tmp),     7); // 5= imputed rent; 7 = in-kind income
    impute(M_PROD_TYPE(tmp),    notappl);
    impute(M_QUANTITY(tmp),     notappl);
    impute(M_NUMBER(tmp),       notappl);
    impute(M_UNIT(tmp),         notappl);
    impute(M_LOCAL(tmp),        notappl);
    impute(M_BENEFICIARY(tmp),  51);
    M_DESCRIPTION(tmp) =        concat("[primary in-kind
```

```
    wage]",tolower(strip(Q40103)),":",tolower(strip(Q40104)));
endif;
PROC S4012_EDT
preproc
    if noccurs(S4012_EDT) > 0 then
        do varying i = noccurs(S4012_EDT) until i <= 0 by (-1)
            if length(strip(Q40124(i))) < 2 then
                delete(S4012_EDT(i));
                //errmsg("deleting blank records in section S4012") summary;
            endif;
        enddo;
    endif;

    if noccurs(S4012_EDT) >= 1 then Q40121(1) = "1"; endif;
    if noccurs(S4012_EDT) >= 2 then Q40121(2) = "2"; endif;
    if noccurs(S4012_EDT) >= 3 then Q40121(3) = "3"; endif;
    if noccurs(S4012_EDT) >= 4 then Q40121(4) = "4"; endif;
    if noccurs(S4012_EDT) >= 5 then Q40121(5) = "5"; endif;
    if noccurs(S4012_EDT) >= 6 then Q40121(6) = "6"; endif;
    if noccurs(S4012_EDT) >= 7 then Q40121(7) = "7"; endif;
    if noccurs(S4012_EDT) >= 8 then Q40121(8) = "8"; endif;

    // these households reported children allowance in wages so we are moving it to the
    household's "other transfer income" and deleting them from this record
    string c;
    if ID07 = 232 then
        c = setStrFlag({questName}"Q40123",{lineNo}2,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}2,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}2,{imputeType}8,{newVal}notappl);
        c = setStrFlag({questName}"Q40123",{lineNo}1,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}1,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}1,{imputeType}8,{newVal}notappl);
        delete(S4012_EDT(2));
        delete(S4012_EDT(1));
        Q40706(11) = setNumFlag({questName}"Q40706",{lineNo}11,{imputeType}1,{newVal}960);
        // 11 is the line number for other income sum of reported in wages
    endif;

    if ID07 = 233 then
        c = setStrFlag({questName}"Q40123",{lineNo}1,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}1,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}1,{imputeType}8,{newVal}notappl);
        delete(S4012_EDT(1));
        Q40706(11) = setNumFlag({questName}"Q40706",{lineNo}11,{imputeType}1,{newVal}1440);
        // 11 is the line number for other income sum of reported in wages
    endif;

    if ID07 = 236 then
        c = setStrFlag({questName}"Q40123",{lineNo}4,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}4,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}4,{imputeType}8,{newVal}notappl);

        c = setStrFlag({questName}"Q40123",{lineNo}3,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}3,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}3,{imputeType}8,{newVal}notappl);

        c = setStrFlag({questName}"Q40123",{lineNo}2,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}2,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}2,{imputeType}8,{newVal}notappl);

        c = setStrFlag({questName}"Q40123",{lineNo}1,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}1,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}1,{imputeType}8,{newVal}notappl);

        delete(S4012_EDT(4));
        delete(S4012_EDT(3));
        delete(S4012_EDT(2));
        delete(S4012_EDT(1));
        Q40706(11) = setNumFlag({questName}"Q40706",{lineNo}11,{imputeType}1,{newVal}4320);
        // 11 is the line number for other income sum of reported in wages
    endif;
```

```
    if ID07 = 241 then
        c = setStrFlag({questName}"Q40123",{lineNo}3,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}3,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}3,{imputeType}8,{newVal}notappl);

        c = setStrFlag({questName}"Q40123",{lineNo}2,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}2,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}2,{imputeType}8,{newVal}notappl);

        c = setStrFlag({questName}"Q40123",{lineNo}1,{imputeType}8,{newVal}"");
        c = setStrFlag({questName}"Q40124",{lineNo}1,{imputeType}8,{newVal}"");
        i = setNumFlag({questName}"Q40125",{lineNo}1,{imputeType}8,{newVal}notappl);

        delete(S4012_EDT(3));
        delete(S4012_EDT(2));
        delete(S4012_EDT(1));
        Q40706(11) = setNumFlag({questName}"Q40706",{lineNo}11,{imputeType}1,{newVal}2100);
        // 11 is the line number for other income sum of reported in wages
    endif;

    if noccurs(S4012_EDT) >= 1 then Q40121(1) = "1"; endif;
    if noccurs(S4012_EDT) >= 2 then Q40121(2) = "2"; endif;
    if noccurs(S4012_EDT) >= 3 then Q40121(3) = "3"; endif;
    if noccurs(S4012_EDT) >= 4 then Q40121(4) = "4"; endif;
    if noccurs(S4012_EDT) >= 5 then Q40121(5) = "5"; endif;
    if noccurs(S4012_EDT) >= 6 then Q40121(6) = "6"; endif;
    if noccurs(S4012_EDT) >= 7 then Q40121(7) = "7"; endif;
    if noccurs(S4012_EDT) >= 8 then Q40121(8) = "8"; endif;
PROC Q40121
recode  ID07    :   tonumber(Q40121)    =>  Q401_B_ISCO;
    151 :   1   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}522);
    151 :   2   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}262);
    223 :   1   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111);
    223 :   2   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}751);
    236 :   1   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111);
    236 :   2   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}111);
    301 :   1   =>
    setNumFlag({questName}'Q401_B_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}752);
endrecode;

recode  ID07    :   tonumber(Q40121)    =>  Q401_B_ISIC ;
    151 :   1   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    151 :   2   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}851) ;
    223 :   1   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    223 :   2   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}472) ;
    236 :   1   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    236 :   2   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    301 :   1   =>
    setNumFlag({questName}'Q401_B_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}310) ;
endrecode;

numeric tmp1 = 0;
if Q40125 > 0 then
    recode int(Q401_B_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
```

```
        10-33 => 5;
        35-35 => 6;
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_B_PACCOI, 11100+tmp1);
        impute (Q401_B_HIES_CODE, 401 + (500+int(Q401_B_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40124));
    endif;
endif;
if Q40125 in 1:99000 then
    // set paccoi
        Q401_B_PACCOI = 1104;

    Q401_B_MULTI = 1;
    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_B_HIES_CODE;
    I_PACCOI(tmp) = Q401_B_PACCOI;
    I_GROSS(tmp) = Q40125;
    I_MULTIPLIER(tmp) = Q401_B_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 1;
    I_INCOME_TYPE(tmp) = 1;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_B_ISIC;
    I_OCCUPATION(tmp) = Q401_B_ISCO;
    impute(I_BENEFICIARY(tmp), Q40122);
    I_DESCRIPTION(tmp) = concat("[secondary cash
    wage]",tolower(strip(Q40123)),":",tolower(strip(Q40124)));
endif;
PROC S4013_EDT
preproc
    if noccurs(S4013_EDT) > 0 then
        do varying i = noccurs(S4013_EDT) until i <= 0 by (-1)
            if length(strip(Q40134(i))) < 2 then
                delete(S4013_EDT(i));
                //errmsg("deleting blank records in section S4013") summary;
            endif;
        enddo;
    endif;
    if noccurs(S4013_EDT) >= 1 then Q40131(1) = "1"; endif;
    if noccurs(S4013_EDT) >= 2 then Q40131(2) = "2"; endif;
    if noccurs(S4013_EDT) >= 3 then Q40131(3) = "3"; endif;
    if noccurs(S4013_EDT) >= 4 then Q40131(4) = "4"; endif;
    if noccurs(S4013_EDT) >= 5 then Q40131(5) = "5"; endif;
    if noccurs(S4013_EDT) >= 6 then Q40131(6) = "6"; endif;
    if noccurs(S4013_EDT) >= 7 then Q40131(7) = "7"; endif;
```

```
    if noccurs(S4013_EDT) >= 8 then Q40131(8) = "8"; endif;
PROC Q40131
recode  ID07    :   tonumber(Q40131)    =>  Q401_C_ISCO ;
    127 :   1   =>
    setNumFlag({questName}'Q401_C_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}112) ;
    132 :   1   =>
    setNumFlag({questName}'Q401_C_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}961) ;
    238 :   1   =>
    setNumFlag({questName}'Q401_C_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}622) ;
    311 :   1   =>
    setNumFlag({questName}'Q401_C_ISCO',{lineNo}curocc(),{imputeType}1,{newVal}921) ;
endrecode;

recode  ID07    :   tonumber(Q40131)    =>  Q401_C_ISIC ;
    127 :   1   =>
    setNumFlag({questName}'Q401_C_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    132 :   1   =>
    setNumFlag({questName}'Q401_C_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}383) ;
    238 :   1   =>
    setNumFlag({questName}'Q401_C_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}841) ;
    311 :   1   =>
    setNumFlag({questName}'Q401_C_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}12)  ;
endrecode;

numeric tmp1 = 0;
if Q40137 > 0 then
    recode int(Q401_C_ISIC / 10) => tmp1;
        01-01 => 1;
        02-02 => 2;
        03-03 => 3;
        05-09 => 4;
        10-33 => 5;
        35-35 => 6;
        36-39 => 7;
        41-43 => 8;
        45-47 => 9;
        49-53 => 10;
        55-56 => 11;
        58-63 => 12;
        64-66 => 13;
        68-68 => 14;
        69-75 => 15;
        77-82 => 16;
        84-84 => 17;
        85-85 => 18;
        86-88 => 19;
        90-93 => 20;
        94-96 => 21;
        97-98 => 22;
        99-99 => 23;
              => 0;
    endrecode;
    if tmp1 in 1:23 then
        //impute (Q401_C_PACCOI, 11100+tmp1);
        impute (Q401_C_HIES_CODE, 401 + (600+int(Q401_C_ISIC/10))/1000);
    else
        errmsg("!!!%s",strip(Q40134));
    endif;
endif;

if Q40137 in 1:99000 then
    // set paccoi
        Q401_C_PACCOI = 1106;

    Q401_C_MULTI = 1;
    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = Q401_C_HIES_CODE;
    I_PACCOI(tmp) = Q401_C_PACCOI;
    I_GROSS(tmp) = Q40137;
```

```
    I_MULTIPLIER(tmp) = Q401_C_MULTI;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * I_MULTIPLIER(tmp);
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_LOCATION(tmp) = 2;
    I_INCOME_TYPE(tmp) = 1;
    I_PROD_TYPE(tmp) = notappl;
    I_QUANTITY(tmp) = notappl;
    I_NUMBER(tmp) = notappl;
    I_UNIT(tmp) = notappl;
    I_INDUSTRY(tmp) = Q401_C_ISIC;
    I_OCCUPATION(tmp) = Q401_C_ISCO;
    impute(I_BENEFICIARY(tmp), Q40132);
    I_DESCRIPTION(tmp) = concat("[overseas cash
    wage]",tolower(strip(Q40133)),":",tolower(strip(Q40134)));
endif;
PROC S4021_EDT
postproc
    if Q40201_1 = notappl then
        delete(S4021_EDT);
    endif;
PROC Q40202
postproc
    if invalueset($) then
        if Q40201_1 = notappl then
            Q40201_1 =
            setNumFlag({questName}'Q40201_1',{lineNo}curocc(),{imputeType}1,{newVal}1);
        endif;
    endif;


    if Q40201_1 in 1:99 then
        if Q40202 in 1:2 then
            putdeck(hd_40202_fr_tenure,Q40202);
        else
            Q40202 =
            setNumFlag({questName}'Q40202',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0202_fr_tenure));
        endif;
    endif;
PROC Q40203
postproc
    if Q40201_1 in 1:99 then
        if Q40203 in 1:3 then
            putdeck(hd_40203_fr_tenure,Q40203);
        else
            Q40203 =
            setNumFlag({questName}'Q40203',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0203_fr_tenure));
        endif;
    endif;
PROC Q40204
postproc
    if Q40201_1 in 1:99 then
        if Q40204 in 1:2 then
            putdeck(hd_40204_fr_tenure,Q40204);
        else
            Q40204 =
            setNumFlag({questName}'Q40204',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0204_fr_tenure));
        endif;
    endif;
PROC Q40205
postproc
    if Q40201_1 in 1:99 then
        if Q40205 in 0:10 then
            putdeck(hd_40205_fr_tenure,Q40205);
        else
            Q40205 =
            setNumFlag({questName}'Q40205',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0205_fr_tenure));
        endif;
```

```
    endif;
PROC Q40206
postproc
    if Q40201_1 in 1:99 then
        if Q40206 in 1:2 then
            putdeck(hd_40206_fr_tenure,Q40206);
        else
            Q40206 =
            setNumFlag({questName}'Q40206',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0206_fr_tenure));
        endif;
    endif;
PROC Q40207
postproc
    if Q40201_1 in 1:99 then
        if Q40207 in 1:2 then
            putdeck(hd_40207_fr_tenure,Q40207);
        else
            Q40207 =
            setNumFlag({questName}'Q40207',{lineNo}curocc(),{imputeType}1,{newVal}getdeck(hd_4
            0207_fr_tenure));
        endif;
    endif;
PROC Q40208_1A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_1B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_1B <> notappl then
            impute(Q40208_1B,notappl);
        endif;
    endif;
endif;
PROC Q40208_1B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904103301;
    numeric tmpPACCOI = 9201;
    numeric tmpHIEScode = 402.901;
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][transportation of goods]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
```

```
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40208_2A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_2B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_2B <> notappl then
            impute(Q40208_2B,notappl);
        endif;
    endif;
endif;
PROC Q40208_2B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904103302;
    numeric tmpPACCOI = 9202;
    numeric tmpHIEScode = 402.902;
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][labour]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
```

```
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40208_3A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_3B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_3B <> notappl then
            impute(Q40208_3B,notappl);
        endif;
    endif;
endif;
PROC Q40208_3B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904103303;
    numeric tmpPACCOI = 9203;
    numeric tmpHIEScode = 402.903;
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][fencing and enclosure]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40208_4A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_4B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
```

```
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_4B <> notappl then
            impute(Q40208_4B,notappl);
        endif;
    endif;
endif;
PROC Q40208_4B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904103304;
    numeric tmpPACCOI = 9204;
    numeric tmpHIEScode = 402.904;
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][rental of equipment]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40208_5A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_5B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_5B <> notappl then
            impute(Q40208_5B,notappl);
        endif;
    endif;
endif;
PROC Q40208_5B
if $ in 0.01:99900 then
```

```
    numeric tmpCOICOP = 904103305;
    numeric tmpPACCOI = 9205;
    numeric tmpHIEScode = 402.905;
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][seeds, fertiliser, and the like]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40208_6A
if Q40201_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40208_6B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40208_6B <> notappl then
            impute(Q40208_6B,notappl);
        endif;
    endif;
endif;
PROC Q40208_6A_O
if Q40208_6A in 2 then
    if length(strip($)) <> 0 then
        $ = "";
    endif;
endif;
PROC Q40208_6B

if $ in 0.01:99900 then
    numeric tmpCOICOP = 904103306;
    numeric tmpPACCOI = 9299;
    numeric tmpHIEScode = 402.906;
```

```
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][intermediate][other]"+Q40208_6A_O;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4022_EDT
postproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if Q40253(i) <> 1 then
            delete($(i));
            //errmsg("deleting blank records in section S4022") summary;
        endif;
    enddo;

PROC Q40251
// manual edit from Iapi's email attachment (pdf) on APR 12
if ID07 = 324 then
    if curocc() in 16 then
        impute(Q40253, 1);
        impute(Q40254, 1);
        impute(Q40255, 200);
    elseif curocc() in 17 then
        impute(Q40253, 1);
        impute(Q40254, 2);
        impute(Q40255, notappl);
    elseif curocc() in 18 then
        impute(Q40253, 1);
        impute(Q40254, 2);
        impute(Q40255, notappl);
    elseif curocc() in 19 then
        impute(Q40253, 1);
        impute(Q40254, 1);
        impute(Q40255, 150);
    elseif curocc() in 21 then
        impute(Q40253, 1);
```

```
        impute(Q40254, 1);
        impute(Q40255, 60);
    else
        impute(Q40253, 2);
        impute(Q40254, 2);
        impute(Q40255, notappl);
    endif;
endif;

PROC Q40254
postproc
    if Q40253 in 1 then
        if Q40254 = notappl then
            impute (Q40254, 2);
        endif;
    endif;
PROC Q40255
if Q40254 = 2 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;

if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // using code below
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 4;
    numeric tmpISCO = 631;
    numeric tmpISIC = 011;
    numeric tmpValue = $;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40201_1;
    string tmpDescription = "[agriculture][cash from crop sales]"+Q40252;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode curocc($) => tmpPACCOI;
        1 => 3101;
        2 => 3102;
        3 => 3103;
        4 => 3104;
        5 => 3105;
        6 => 3106;
        7 => 3107;
        8 => 3108;
        9 => 3109;
        10 => 3110;
        11 => 3111;
        12 => 3112;
        13 => 3113;
        14 => 3114;
        15 => 3115;
        16 => 3116;
        17 => 3117;
        18 => 3118;
        19 => 3119;
        20 => 3120;
        21 => 3121;
        22 => 3122;
        23 => 3123;
        24 => 3124;
        25 => 3125;
        26 => 3126;
```

```
        27 => 3127;
        28 => 3128;
        29 => 3129;
    endrecode;
    recode curocc($) => tmpCOICOP;
        1   =>  011704517;
        2   =>  011704514;
        3   =>  011704501;
        4   =>  011704524;
        5   =>  011704507;
        6   =>  011704515;
        7   =>  011704530;
        8   =>  011704522;
        9   =>  011704533;
        10  =>  011704603;
        11  =>  011704605;
        12  =>  011704604;
        13  =>  011704602;
        14  =>  011704606;
        15  =>  011604221;
        16  =>  011604209;
        17  =>  011604210;
        18  =>  011604207;
        19  =>  011604205;
        20  =>  011604219;
        21  =>  011604225;
        22  =>  011604231;
        23  =>  011604231;
        24  =>  011604231;
        25  =>  093323403;
        26  =>  011604231;
        27  =>  043111811;
        28  =>  045412903;
        29  =>  notappl;
    endrecode;

    tmpHIEScode = 402+(curocc()/1000);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4031_EDT
postproc
    if Q40301_1 = notappl then
        delete(S4031_EDT);
    endif;
PROC Q40302
postproc
    if invalueset(Q40303A) then
        if Q40301_1 = notappl then
```

```
            Q40301_1 =
            setNumFlag({questName}'Q40301_1',{lineNo}curocc(),{imputeType}1,{newVal}1);
        endif;
    endif;
    if Q40301_1 in notappl then
        if $ <> notappl then
            $ =
            setNumFlag({questName}'Q40302',{lineNo}curocc(),{imputeType}1,{newVal}notappl);
        endif;
    endif;
    if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
        if Q40302 in notappl then
            Q40302 = setNumFlag({questName}'Q40302',{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
    endif;


PROC Q40305
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if $ in notappl then
        impute($, 0);
    endif;
endif;
PROC Q40306
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if $ in notappl then
        impute($, 0);
    endif;
endif;
PROC Q40307
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if $ in 1:4 then
        putdeck(hd_40307_fr_tenure,$);
    else
        impute($, getdeck(hd_40307_fr_tenure));
    endif;
endif;

PROC Q40308_1A
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40308_1B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40308_1B <> notappl then
            impute(Q40308_1B,notappl);
        endif;
    endif;
endif;
PROC Q40308_1B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904106303;
    numeric tmpPACCOI = 9301;
    numeric tmpHIEScode = 403.901;
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fishing][intermediate][equipment]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
```

```
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;

PROC Q40308_2A
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40308_2B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40308_2B <> notappl then
            impute(Q40308_2B,notappl);
        endif;
    endif;
endif;
PROC Q40308_2B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904106304;
    numeric tmpPACCOI = 9302;
    numeric tmpHIEScode = 403.902;
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fishing][intermediate][transportation of goods]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
```

```
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40308_3A
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40308_3B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40308_3B <> notappl then
            impute(Q40308_3B,notappl);
        endif;
    endif;
endif;
PROC Q40308_3B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904106305;
    numeric tmpPACCOI = 9303;
    numeric tmpHIEScode = 403.903;
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fishing][intermediate][labor]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
```

```
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40308_4A
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40308_4B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40308_4B <> notappl then
            impute(Q40308_4B,notappl);
        endif;
    endif;
endif;
PROC Q40308_4B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904106306;
    numeric tmpPACCOI = 9304;
    numeric tmpHIEScode = 403.904;
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fishing][intermediate][ice]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40308_5A
```

```
if Q40301_1 in 1:count(S1010_EDT where sex in 1:2) then
    if Q40308_5B in 1:99000 then
        if $ <> 1 then
            impute($, 1);
        endif;
    else
        if $ <> 2 then
            impute($, 2);
        endif;
        if Q40308_5B <> notappl then
            impute(Q40308_5B,notappl);
        endif;
    endif;
endif;
PROC Q40308_5B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904106307;
    numeric tmpPACCOI = 9399;
    numeric tmpHIEScode = 403.905;
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fishing][intermediate][other]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4032_EDT
preproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if Q40353(i) <> 1 then
            delete($(i));
            //errmsg("deleting blank records in section S4032") summary;
        endif;
    enddo;

PROC Q40354
postproc
```

```
    if Q40353 in 1 then
        if Q40354 = notappl then
            impute (Q40354, 2);
        endif;
    endif;
PROC Q40355
postproc
    if Q40354 in 2 then
        if Q40355 = 0 then
            impute (Q40355, notappl);
        endif;
        if Q40355 > 0 then
            impute (Q40354, 1);
        endif;
    endif;

if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // using code below
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 4;
    numeric tmpISCO = 634;
    numeric tmpISIC = 031;
    numeric tmpValue = $;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40301_1;
    string tmpDescription = "[fisheries][cash from fish sales]"+Q40352;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode curocc($) => tmpPACCOI;
        1 => 3301;
        2 => 3302;
        3 => 3303;
        4 => 3304;
        5 => 3305;
        6 => 3306;
        7 => 3307;
        8 => 3308;
        9 => 3309;
        10 => 3310;
        11 => 3311;
        12 => 3312;
        13 => 3313;
        14 => 3314;
        15 => 3315;
        16 => 3316;
        17 => 3317;
        18 => 3318;
        19 => 3319;
        20 => 3320;
        21 => 3321;
        22 => 3322;
        23 => 3323;
        24 => 3324;
        25 => 3325;
    endrecode;
    recode curocc($) => tmpCOICOP;
        1   =>  011302231;
        2   =>  011302228;
        3   =>  011302232;
        4   =>  011302234;
        5   =>  011302234;
        6   =>  011302234;
```

```
        7   =>  011302234;
        8   =>  011302234;
        9   =>  011302234;
        10  =>  011302233;
        11  =>  011302233;
        12  =>  011302236;
        13  =>  011302305;
        14  =>  011302303;
        15  =>  011302304;
        16  =>  011302313;
        17  =>  011302306;
        18  =>  011302313;
        19  =>  011302310;
        20  =>  011302301;
        21  =>  011302101;
        22  =>  011302313;
        23  =>  011201701;
        24  =>  011201203;
        25  =>  011201701;
    endrecode;

    tmpHIEScode = 403+(curocc()/1000);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4041_EDT
postproc
    if Q40401_1 = notappl then
        delete(S4041_EDT);
    endif;
PROC Q40402
postproc
    if invalueset(Q40402) then
        if Q40401_1 = notappl and Q40401_2 = notappl then
            Q40401_2 =
            setNumFlag({questName}'Q40401_2',{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
        if Q40401_1 = notappl then
            Q40401_1 =
            setNumFlag({questName}'Q40401_1',{lineNo}curocc(),{imputeType}1,{newVal}1);
        endif;
    endif;
    {if Q40301(1) in notappl then
        if $ <> notappl then
            impute($, notappl);
        endif;
    endif;
    }
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
```

```
        if $ in notappl then
            $ = setNumFlag({questName}getsymbol(),{lineNo}curocc(),{imputeType}1,{newVal}2);
        endif;
    endif;


PROC Q40403
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40404_4A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40404_5A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40404_6A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_1B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105301;
    numeric tmpPACCOI = 9401;
    numeric tmpHIEScode = 404.901;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][feed]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
```

```
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40405_2A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_2B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105302;
    numeric tmpPACCOI = 9402;
    numeric tmpHIEScode = 404.902;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][fencing and enclosure]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40405_3A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_3B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105303;
    numeric tmpPACCOI = 9403;
    numeric tmpHIEScode = 404.903;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
```

```
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][transportation of goods and animals]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40405_4A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_4B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105304;
    numeric tmpPACCOI = 9404;
    numeric tmpHIEScode = 404.904;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][purchase of stock animals]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
```

```
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40405_5A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_5B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105305;
    numeric tmpPACCOI = 9405;
    numeric tmpHIEScode = 404.905;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][labour]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
```

```
PROC Q40405_6A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_6B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105306;
    numeric tmpPACCOI = 9406;
    numeric tmpHIEScode = 404.906;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][intermediate][veterinary services]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40405_7A
    if Q40401_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40405_7B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904105307;
    numeric tmpPACCOI = 9499;
    numeric tmpHIEScode = 404.907;
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
```

```
    string tmpDescription = "[livestock][intermediate][other]" + Q40405_7A_O;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4042_EDT
postproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if Q40453(i) <> 1 then
            delete($(i));
            //errmsg("deleting blank records in section S4042") summary;
        endif;
    enddo;

PROC Q40454
postproc
    if $ in 1:5000 then
        if Q40453 <> 1 then
            impute(Q40453, 1);
        endif;
    endif;
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // using code below
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 1;
    numeric tmpISCO = 632;
    numeric tmpISIC = 014;
    numeric tmpValue = $;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40401_1;
    string tmpDescription = "[livestock][cash from livestock and aquaculture sales]"+Q40452;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode curocc($) => tmpPACCOI;
```

```
        1 => 3503;
        2 => 3508;
        3 => 3509;
        4 => 3524;
        5 => 3525;
        6 => 3527;
    endrecode;
    recode curocc($) => tmpCOICOP;
        1   =>  011201202;
        2   =>  011201510;
        3   =>  011201701;
        4   =>  011302310;
        5   =>  011302303;
        6   =>  011302313;
    endrecode;

    tmpHIEScode = 404+(curocc()/1000);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4051_EDT
postproc
    if Q40501_1 = notappl then
        delete(S4051_EDT);
    endif;

PROC Q40502
    if invalueset($) then
        if Q40501_1 = notappl then
            impute(Q40501_1, 1);
        endif;
    endif;
    if Q40501_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40503_1A
    if Q40501_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40503_1B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904104302;
    numeric tmpPACCOI = 9501;
    numeric tmpHIEScode = 405.901;
    numeric tmpMulti = 4;
```

```
    numeric tmpISCO = 731;
    numeric tmpISIC = 131;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40501_1;
    string tmpDescription = "[handicraft][intermediate][raw materials]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40503_2A
    if Q40501_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40503_2B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904104303;
    numeric tmpPACCOI = 9502;
    numeric tmpHIEScode = 405.902;
    numeric tmpMulti = 4;
    numeric tmpISCO = 731;
    numeric tmpISIC = 131;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40501_1;
    string tmpDescription = "[handicraft][intermediate][transporation of goods]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));
```

```
    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40503_3A
    if Q40501_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40503_3B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904104304;
    numeric tmpPACCOI = 9503;
    numeric tmpHIEScode = 405.903;
    numeric tmpMulti = 4;
    numeric tmpISCO = 731;
    numeric tmpISIC = 131;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40501_1;
    string tmpDescription = "[handicraft][intermediate][labour]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
```

```
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40503_4A
    if Q40501_1 in 1:count(S1010_EDT where sex in 1:2) then
        if $ in notappl then
            impute($, 2);
        endif;
    endif;
PROC Q40503_4B
if $ in 0.01:99900 then
    numeric tmpCOICOP = 904104305;
    numeric tmpPACCOI = 9599;
    numeric tmpHIEScode = 405.904;
    numeric tmpMulti = 4;
    numeric tmpISCO = 731;
    numeric tmpISIC = 131;
    numeric tmpValue = $;
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40501_1;
    string tmpDescription = "[handicraft][intermediate][other]";
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4052_EDT
postproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if Q40553(i) <> 1 then
            delete($(i));
            //errmsg("deleting blank records in section S4052") summary;
        endif;
    enddo;
PROC Q40555
postproc
    if $ in 1:5000 then
        if Q40554 <> 1 then
            impute(Q40554, 1);
        endif;
    else
        if Q40554 <> 2 then
            impute(Q40554, 2);
```

```
        endif;
        if $ <> notappl then
            impute(Q40555, notappl);
        endif;
    endif;
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // using code below
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 4;
    numeric tmpISCO = 731;
    numeric tmpISIC = 131;
    numeric tmpValue = $;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = 1;
    numeric tmpBeneficiary = Q40501_1;
    string tmpDescription = "[handicraft][cash from sales]"+Q40552;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode curocc($) => tmpPACCOI;
        1 => 3701;
        2 => 3702;
        3 => 3703;
        4 => 3704;
        5 => 3705;
        6 => 3706;
        7 => 3707;
        8 => 3708;
        9 => 3709;
        10 => 3710;
        11 => 3711;
        12 => 3712;
        13 => 3713;
    endrecode;
    recode curocc($) => tmpCOICOP;
        1 => 051214002;
        2 => 123228401;
        3 => 123228703;
        4 => 056117008;
        5 => 123228706;
        6 => 051113904;
        7 => 051113904;
        8 => 123228706;
        9 => 123228706;
        10 => 031109101;
        11 => 123128101;
        12 => 123128101;
        13 => 123228706;
    endrecode;

    tmpHIEScode = 405+(curocc()/1000);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
```

```
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4060_EDT
preproc
    do varying i = noccurs(S4060_EDT) until i <= 0 by (-1)
        if length(strip(Q40602(i))) <= 1 then
            delete($(i));
            //errmsg("deleting blank records in section S4060") summary;
        endif;
    enddo;
PROC Q40601
recode ID07 : Q40601 => Q406_ISIC;
    132 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471);
    233 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}471);
    256 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}110);
    318 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}561);
    323 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}561);
    341 :   1   =>
    setNumFlag({questName}'Q406_ISIC',{lineNo}curocc(),{imputeType}1,{newVal}561);
endrecode;
PROC Q40611
postproc
    if ID07 in 132 then
        Q40611 = setNumFlag({questName}'Q40611',{lineNo}curocc(),{imputeType}1,{newVal}15150);
        Q40613 = setNumFlag({questName}'Q40613',{lineNo}curocc(),{imputeType}1,{newVal}15000);

        Q40616 = setNumFlag({questName}'Q40616',{lineNo}curocc(),{imputeType}1,{newVal}1);
        Q40617 = setNumFlag({questName}'Q40617',{lineNo}curocc(),{imputeType}1,{newVal}13);
        Q40618 = setNumFlag({questName}'Q40618',{lineNo}curocc(),{imputeType}1,{newVal}100);

    endif;
    if ID07 in 233 then
        Q40611 = setNumFlag({questName}'Q40611',{lineNo}curocc(),{imputeType}1,{newVal}3600);
        Q40612 = setNumFlag({questName}'Q40612',{lineNo}curocc(),{imputeType}1,{newVal}200);
        Q40613 = setNumFlag({questName}'Q40613',{lineNo}curocc(),{imputeType}1,{newVal}3400);

        Q40616 = setNumFlag({questName}'Q40616',{lineNo}curocc(),{imputeType}1,{newVal}1);
        Q40617 = setNumFlag({questName}'Q40617',{lineNo}curocc(),{imputeType}1,{newVal}13);
        Q40618 = setNumFlag({questName}'Q40618',{lineNo}curocc(),{imputeType}1,{newVal}100);
    endif;

    if ID07 in 318 then
        Q40619 = setNumFlag({questName}'Q40619',{lineNo}curocc(),{imputeType}1,{newVal}13);
        Q40620 = setNumFlag({questName}'Q40620',{lineNo}curocc(),{imputeType}1,{newVal}72);
    endif;

    if ID07 in 323 then
        Q40619 = setNumFlag({questName}'Q40619',{lineNo}curocc(),{imputeType}1,{newVal}13);
        Q40620 = setNumFlag({questName}'Q40620',{lineNo}curocc(),{imputeType}1,{newVal}75);
    endif;

if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl;
    numeric tmpPACCOI = 2100;
    numeric tmpHIEScode = 406 + (100+int(Q406_ISIC/10))/1000;
    numeric tmpMulti = 1;
```

```
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = $;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = "[non-subsistence business][gross]"+Q40602;
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40614
postproc
    if $ in 2 then
        $ = setNumFlag({questName}getsymbol(),{lineNo}curocc(),{imputeType}1,{newVal}1);
        Q40615 = setNumFlag({questName}'Q40615',{lineNo}curocc(),{imputeType}1,{newVal}100);
    endif;

PROC Q40618
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // assigned below
    numeric tmpPACCOI = notappl; // assigned below
    numeric tmpHIEScode = notappl; // assigned below
    numeric tmpMulti = 1;
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = notappl; // assigned below
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = ""; //assigned below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
```

```
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    recode Q40617 => tmpCOICOP;
        1   =>  904102301   ;
        2   =>  904102302   ;
        3   =>  904102303   ;
        4   =>  904102304   ;
        5   =>  904102305   ;
        6   =>  904102306   ;
        7   =>  904102307   ;
        8   =>  904102308   ;
        9   =>  904102309   ;
        10  =>  904102310   ;
        11  =>  904102311   ;
        12  =>  904102312   ;
        13  =>  904102313   ;
    endrecode;

    recode Q40617 => tmpPACCOI;
        1   =>  9101    ;
        2   =>  9102    ;
        3   =>  9103    ;
        4   =>  9104    ;
        5   =>  9105    ;
        6   =>  9106    ;
        7   =>  9107    ;
        8   =>  9108    ;
        9   =>  9109    ;
        10  =>  9110    ;
        11  =>  9111    ;
        12  =>  9112    ;
        13  =>  9199    ;
    endrecode;
    tmpHIEScode = 406 + (900 + Q40617)/1000;
    tmpValue = Q40612 * (Q40618/100);

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[non-subsistence business][intermediate]" + getlabel(Q40617,Q40617);



    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
```

```
PROC Q40620
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // assigned below
    numeric tmpPACCOI = notappl; // assigned below
    numeric tmpHIEScode = notappl; // assigned below
    numeric tmpMulti = 1;
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = notappl; // assigned below
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = ""; //assigned below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    recode Q40619 => tmpCOICOP;
        1   =>  904102301   ;
        2   =>  904102302   ;
        3   =>  904102303   ;
        4   =>  904102304   ;
        5   =>  904102305   ;
        6   =>  904102306   ;
        7   =>  904102307   ;
        8   =>  904102308   ;
        9   =>  904102309   ;
        10  =>  904102310   ;
        11  =>  904102311   ;
        12  =>  904102312   ;
        13  =>  904102313   ;
    endrecode;

    recode Q40619 => tmpPACCOI;
        1   =>  9101    ;
        2   =>  9102    ;
        3   =>  9103    ;
        4   =>  9104    ;
        5   =>  9105    ;
        6   =>  9106    ;
        7   =>  9107    ;
        8   =>  9108    ;
        9   =>  9109    ;
        10  =>  9110    ;
        11  =>  9111    ;
        12  =>  9112    ;
        13  =>  9199    ;
    endrecode;
    tmpHIEScode = 406 + (900 + Q40619)/1000;
    tmpValue = Q40612 * (Q40620/100);

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[non-subsistence business][intermediate]" + getlabel(Q40619,Q40619);



    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
```

```
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40622
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // assigned below
    numeric tmpPACCOI = notappl; // assigned below
    numeric tmpHIEScode = notappl; // assigned below
    numeric tmpMulti = 1;
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = notappl; // assigned below
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = ""; //assigned below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    recode Q40621 => tmpCOICOP;
        1   =>  904102301   ;
        2   =>  904102302   ;
        3   =>  904102303   ;
        4   =>  904102304   ;
        5   =>  904102305   ;
        6   =>  904102306   ;
        7   =>  904102307   ;
        8   =>  904102308   ;
        9   =>  904102309   ;
        10  =>  904102310   ;
        11  =>  904102311   ;
        12  =>  904102312   ;
        13  =>  904102313   ;
    endrecode;

    recode Q40621 => tmpPACCOI;
        1   =>  9101    ;
        2   =>  9102    ;
        3   =>  9103    ;
        4   =>  9104    ;
        5   =>  9105    ;
        6   =>  9106    ;
        7   =>  9107    ;
        8   =>  9108    ;
        9   =>  9109    ;
        10  =>  9110    ;
        11  =>  9111    ;
        12  =>  9112    ;
        13  =>  9199    ;
    endrecode;
```

```
    tmpHIEScode = 406 + (900 + Q40621)/1000;
    tmpValue = Q40612 * (Q40622/100);

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[non-subsistence business][intermediate]" + getlabel(Q40621,Q40621);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40624
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // assigned below
    numeric tmpPACCOI = notappl; // assigned below
    numeric tmpHIEScode = notappl; // assigned below
    numeric tmpMulti = 1;
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = notappl; // assigned below
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = ""; //assigned below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;

    recode Q40623 => tmpCOICOP;
        1   =>  904102301   ;
        2   =>  904102302   ;
        3   =>  904102303   ;
        4   =>  904102304   ;
        5   =>  904102305   ;
        6   =>  904102306   ;
        7   =>  904102307   ;
        8   =>  904102308   ;
        9   =>  904102309   ;
        10  =>  904102310   ;
        11  =>  904102311   ;
        12  =>  904102312   ;
        13  =>  904102313   ;
    endrecode;
```

```
    recode Q40623 => tmpPACCOI;
        1   =>  9101    ;
        2   =>  9102    ;
        3   =>  9103    ;
        4   =>  9104    ;
        5   =>  9105    ;
        6   =>  9106    ;
        7   =>  9107    ;
        8   =>  9108    ;
        9   =>  9109    ;
        10  =>  9110    ;
        11  =>  9111    ;
        12  =>  9112    ;
        13  =>  9199    ;
    endrecode;
    tmpHIEScode = 406 + (900 + Q40623)/1000;
    tmpValue = Q40612 * (Q40624/100);

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[non-subsistence business][intermediate]" + getlabel(Q40623,Q40623);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC Q40626
if $ in 0.01:99900 then
    numeric tmpCOICOP = notappl; // assigned below
    numeric tmpPACCOI = notappl; // assigned below
    numeric tmpHIEScode = notappl; // assigned below
    numeric tmpMulti = 1;
    numeric tmpISCO = 142;
    numeric tmpISIC = Q406_ISIC;
    numeric tmpValue = notappl; // assigned below
    numeric tmpIncomeType = 6; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // assigned below
    numeric tmpBeneficiary = Q40603;
    string tmpDescription = ""; //assigned below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
```

```
    recode Q40625 => tmpCOICOP;
        1   =>  904102301   ;
        2   =>  904102302   ;
        3   =>  904102303   ;
        4   =>  904102304   ;
        5   =>  904102305   ;
        6   =>  904102306   ;
        7   =>  904102307   ;
        8   =>  904102308   ;
        9   =>  904102309   ;
        10  =>  904102310   ;
        11  =>  904102311   ;
        12  =>  904102312   ;
        13  =>  904102313   ;
    endrecode;

    recode Q40625 => tmpPACCOI;
        1   =>  9101    ;
        2   =>  9102    ;
        3   =>  9103    ;
        4   =>  9104    ;
        5   =>  9105    ;
        6   =>  9106    ;
        7   =>  9107    ;
        8   =>  9108    ;
        9   =>  9109    ;
        10  =>  9110    ;
        11  =>  9111    ;
        12  =>  9112    ;
        13  =>  9199    ;
    endrecode;
    tmpHIEScode = 406 + (900 + Q40625)/1000;
    tmpValue = Q40612 * (Q40626/100);

    if Q40607 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[non-subsistence business][intermediate]" + getlabel(Q40625,Q40625);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4070_EDT
preproc
    if hh_ok then
        if totocc(S4070_EDT) < maxocc(S4070_EDT) then
            do varying i = totocc(S4070_EDT) + 1 until i > maxocc(S4070_EDT);
```

```
                insert(S4070_EDT(i));
                //errmsg("4070(%d) insert sucessful!",i);
            enddo;
        endif;

        if ID07 = 304 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 24 and Q40704(i) > 0 then
                    Q40701(i) = 25;
                    if i <> 11 then
                        swap(S4070_EDT,i, 11);
                    endif;
                endif;
            enddo;
        endif;
        if ID07 = 309 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 22 and Q40704(i) > 0 then
                    Q40701(i) = 25;
                    if i <> 11 then
                        swap(S4070_EDT,i, 11);
                    endif;
                endif;
            enddo;
        endif;
        if ID07 = 310 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 22 and Q40704(i) > 0 then
                    Q40701(i) = 22;
                    if i <> 8 then
                        swap(S4070_EDT,i, 8);
                    endif;
                endif;
            enddo;
        endif;
        if ID07 = 310 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 34 and Q40704(i) > 0 then
                    Q40701(i) = 25;
                    if i <> 11 then
                        swap(S4070_EDT,i, 11);
                    endif;
                endif;
            enddo;
        endif;
        if ID07 = 311 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 22 and Q40704(i) > 0 then
                    Q40701(i) = 22;
                    if i <> 8 then
                        swap(S4070_EDT,i, 8);
                    endif;
                endif;
            enddo;
        endif;
        if ID07 = 311 then
            do varying i = 1 until i > maxocc(S4070_EDT)
                if Q40701(i) = 25 and Q40704(i) > 0 then
                    Q40701(i) = 25;
                    if i <> 11 then
                        swap(S4070_EDT,i, 11);
                    endif;
                endif;
            enddo;
        endif;
    endif;

postproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if Q40703(i) <> 1 then
            delete($(i));
```

```
            //errmsg("deleting blank records in section S4070") summary;
        endif;
    enddo;

PROC Q40701
postproc
// manual fix of section 4.7
if curocc() = 1 and Q40701 <> 11 then   if $ = notappl then impute(Q40701, 11); Q40702 =
getlabel(Q40701,11);   impute(Q40703, 2);  endif;  endif;
if curocc() = 2 and Q40701 <> 12 then   if $ = notappl then impute(Q40701, 12); Q40702 =
getlabel(Q40701,12);   impute(Q40703, 2);  endif;  endif;
if curocc() = 3 and Q40701 <> 13 then   if $ = notappl then impute(Q40701, 13); Q40702 =
getlabel(Q40701,13);   impute(Q40703, 2);  endif;  endif;
if curocc() = 4 and Q40701 <> 14 then   if $ = notappl then impute(Q40701, 14); Q40702 =
getlabel(Q40701,14);   impute(Q40703, 2);  endif;  endif;
if curocc() = 5 and Q40701 <> 15 then   if $ = notappl then impute(Q40701, 15); Q40702 =
getlabel(Q40701,15);   impute(Q40703, 2);  endif;  endif;
if curocc() = 6 and Q40701 <> 16 then   if $ = notappl then impute(Q40701, 16); Q40702 =
getlabel(Q40701,16);   impute(Q40703, 2);  endif;  endif;
if curocc() = 7 and Q40701 <> 21 then   if $ = notappl then impute(Q40701, 21); Q40702 =
getlabel(Q40701,21);   impute(Q40703, 2);  endif;  endif;
if curocc() = 8 and Q40701 <> 22 then   if $ = notappl then impute(Q40701, 22); Q40702 =
getlabel(Q40701,22);   impute(Q40703, 2);  endif;  endif;
if curocc() = 9 and Q40701 <> 23 then   if $ = notappl then impute(Q40701, 23); Q40702 =
getlabel(Q40701,23);   impute(Q40703, 2);  endif;  endif;
if curocc() = 10 and Q40701 <> 24 then  if $ = notappl then impute(Q40701, 24); Q40702 =
getlabel(Q40701,24);   impute(Q40703, 2);  endif;  endif;
if curocc() = 11 and Q40701 <> 25 then  if $ = notappl then impute(Q40701, 25); Q40702 =
getlabel(Q40701,25);   impute(Q40703, 2);  endif;  endif;
if curocc() = 12 and Q40701 <> 31 then  if $ = notappl then impute(Q40701, 31); Q40702 =
getlabel(Q40701,31);   impute(Q40703, 2);  endif;  endif;
if curocc() = 13 and Q40701 <> 32 then  if $ = notappl then impute(Q40701, 32); Q40702 =
getlabel(Q40701,32);   impute(Q40703, 2);  endif;  endif;
if curocc() = 14 and Q40701 <> 33 then  if $ = notappl then impute(Q40701, 33); Q40702 =
getlabel(Q40701,33);   impute(Q40703, 2);  endif;  endif;
if curocc() = 15 and Q40701 <> 34 then  if $ = notappl then impute(Q40701, 34); Q40702 =
getlabel(Q40701,34);   impute(Q40703, 2);  endif;  endif;
PROC Q40703
postproc
    if $ = notappl then
        impute($, 2);
    endif;
    if Q40704 > 0 or Q40706 > 0 then
        if $ <> 1 then
            impute ($, 1);
        endif;
    endif;

    if $ = 1 then
        if Q40701 in 11:24 then
            if Q40705 = notappl then
                impute(Q40705, 1);
                if Q40706 = notappl then
                    impute(Q40706, Q40704);
                endif;
            endif;
        endif;
    endif;

PROC Q40704
if Q40703 = 2 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC Q40705
if Q40703 = 2 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
```

```
PROC Q40706
if Q40703 = 2 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC Q40707
if Q40703 = 2 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;

//lets fix some totals and amounts
if Q40701 in 11:16, 21:24 then
    if Q40704 * Q40705 <> Q40706 then
        Q40706 = setNumFlag({questName}'Q40706',{lineNo}curocc(),{imputeType}1,{newVal}
        Q40704*Q40705);
    endif;
elseif Q40701 in 25, 31:34 then
    if Q40704 <> notappl then
        Q40704 = setNumFlag({questName}'Q40704',{lineNo}curocc(),{imputeType}1,{newVal}
        notappl);
    endif;
    if Q40705 <> notappl then
        Q40705 = setNumFlag({questName}'Q40705',{lineNo}curocc(),{imputeType}1,{newVal}
        notappl);
    endif;
else
    errmsg("WARNING::CODE NOT FOUND!");
endif;

if Q40703 in 1 then
    if Q40707 in 1:3 then
        putdeck(hd_40707_fr_tenure, Q40707);
    else
        Q40707 = setNumFlag({questName}'Q40707',{lineNo}curocc(),{imputeType}1,{newVal}
        getdeck(hd_40707_fr_tenure));
    endif;
endif;
PROC Q40799
if Q40706 in 0.01:999900 then
    numeric tmpCOICOP = notappl; // not necessary for this section
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 1;
    numeric tmpISCO = notappl; // not necessary for this section
    numeric tmpISIC = notappl; // not necessary for this seciton
    numeric tmpValue = Q40706;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // using the code below
    numeric tmpBeneficiary = 51;
    string tmpDescription = ""; // using the code below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode Q40701 => tmpPACCOI;
        11 => 4100;
        12 => 4200;
        13 => 4300;
        21 => 5100;
        22 => 5200;
        23 => 5300;
        24 => 5500;
        25 => 5900;
```

```
        31 => 6200;
    endrecode;
    tmpHIEScode = 407+(Q40701/1000);
    if Q40707 in 1:2 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[property, transfer and casual income]" + getlabel(Q40701,Q40701);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S4080_EDT
postproc
    do varying i = noccurs($) until i <= 0 by (-1)
        if length(strip(Q40802(i))) = 0 then
            delete($(i));
            errmsg("deleting blank records in section S4080") summary;
        endif;
    enddo;

PROC Q40801
if ID07 in 150 then
    if Q40801 in notappl then
        impute(Q40801, 1);
    endif;
endif;

if ID07 in 327 then
    if Q40801 in 1 then
        Q40802 = "NO NAME";
    endif;
endif;
PROC Q40808
if Q40808 in 0.01:999900 then
    numeric tmpCOICOP = notappl; // not necessary for this section
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = notappl; // using code below
    numeric tmpMulti = 1;
    numeric tmpISCO = notappl; // not necessary for this section
    numeric tmpISIC = notappl; // not necessary for this seciton
    numeric tmpValue = Q40808;
    numeric tmpIncomeType = 1; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind home
    [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = notappl; // using the code below
    numeric tmpBeneficiary = 51;
    string tmpDescription = ""; // using the code below
    numeric tmpIncomeCategory = notappl;
    numeric tmpProductType = notappl;
```

```
    numeric tmpQuantity = notappl;
    numeric tmpNumber = notappl;
    numeric tmpUnit = notappl;
    numeric tmpCategory = notappl;
    numeric tmpDurability = notappl;
    //cash sales of ag produce - paccoi
    recode Q40807 => tmpPACCOI;
        1 => 7101;
        2 => 7102;
        3 => 7103;
        4 => 7104;
        5 => 7105;
        6 => 7199;
    endrecode;
    tmpHIEScode = 408+(Q40807/1000);
    if Q40804 in 1 then
        tmpLocation = 1;
    else
        tmpLocation = 2;
    endif;
    tmpDescription = "[remittance]" + getlabel(Q40807,Q40807);

    numeric tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpIncomeCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;
endif;
PROC S5011XA_EDT
if length(strip(S500_MO_A)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_A)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_A)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_A)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_A)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_A)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_A)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
```

```
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_A
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_A;
        P_AMOUNT(tmp) = Q51105_A;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_A;
        if special(Q51103_A) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_A;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_A;
        if Q51106_A = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51107_A;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_A));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_A
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_A;
        P_AMOUNT(tmp) = Q51205_A;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_A;
        if special(Q51203_A) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_A;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_A;
        if Q51206_A = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_A;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_A));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_A
    if length(strip($)) > 0 then
```

```
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_A;
        impute(P_AMOUNT(tmp),Q52002_A);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_A);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_A));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_A
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_A);
        impute(F_AMOUNT(tmp),       Q54005_A);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_A);
        impute(F_QUANTITY(tmp),     Q54003_A);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_A);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_A
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_A);
        impute(HC_AMOUNT(tmp),      Q55105_A);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_A);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_A);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
```

```
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_A
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_A);
        impute(HS_AMOUNT(tmp),      Q55205_A);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_A);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_A);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_A
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_A);
        impute(HG_AMOUNT(tmp),      Q55305_A);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_A);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_A);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XB_EDT
if length(strip(S500_MO_B)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_B)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_B)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_B)) > 0 then
                day1day = 4;
            else
```

```
                if length(strip(S500_FR_B)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_B)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_B)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_B
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_B;
        P_AMOUNT(tmp) = Q51105_B;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_B;
        if special(Q51103_B) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_B;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_B;
        if Q51106_B = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_B;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_B));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_B
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_B;
        P_AMOUNT(tmp) = Q51205_B;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_B;
        if special(Q51203_B) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_B;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_B;
```

```
        if Q51206_B = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_B;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_B));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_B
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_B;
        impute(P_AMOUNT(tmp),Q52002_B);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_B);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_B));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_B
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_B);
        impute(F_AMOUNT(tmp),       Q54005_B);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_B);
        impute(F_QUANTITY(tmp),     Q54003_B);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_B);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_B
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_B);
        impute(HC_AMOUNT(tmp),      Q55105_B);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
```

```
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_B);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_B);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_B
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_B);
        impute(HS_AMOUNT(tmp),      Q55205_B);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_B);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_B);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_B
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_B);
        impute(HG_AMOUNT(tmp),      Q55305_B);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_B);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_B);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XC_EDT
```

```
if length(strip(S500_MO_C)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_C)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_C)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_C)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_C)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_C)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_C)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_C
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_C;
        P_AMOUNT(tmp) = Q51105_C;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_C;
        if special(Q51103_C) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_C;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_C;
        if Q51106_C = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_C;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_C));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_C
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_C;
        P_AMOUNT(tmp) = Q51205_C;
        P_MULTIPLIER(tmp) = 26; //change
```

```
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_C;
        if special(Q51203_C) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_C;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_C;
        if Q51206_C = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_C;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_C));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_C
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_C;
        impute(P_AMOUNT(tmp),Q52002_C);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_C);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_C));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_C
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_C);
        impute(F_AMOUNT(tmp),       Q54005_C);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_C);
        impute(F_QUANTITY(tmp),     Q54003_C);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_C);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
```

```
    endif;
PROC Q55101_C
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_C);
        impute(HC_AMOUNT(tmp),      Q55105_C);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_C);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_C);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_C
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_C);
        impute(HS_AMOUNT(tmp),      Q55205_C);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_C);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_C);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_C
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_C);
        impute(HG_AMOUNT(tmp),      Q55305_C);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_C);
        impute(HG_NUMBER(tmp),      notappl);
```

```
        impute(HG_UNIT(tmp),        Q55304_C);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XD_EDT
if length(strip(S500_MO_D)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_D)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_D)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_D)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_D)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_D)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_D)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_D
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_D;
        P_AMOUNT(tmp) = Q51105_D;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_D;
        if special(Q51103_D) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_D;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_D;
        if Q51106_D = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_D;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_D));
        P_DAY(tmp) = getDay(getsymbol($));
```

```
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_D
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_D;
        P_AMOUNT(tmp) = Q51205_D;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_D;
        if special(Q51203_D) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_D;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_D;
        if Q51206_D = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_D;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_D));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_D
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_D;
        impute(P_AMOUNT(tmp),Q52002_D);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_D);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_D));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_D
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_D);
        impute(F_AMOUNT(tmp),       Q54005_D);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_D);
```

```
        impute(F_QUANTITY(tmp),     Q54003_D);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_D);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_D
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_D);
        impute(HC_AMOUNT(tmp),      Q55105_D);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_D);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_D);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_D
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_D);
        impute(HS_AMOUNT(tmp),      Q55205_D);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_D);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_D);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_D
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
```

```
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_D);
        impute(HG_AMOUNT(tmp),      Q55305_D);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_D);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_D);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XE_EDT
if length(strip(S500_MO_E)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_E)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_E)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_E)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_E)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_E)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_E)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_E
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_E;
        P_AMOUNT(tmp) = Q51105_E;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_E;
        if special(Q51103_E) then
            P_QUANTITY(tmp) = notappl;
        else
```

```
            P_QUANTITY(tmp) = Q51103_E;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_E;
        if Q51106_E = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_E;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_E));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_E
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_E;
        P_AMOUNT(tmp) = Q51205_E;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_E;
        if special(Q51203_E) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_E;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_E;
        if Q51206_E = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_E;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_E));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_E
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_E;
        impute(P_AMOUNT(tmp),Q52002_E);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_E);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_E));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_E
    if length(strip($)) > 0 then
```

```
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_E);
        impute(F_AMOUNT(tmp),       Q54005_E);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_E);
        impute(F_QUANTITY(tmp),     Q54003_E);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_E);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_E
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_E);
        impute(HC_AMOUNT(tmp),      Q55105_E);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_E);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_E);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_E
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_E);
        impute(HS_AMOUNT(tmp),      Q55205_E);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_E);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_E);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
```

```
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_E
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_E);
        impute(HG_AMOUNT(tmp),      Q55305_E);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_E);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_E);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XF_EDT
if length(strip(S500_MO_F)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_F)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_F)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_F)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_F)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_F)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_F)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
incDay = 0;
PROC Q51101_F
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
```

```
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_F;
        P_AMOUNT(tmp) = Q51105_F;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_F;
        if special(Q51103_F) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_F;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_F;
        if Q51106_F = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_F;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_F));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_F
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_F;
        P_AMOUNT(tmp) = Q51205_F;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_F;
        if special(Q51203_F) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_F;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_F;
        if Q51206_F = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_F;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_F));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_F
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_F;
        impute(P_AMOUNT(tmp),Q52002_F);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
```

```
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_F);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_F));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_F
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_F);
        impute(F_AMOUNT(tmp),       Q54005_F);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_F);
        impute(F_QUANTITY(tmp),     Q54003_F);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_F);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_F
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_F);
        impute(HC_AMOUNT(tmp),      Q55105_F);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_F);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_F);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_F
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
```

```
        impute(HS_COICOP(tmp),      Q552_COICOP_F);
        impute(HS_AMOUNT(tmp),      Q55205_F);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_F);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_F);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_F
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_F);
        impute(HG_AMOUNT(tmp),      Q55305_F);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_F);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_F);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XG_EDT
if length(strip(S500_MO_G)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_G)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_G)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_G)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_G)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_G)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_G)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
```

```
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_G
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_G;
        P_AMOUNT(tmp) = Q51105_G;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_G;
        if special(Q51103_G) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_G;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_G;
        if Q51106_G = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_G;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_G));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_G
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_G;
        P_AMOUNT(tmp) = Q51205_G;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_G;
        if special(Q51203_G) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_G;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_G;
        if Q51206_G = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_G;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_G));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
```

```
PROC Q52001_G
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_G;
        impute(P_AMOUNT(tmp),Q52002_G);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_G);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_G));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_G
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_G);
        impute(F_AMOUNT(tmp),       Q54005_G);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_G);
        impute(F_QUANTITY(tmp),     Q54003_G);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_G);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_G
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_G);
        impute(HC_AMOUNT(tmp),      Q55105_G);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_G);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_G);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
```

```
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_G
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_G);
        impute(HS_AMOUNT(tmp),      Q55205_G);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_G);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_G);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_G
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_G);
        impute(HG_AMOUNT(tmp),      Q55305_G);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_G);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_G);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XH_EDT
if length(strip(S500_MO_H)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_H)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_H)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_H)) > 0 then
```

```
                day1day = 4;
            else
                if length(strip(S500_FR_H)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_H)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_H)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_H
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_H;
        P_AMOUNT(tmp) = Q51105_H;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_H;
        if special(Q51103_H) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_H;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_H;
        if Q51106_H = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_H;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_H));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_H
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_H;
        P_AMOUNT(tmp) = Q51205_H;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_H;
        if special(Q51203_H) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_H;
        endif;
```

```
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_H;
        if Q51206_H = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_H;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_H));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_H
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_H;
        impute(P_AMOUNT(tmp),Q52002_H);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_H);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_H));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_H
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_H);
        impute(F_AMOUNT(tmp),       Q54005_H);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_H);
        impute(F_QUANTITY(tmp),     Q54003_H);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_H);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_H
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_H);
        impute(HC_AMOUNT(tmp),      Q55105_H);
        impute(HC_MULTIPLIER(tmp),  26); //change
```

```
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_H);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_H);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_H
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_H);
        impute(HS_AMOUNT(tmp),      Q55205_H);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_H);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_H);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_H
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_H);
        impute(HG_AMOUNT(tmp),      Q55305_H);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_H);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_H);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
```

```
    endif;
PROC S5011XI_EDT
if length(strip(S500_MO_I)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_I)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_I)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_I)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_I)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_I)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_I)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_I
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_I;
        P_AMOUNT(tmp) = Q51105_I;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_I;
        if special(Q51103_I) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_I;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_I;
        if Q51106_I = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_I;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_I));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_I
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_I;
```

```
        P_AMOUNT(tmp) = Q51205_I;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_I;
        if special(Q51203_I) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_I;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_I;
        if Q51206_I = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_I;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_I));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_I
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_I;
        impute(P_AMOUNT(tmp),Q52002_I);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_I);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_I));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_I
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_I);
        impute(F_AMOUNT(tmp),       Q54005_I);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_I);
        impute(F_QUANTITY(tmp),     Q54003_I);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_I);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
```

```
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_I
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_I);
        impute(HC_AMOUNT(tmp),      Q55105_I);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_I);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_I);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_I
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_I);
        impute(HS_AMOUNT(tmp),      Q55205_I);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_I);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_I);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_I
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_I);
        impute(HG_AMOUNT(tmp),      Q55305_I);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
```

```
        impute(HG_QUANTITY(tmp),        Q55303_I);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_I);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XJ_EDT
if length(strip(S500_MO_J)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_J)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_J)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_J)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_J)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_J)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_J)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_J
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_J;
        P_AMOUNT(tmp) = Q51105_J;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_J;
        if special(Q51103_J) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_J;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_J;
        if Q51106_J = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_J;
```

```
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_J));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_J
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_J;
        P_AMOUNT(tmp) = Q51205_J;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_J;
        if special(Q51203_J) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_J;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_J;
        if Q51206_J = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_J;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_J));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_J
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_J;
        impute(P_AMOUNT(tmp),Q52002_J);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_J);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_J));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_J
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_J);
        impute(F_AMOUNT(tmp),       Q54005_J);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
```

```
        impute(F_EXP_TYPE(tmp),      notappl);
        impute(F_PROD_TYPE(tmp),     Q54002_J);
        impute(F_QUANTITY(tmp),      Q54003_J);
        impute(F_NUMBER(tmp),        notappl);
        impute(F_UNIT(tmp),          Q54004_J);
        impute(F_LOCAL(tmp),         notappl);
        impute(F_PURPOSE(tmp),       1);
        impute(F_DAY(tmp),           getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =         tolower(strip($));
    endif;
PROC Q55101_J
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),    notappl); //change
        impute(HC_COICOP(tmp),       Q551_COICOP_J);
        impute(HC_AMOUNT(tmp),       Q55105_J);
        impute(HC_MULTIPLIER(tmp),   26); //change
        impute(HC_AN_AMT(tmp),       HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),      notappl);
        impute(HC_LOCATION(tmp),         1);
        impute(HC_EXP_TYPE(tmp),         2);
        impute(HC_PROD_TYPE(tmp),        notappl);
        impute(HC_QUANTITY(tmp),         Q55103_J);
        impute(HC_NUMBER(tmp),       notappl);
        impute(HC_UNIT(tmp),         Q55104_J);
        impute(HC_LOCAL(tmp),        1);
        impute(HC_PURPOSE(tmp),      1);
        impute(HC_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),     day1day);
        else
            impute(HC_DAY_NAME(tmp),     notappl);
        endif;
        HC_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55201_J
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),    notappl); //change
        impute(HS_COICOP(tmp),       Q552_COICOP_J);
        impute(HS_AMOUNT(tmp),       Q55205_J);
        impute(HS_MULTIPLIER(tmp),   26); //change
        impute(HS_AN_AMT(tmp),       HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),      notappl);
        impute(HS_LOCATION(tmp),         1);
        impute(HS_EXP_TYPE(tmp),         2);
        impute(HS_PROD_TYPE(tmp),        notappl);
        impute(HS_QUANTITY(tmp),         Q55203_J);
        impute(HS_NUMBER(tmp),       notappl);
        impute(HS_UNIT(tmp),         Q55204_J);
        impute(HS_LOCAL(tmp),        1);
        impute(HS_PURPOSE(tmp),      1);
        impute(HS_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),     day1day);
        else
            impute(HS_DAY_NAME(tmp),     notappl);
        endif;
        HS_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55301_J
    if length(strip($)) > 1 then
```

```
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_J);
        impute(HG_AMOUNT(tmp),      Q55305_J);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_J);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_J);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XK_EDT
if length(strip(S500_MO_K)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_K)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_K)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_K)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_K)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_K)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_K)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_K
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_K;
        P_AMOUNT(tmp) = Q51105_K;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_K;
        if special(Q51103_K) then
```

```
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_K;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_K;
        if Q51106_K = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_K;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_K));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_K
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_K;
        P_AMOUNT(tmp) = Q51205_K;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_K;
        if special(Q51203_K) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_K;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_K;
        if Q51206_K = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_K;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_K));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_K
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_K;
        impute(P_AMOUNT(tmp),Q52002_K);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_K);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_K));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
```

```
PROC Q54001_K
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_K);
        impute(F_AMOUNT(tmp),       Q54005_K);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_K);
        impute(F_QUANTITY(tmp),     Q54003_K);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_K);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_K
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_K);
        impute(HC_AMOUNT(tmp),      Q55105_K);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_K);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_K);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_K
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_K);
        impute(HS_AMOUNT(tmp),      Q55205_K);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_K);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_K);
```

```
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_K
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_K);
        impute(HG_AMOUNT(tmp),      Q55305_K);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_K);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_K);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XL_EDT
if length(strip(S500_MO_L)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_L)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_L)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_L)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_L)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_L)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_L)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_L
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
```

```
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_L;
        P_AMOUNT(tmp) = Q51105_L;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_L;
        if special(Q51103_L) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_L;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_L;
        if Q51106_L = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_L;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_L));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_L
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_L;
        P_AMOUNT(tmp) = Q51205_L;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_L;
        if special(Q51203_L) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_L;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_L;
        if Q51206_L = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_L;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_L));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_L
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_L;
        impute(P_AMOUNT(tmp),Q52002_L);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
```

```
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_L);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_L));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_L
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_L);
        impute(F_AMOUNT(tmp),       Q54005_L);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_L);
        impute(F_QUANTITY(tmp),     Q54003_L);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_L);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_L
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_L);
        impute(HC_AMOUNT(tmp),      Q55105_L);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_L);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_L);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_L
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
```

```
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_L);
        impute(HS_AMOUNT(tmp),      Q55205_L);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_L);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_L);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_L
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_L);
        impute(HG_AMOUNT(tmp),      Q55305_L);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_L);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_L);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XM_EDT
if length(strip(S500_MO_M)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_M)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_M)) > 0 then
            day1day = 3;
        else
            if length(strip(S500_TH_M)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_M)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_M)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_M)) > 0 then
                            day1day = 7;
                        else
```

```
                                day1day = 0;
                            endif;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
PROC Q51101_M
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_M;
        P_AMOUNT(tmp) = Q51105_M;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_M;
        if special(Q51103_M) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_M;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_M;
        if Q51106_M = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_M;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_M));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_M
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_M;
        P_AMOUNT(tmp) = Q51205_M;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_M;
        if special(Q51203_M) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_M;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_M;
        if Q51206_M = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_M;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_M));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
```

```
    endif;
PROC Q52001_M
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_M;
        impute(P_AMOUNT(tmp),Q52002_M);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_M);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_M));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_M
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_M);
        impute(F_AMOUNT(tmp),       Q54005_M);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_M);
        impute(F_QUANTITY(tmp),     Q54003_M);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_M);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_M
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_M);
        impute(HC_AMOUNT(tmp),      Q55105_M);
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_M);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_M);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
```

```
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_M
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_M);
        impute(HS_AMOUNT(tmp),      Q55205_M);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_M);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_M);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_M
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_M);
        impute(HG_AMOUNT(tmp),      Q55305_M);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_M);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_M);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC S5011XN_EDT
if length(strip(S500_MO_N)) > 0 then
    day1day = 1;
else
    if length(strip(S500_TU_N)) > 0 then
        day1day = 2;
    else
        if length(strip(S500_WE_N)) > 0 then
            day1day = 3;
        else
```

```
            if length(strip(S500_TH_N)) > 0 then
                day1day = 4;
            else
                if length(strip(S500_FR_N)) > 0 then
                    day1day = 5;
                else
                    if length(strip(S500_SA_N)) > 0 then
                        day1day = 6;
                    else
                        if length(strip(S500_SU_N)) > 0 then
                            day1day = 7;
                        else
                            day1day = 0;
                        endif;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
PROC Q51101_N
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_N;
        P_AMOUNT(tmp) = Q51105_N;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_N;
        if special(Q51103_N) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_N;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_N;
        if Q51106_N = "X" then

        else

        endif;
        P_PURPOSE(tmp) = Q51107_N;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_N));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q51201_N
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        numeric tmp2 = 0;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_N;
        P_AMOUNT(tmp) = Q51205_N;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_N;
        if special(Q51203_N) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_N;
```

```
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_N;
        if Q51206_N = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_N;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_N));
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q52001_N
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        impute(P_HIES_CODE(tmp),notappl); //change
        P_COICOP(tmp) = Q520_COICOP_N;
        impute(P_AMOUNT(tmp),Q52002_N);
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = P_AN_AMT(tmp) * WEIGHT;
        P_LOCATION(tmp) = notappl;
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = notappl;
        P_QUANTITY(tmp) = notappl;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = notappl;
        P_LOCAL(tmp) = notappl;
        impute(P_PURPOSE(tmp),Q52003_N);
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_N));//change
        P_DAY(tmp) = getDay(getsymbol($));
        if day1day > 0 then impute(P_DAY_NAME(tmp),day1day); endif;
    endif;
PROC Q54001_N
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        impute(F_LINE_NUMBER(tmp),  tmp);
        impute(F_HIES_CODE(tmp),    notappl); //change
        impute(F_COICOP(tmp),       Q540_COICOP_N);
        impute(F_AMOUNT(tmp),       Q54005_N);
        impute(F_MULTIPLIER(tmp),   26); //change
        impute(F_AN_AMT(tmp),       F_AMOUNT(tmp) * F_MULTIPLIER(tmp));
        impute(F_WGHT_AN_AMT(tmp),  notappl);
        impute(F_LOCATION(tmp),     notappl);
        impute(F_EXP_TYPE(tmp),     notappl);
        impute(F_PROD_TYPE(tmp),    Q54002_N);
        impute(F_QUANTITY(tmp),     Q54003_N);
        impute(F_NUMBER(tmp),       notappl);
        impute(F_UNIT(tmp),         Q54004_N);
        impute(F_LOCAL(tmp),        notappl);
        impute(F_PURPOSE(tmp),      1);
        impute(F_DAY(tmp),          getDay(getsymbol($)));
        if day1day > 0 then
            impute(F_DAY_NAME(tmp), day1day);
        else
            impute(F_DAY_NAME(tmp), notappl);
        endif;
        F_DESCRIPTION(tmp) =        tolower(strip($));
    endif;
PROC Q55101_N
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        impute(HC_LINE_NUMBER(tmp), tmp);
        impute(HC_HIES_CODE(tmp),   notappl); //change
        impute(HC_COICOP(tmp),      Q551_COICOP_N);
        impute(HC_AMOUNT(tmp),      Q55105_N);
```

```
        impute(HC_MULTIPLIER(tmp),  26); //change
        impute(HC_AN_AMT(tmp),      HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp));
        impute(HC_WGHT_AN_AMT(tmp),     notappl);
        impute(HC_LOCATION(tmp),        1);
        impute(HC_EXP_TYPE(tmp),        2);
        impute(HC_PROD_TYPE(tmp),       notappl);
        impute(HC_QUANTITY(tmp),        Q55103_N);
        impute(HC_NUMBER(tmp),      notappl);
        impute(HC_UNIT(tmp),        Q55104_N);
        impute(HC_LOCAL(tmp),       1);
        impute(HC_PURPOSE(tmp),     1);
        impute(HC_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HC_DAY_NAME(tmp),    day1day);
        else
            impute(HC_DAY_NAME(tmp),    notappl);
        endif;
        HC_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55201_N
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        impute(HS_LINE_NUMBER(tmp), tmp);
        impute(HS_HIES_CODE(tmp),   notappl); //change
        impute(HS_COICOP(tmp),      Q552_COICOP_N);
        impute(HS_AMOUNT(tmp),      Q55205_N);
        impute(HS_MULTIPLIER(tmp),  26); //change
        impute(HS_AN_AMT(tmp),      HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp));
        impute(HS_WGHT_AN_AMT(tmp),     notappl);
        impute(HS_LOCATION(tmp),        1);
        impute(HS_EXP_TYPE(tmp),        2);
        impute(HS_PROD_TYPE(tmp),       notappl);
        impute(HS_QUANTITY(tmp),        Q55203_N);
        impute(HS_NUMBER(tmp),      notappl);
        impute(HS_UNIT(tmp),        Q55204_N);
        impute(HS_LOCAL(tmp),       1);
        impute(HS_PURPOSE(tmp),     1);
        impute(HS_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HS_DAY_NAME(tmp),    day1day);
        else
            impute(HS_DAY_NAME(tmp),    notappl);
        endif;
        HS_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q55301_N
    if length(strip($)) > 1 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        impute(HG_LINE_NUMBER(tmp), tmp);
        impute(HG_HIES_CODE(tmp),   notappl); //change
        impute(HG_COICOP(tmp),      Q553_COICOP_N);
        impute(HG_AMOUNT(tmp),      Q55305_N);
        impute(HG_MULTIPLIER(tmp),  26); //change
        impute(HG_AN_AMT(tmp),      HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp));
        impute(HG_WGHT_AN_AMT(tmp),     notappl);
        impute(HG_LOCATION(tmp),        1);
        impute(HG_EXP_TYPE(tmp),        2);
        impute(HG_PROD_TYPE(tmp),       notappl);
        impute(HG_QUANTITY(tmp),        Q55303_N);
        impute(HG_NUMBER(tmp),      notappl);
        impute(HG_UNIT(tmp),        Q55304_N);
        impute(HG_LOCAL(tmp),       1);
        impute(HG_PURPOSE(tmp),     1);
        impute(HG_DAY(tmp),         getDay(getsymbol($)));
        if day1day > 0 then
            impute(HG_DAY_NAME(tmp),    day1day);
        else
            impute(HG_DAY_NAME(tmp),    notappl);
        endif;
```

```
        HG_DESCRIPTION(tmp) =       tolower(strip($));
    endif;
PROC Q51101_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_O1;
        P_AMOUNT(tmp) = Q51105_O1;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_O1;
        if special(Q51103_O1) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_O1;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_O1;
        if Q51106_O1 = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51107_O1;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_O1));
        P_DAY(tmp) = Q51100_O1;
    endif;
PROC Q51201_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q512_COICOP_O1;
        P_AMOUNT(tmp) = Q51205_O1;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_O1;
        if special(Q51203_O1) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_O1;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_O1;
        if Q51206_O1 = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_O1;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_O1));
        P_DAY(tmp) = Q51200_O1;
    endif;
PROC Q52001_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
```

```
        P_COICOP(tmp) = Q520_COICOP_O1;
        P_AMOUNT(tmp) = Q52002_O1;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PURPOSE(tmp) = Q52003_O1;
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_O1));
        P_DAY(tmp) = Q52000_O1;
    endif;
PROC Q54001_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        F_LINE_NUMBER(tmp) = tmp;
        F_HIES_CODE(tmp) = notappl; //change
        F_COICOP(tmp) = Q540_COICOP_O1;
        F_AMOUNT(tmp) = Q54005_O1;
        F_MULTIPLIER(tmp) = 26; //change
        F_AN_AMT(tmp) = F_AMOUNT(tmp) * F_MULTIPLIER(tmp);
        F_WGHT_AN_AMT(tmp) = notappl; //change
        F_EXP_TYPE(tmp) = notappl;
        F_PROD_TYPE(tmp) = Q54002_O1;
        if special(Q54003_O1) then
            F_QUANTITY(tmp) = notappl;
        else
            F_QUANTITY(tmp) = Q54003_O1;
        endif;
        F_NUMBER(tmp) = notappl;
        F_UNIT(tmp) = Q54004_O1;
        F_DESCRIPTION(tmp) = tolower(strip(Q54001_O1));
        F_DAY(tmp) = Q54000_O1;
    endif;
PROC Q55101_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
        HC_LINE_NUMBER(tmp) = tmp;
        HC_HIES_CODE(tmp) = notappl; //change
        HC_COICOP(tmp) = Q551_COICOP_O1;
        HC_AMOUNT(tmp) = Q55105_O1;
        HC_MULTIPLIER(tmp) = 26; //change
        HC_AN_AMT(tmp) = HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp);
        HC_WGHT_AN_AMT(tmp) = notappl; //change
        HC_EXP_TYPE(tmp) = notappl;
        if special(Q55103_O1) then
            HC_QUANTITY(tmp) = notappl;
        else
            HC_QUANTITY(tmp) = Q55103_O1;
        endif;
        HC_NUMBER(tmp) = notappl;
        HC_UNIT(tmp) = Q55104_O1;
        HC_DESCRIPTION(tmp) = tolower(strip(Q55101_O1));
        HC_DAY(tmp) = Q55100_O1;
    endif;
PROC Q55201_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        HS_LINE_NUMBER(tmp) = tmp;
        HS_HIES_CODE(tmp) = notappl; //change
        HS_COICOP(tmp) = Q552_COICOP_O1;
        HS_AMOUNT(tmp) = Q55205_O1;
        HS_MULTIPLIER(tmp) = 26; //change
        HS_AN_AMT(tmp) = HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp);
        HS_WGHT_AN_AMT(tmp) = notappl; //change
        HS_EXP_TYPE(tmp) = notappl;
```

```
        if special(Q55203_O1) then
            HS_QUANTITY(tmp) = notappl;
        else
            HS_QUANTITY(tmp) = Q55203_O1;
        endif;
        HS_NUMBER(tmp) = notappl;
        HS_UNIT(tmp) = Q55204_O1;
        HS_DESCRIPTION(tmp) = tolower(strip(Q55201_O1));
        HS_DAY(tmp) = Q55200_O1;
    endif;
PROC Q55301_O1
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        HG_LINE_NUMBER(tmp) = tmp;
        HG_HIES_CODE(tmp) = notappl; //change
        HG_COICOP(tmp) = Q553_COICOP_O1;
        HG_AMOUNT(tmp) = Q55305_O1;
        HG_MULTIPLIER(tmp) = 26; //change
        HG_AN_AMT(tmp) = HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp);
        HG_WGHT_AN_AMT(tmp) = notappl; //change
        HG_EXP_TYPE(tmp) = notappl;
        if special(Q55303_O1) then
            HG_QUANTITY(tmp) = notappl;
        else
            HG_QUANTITY(tmp) = Q55303_O1;
        endif;
        HG_NUMBER(tmp) = notappl;
        HG_UNIT(tmp) = Q55304_O1;
        HG_DESCRIPTION(tmp) = tolower(strip(Q55301_O1));
        HG_DAY(tmp) = Q55300_O1;
    endif;
PROC Q51101_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q511_COICOP_O2;
        P_AMOUNT(tmp) = Q51105_O2;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51102_O2;
        if special(Q51103_O2) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51103_O2;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51104_O2;
        if Q51106_O2 = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51107_O2;
        P_DESCRIPTION(tmp) = tolower(strip(Q51101_O2));
        P_DAY(tmp) = Q51100_O2;
    endif;
PROC Q51201_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
```

```
        P_COICOP(tmp) = Q512_COICOP_O2;
        P_AMOUNT(tmp) = Q51205_O2;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PROD_TYPE(tmp) = Q51202_O2;
        if special(Q51203_O2) then
            P_QUANTITY(tmp) = notappl;
        else
            P_QUANTITY(tmp) = Q51203_O2;
        endif;
        P_NUMBER(tmp) = notappl;
        P_UNIT(tmp) = Q51204_O2;
        if Q51206_O2 = "X" then
            P_LOCAL(tmp) = 1;
        else
            P_LOCAL(tmp) = 2;
        endif;
        P_PURPOSE(tmp) = Q51207_O2;
        P_DESCRIPTION(tmp) = tolower(strip(Q51201_O2));
        P_DAY(tmp) = Q51200_O2;
    endif;
PROC Q52001_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(PURCHASE_EDT)+1;
        insert(PURCHASE_EDT(tmp));
        P_LINE_NUMBER(tmp) = tmp;
        P_HIES_CODE(tmp) = notappl; //change
        P_COICOP(tmp) = Q520_COICOP_O2;
        P_AMOUNT(tmp) = Q52002_O2;
        P_MULTIPLIER(tmp) = 26; //change
        P_AN_AMT(tmp) = P_AMOUNT(tmp) * P_MULTIPLIER(tmp);
        P_WGHT_AN_AMT(tmp) = notappl; //change
        P_LOCATION(tmp) = notappl; //change
        P_EXP_TYPE(tmp) = 1;
        P_PURPOSE(tmp) = Q52003_O2;
        P_DESCRIPTION(tmp) = tolower(strip(Q52001_O2));
        P_DAY(tmp) = Q52000_O2;
    endif;
PROC Q54001_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(FREE_EDT)+1;
        insert(FREE_EDT(tmp));
        F_LINE_NUMBER(tmp) = tmp;
        F_HIES_CODE(tmp) = notappl; //change
        F_COICOP(tmp) = Q540_COICOP_O2;
        F_AMOUNT(tmp) = Q54005_O2;
        F_MULTIPLIER(tmp) = 26; //change
        F_AN_AMT(tmp) = F_AMOUNT(tmp) * F_MULTIPLIER(tmp);
        F_WGHT_AN_AMT(tmp) = notappl; //change
        F_EXP_TYPE(tmp) = notappl;
        F_PROD_TYPE(tmp) = Q54002_O2;
        if special(Q54003_O2) then
            F_QUANTITY(tmp) = notappl;
        else
            F_QUANTITY(tmp) = Q54003_O2;
        endif;
        F_NUMBER(tmp) = notappl;
        F_UNIT(tmp) = Q54004_O2;
        F_DESCRIPTION(tmp) = tolower(strip(Q54001_O2));
        F_DAY(tmp) = Q54000_O2;
    endif;
PROC Q55101_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_CONSUME_EDT)+1;
        insert(HOME_CONSUME_EDT(tmp));
```

```
        HC_LINE_NUMBER(tmp) = tmp;
        HC_HIES_CODE(tmp) = notappl; //change
        HC_COICOP(tmp) = Q551_COICOP_O2;
        HC_AMOUNT(tmp) = Q55105_O2;
        HC_MULTIPLIER(tmp) = 26; //change
        HC_AN_AMT(tmp) = HC_AMOUNT(tmp) * HC_MULTIPLIER(tmp);
        HC_WGHT_AN_AMT(tmp) = notappl; //change
        HC_EXP_TYPE(tmp) = notappl;
        if special(Q55103_O2) then
            HC_QUANTITY(tmp) = notappl;
        else
            HC_QUANTITY(tmp) = Q55103_O2;
        endif;
        HC_NUMBER(tmp) = notappl;
        HC_UNIT(tmp) = Q55104_O2;
        HC_DESCRIPTION(tmp) = tolower(strip(Q55101_O2));
        HC_DAY(tmp) = Q55100_O2;
    endif;
PROC Q55201_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_SOLD_EDT)+1;
        insert(HOME_SOLD_EDT(tmp));
        HS_LINE_NUMBER(tmp) = tmp;
        HS_HIES_CODE(tmp) = notappl; //change
        HS_COICOP(tmp) = Q552_COICOP_O2;
        HS_AMOUNT(tmp) = Q55205_O2;
        HS_MULTIPLIER(tmp) = 26; //change
        HS_AN_AMT(tmp) = HS_AMOUNT(tmp) * HS_MULTIPLIER(tmp);
        HS_WGHT_AN_AMT(tmp) = notappl; //change
        HS_EXP_TYPE(tmp) = notappl;
        if special(Q55203_O2) then
            HS_QUANTITY(tmp) = notappl;
        else
            HS_QUANTITY(tmp) = Q55203_O2;
        endif;
        HS_NUMBER(tmp) = notappl;
        HS_UNIT(tmp) = Q55204_O2;
        HS_DESCRIPTION(tmp) = tolower(strip(Q55201_O2));
        HS_DAY(tmp) = Q55200_O2;
    endif;
PROC Q55301_O2
    //  modified for overflow
    if length(strip($)) > 0 then
        numeric tmp = noccurs(HOME_GIVEN_EDT)+1;
        insert(HOME_GIVEN_EDT(tmp));
        HG_LINE_NUMBER(tmp) = tmp;
        HG_HIES_CODE(tmp) = notappl; //change
        HG_COICOP(tmp) = Q553_COICOP_O2;
        HG_AMOUNT(tmp) = Q55305_O2;
        HG_MULTIPLIER(tmp) = 26; //change
        HG_AN_AMT(tmp) = HG_AMOUNT(tmp) * HG_MULTIPLIER(tmp);
        HG_WGHT_AN_AMT(tmp) = notappl; //change
        HG_EXP_TYPE(tmp) = notappl;
        if special(Q55303_O2) then
            HG_QUANTITY(tmp) = notappl;
        else
            HG_QUANTITY(tmp) = Q55303_O2;
        endif;
        HG_NUMBER(tmp) = notappl;
        HG_UNIT(tmp) = Q55304_O2;
        HG_DESCRIPTION(tmp) = tolower(strip(Q55301_O2));
        HG_DAY(tmp) = Q55300_O2;
    endif;
PROC M_LINE_NUMBER
recode  ID07    :   M_LINE_NUMBER   =>  M_HIES_CODE ;
    140 :   6   =>  207.202 ;
    241 :   16  =>  207.202 ;
    135 :   36  =>  306.005 ;
endrecode;
```

```
recode  ID07    :   M_LINE_NUMBER   =>  M_COICOP    ;
    103 :   8   =>  92121801    ;
    105 :   21  =>  22108301    ;
    106 :   12  =>  45212702    ;
    106 :   27  =>  31209411    ;
    108 :   9   =>  92121801    ;
    109 :   6   =>  71119101    ;
    109 :   20  =>  31209306    ;
{fixed for offset}  111 :   {5}6    =>  45212702    ; // gas refill
{fixed for offset}  111 :   {8}9    =>  53215204    ; // pedestal fan
    113 :   12  =>  43111811    ;
    116 :   9   =>  105026203   ;
    116 :   12  =>  105026203   ;
    116 :   16  =>  31209412    ;
    117 :   32  =>  22108301    ;
    118 :   12  =>  105026203   ;
    119 :   6   =>  72319801    ;
    119 :   18  =>  105026203   ;
    119 :   20  =>  105026203   ;
{fixed for offset}  120 :   {5}6    =>  31209411    ; // girls lavalava;
    124 :   23  =>  63018901    ;
    124 :   24  =>  31209411    ;
    125 :   23  =>  45212702    ;
    125 :   38  =>  105026203   ;
    125 :   40  =>  105026203   ;
    125 :   42  =>  105026203   ;
{fixed for offset}  126 :   {5}6    =>  901100706   ;// other taxes;
{fixed for offset}  126 :   {18}19  =>  105026203   ;// other exp for school;
{fixed for offset}  126 :   {19}20  =>  31209407    ;
{fixed for offset}  126 :   {23}24  =>  999999998   ;
{fixed for offset}  126 :   {24}25  =>  999999998   ;
    127 :   26  =>  105026203   ;
    127 :   28  =>  62318601    ;
    127 :   30  =>  32110201    ;
    132 :   8   =>  55216411    ;
    132 :   20  =>  45212702    ;
    132 :   22  =>  53115109    ;
    132 :   31  =>  91421604    ;
    132 :   33  =>  72219601    ;
    132 :   47  =>  95425416    ;
    132 :   49  =>  95425416    ;
    132 :   54  =>  31209411    ;
    132 :   55  =>  32110301    ;
    132 :   57  =>  121127208   ;
    134 :   10  =>  51113102    ;
    135 :   13  =>  91421604    ;
    135 :   14  =>  71319301    ;
    135 :   16  =>  999999998   ;
    135 :   25  =>  95425416    ;
    135 :   26  =>  31209407    ;
    135 :   27  =>  95425416    ;
    135 :   28  =>  32110301    ;
    135 :   29  =>  32110201    ;
    135 :   30  =>  32110201    ;
    135 :   31  =>  32110301    ;
    135 :   32  =>  32110201    ;
    135 :   33  =>  32110301    ;
    135 :   34  =>  31209310    ;
    135 :   35  =>  31209412    ;
    135 :   36  =>  22108202    ;
    135 :   37  =>  21308003    ;
    135 :   38  =>  21107611    ;
    135 :   39  =>  21107611    ;
    136 :   6   =>  55216408    ;
    138 :   11  =>  95125008    ;
    138 :   12  =>  31209407    ;
    138 :   13  =>  95125008    ;
    138 :   14  =>  32110301    ;
    138 :   16  =>  32110301    ;
    138 :   17  =>  32110301    ;
```

```
138 :   22  =>  22108301    ;
139 :   6   =>  55216722    ;
140 :   15  =>  31209411    ;
141 :   8   =>  55216713    ;
141 :   9   =>  72419902    ;
141 :   10  =>  73420301    ;
141 :   20  =>  31209407    ;
141 :   21  =>  95125008    ;
141 :   22  =>  31209306    ;
141 :   23  =>  95125008    ;
141 :   24  =>  31209301    ;
141 :   25  =>  32110301    ;
141 :   26  =>  31209402    ;
141 :   27  =>  32110201    ;
141 :   28  =>  83020905    ;
141 :   29  =>  21308003    ;
141 :   30  =>  21308003    ;
143 :   8   =>  43111703    ;
143 :   10  =>  51214003    ;
143 :   13  =>  92121801    ;
143 :   14  =>  92121803    ;
143 :   27  =>  32110301    ;
143 :   29  =>  21308003    ;
143 :   30  =>  21308003    ;
143 :   31  =>  22108202    ;
147 :   16  =>  31209306    ;
147 :   19  =>  31209302    ;
147 :   20  =>  31209412    ;
147 :   21  =>  31209303    ;
147 :   22  =>  21107605    ;
147 :   23  =>  21107605    ;
147 :   24  =>  22108202    ;
148 :   7   =>  73420301    ;
148 :   8   =>  112026801   ;
148 :   18  =>  31209407    ;
148 :   19  =>  31209411    ;
148 :   20  =>  31209406    ;
148 :   21  =>  22108301    ;
148 :   22  =>  22108202    ;
148 :   23  =>  22108301    ;
150 :   4   =>  43111501    ;
150 :   5   =>  55216725    ;
150 :   6   =>  53114901    ;
150 :   7   =>  45212702    ;
150 :   19  =>  31209407    ;
150 :   20  =>  95125008    ;
150 :   21  =>  31209407    ;
150 :   22  =>  95125008    ;
150 :   23  =>  31209407    ;
150 :   24  =>  95125008    ;
150 :   25  =>  31209402    ;
150 :   26  =>  32110203    ;
150 :   27  =>  31209307    ;
150 :   28  =>  999999998   ;
150 :   29  =>  22108301    ;
150 :   30  =>  22108202    ;
150 :   31  =>  21107611    ;
150 :   32  =>  22108202    ;
150 :   33  =>  22108202    ;
150 :   34  =>  21308003    ;
150 :   35  =>  22108202    ;
151 :   4   =>  43111703    ;
151 :   5   =>  52114401    ;
151 :   6   =>  52114603    ;
151 :   7   =>  72219602    ;
151 :   18  =>  31209407    ;
151 :   19  =>  31209407    ;
151 :   20  =>  31209411    ;
151 :   21  =>  31209411    ;
151 :   22  =>  31209305    ;
151 :   23  =>  83020905    ;
```

```
    151 :   24  =>  21308003    ;
    151 :   25  =>  22108202    ;
    151 :   26  =>  22108202    ;
    151 :   27  =>  21308003    ;
    151 :   28  =>  22108202    ;
    151 :   29  =>  21308003    ;
    154 :   6   =>  43111804    ;
    154 :   7   =>  91321505    ;
    154 :   8   =>  72219602    ;
    154 :   9   =>  73420301    ;
    154 :   21  =>  31209306    ;
    154 :   22  =>  95125008    ;
    154 :   23  =>  31209402    ;
    154 :   24  =>  32110201    ;
    154 :   25  =>  31209307    ;
    154 :   26  =>  31209402    ;
    154 :   27  =>  83020905    ;
    154 :   28  =>  83020905    ;
    154 :   29  =>  83020905    ;
    154 :   30  =>  83020905    ;
    154 :   31  =>  83020905    ;
    154 :   32  =>  22108301    ;
    154 :   33  =>  22108202    ;
    155 :   7   =>  52114401    ;
    155 :   8   =>  52114202    ;
    155 :   9   =>  52114603    ;
    155 :   10  =>  56117108    ;
    155 :   11  =>  91121008    ;
    155 :   12  =>  91121011    ;
    155 :   13  =>  72219602    ;
    155 :   14  =>  73320201    ;
    155 :   15  =>  73420301    ;
    155 :   16  =>  112026801   ;
    155 :   18  =>  999999998   ;
    155 :   19  =>  72420001    ;
    155 :   20  =>  999999998   ;
    155 :   40  =>  32110301    ;
    155 :   44  =>  83020905    ;
    155 :   45  =>  83020905    ;
    155 :   46  =>  22108202    ;
    155 :   47  =>  22108202    ;
    155 :   48  =>  21308003    ;
    155 :   49  =>  21308003    ;
    155 :   50  =>  22108202    ;
    155 :   51  =>  21308003    ;
    155 :   52  =>  22108202    ;
    158 :   9   =>  31109202    ;
    158 :   10  =>  32110201    ;
    158 :   11  =>  83020905    ;
    158 :   12  =>  21107605    ;
    158 :   13  =>  22108202    ;
{fixed for offset}  159 :   {5}6    =>  55216722    ;
{fixed for offset}  159 :   {6}7    =>  53114908    ;
{fixed for offset}  159 :   {7}8    =>  53114910    ;
{fixed for offset}  159 :   {8}9    =>  71319301    ;
{fixed for offset}  159 :   {9}10   =>  72219602    ;
{fixed for offset}  159 :   {18}19  =>  31209402    ;
{fixed for offset}  159 :   {19}20  =>  31309605    ;
{fixed for offset}  159 :   {20}21  =>  121327716   ;
{fixed for offset}  159 :   {21}22  =>  31209305    ;
{fixed for offset}  159 :   {22}23  =>  32110201    ;
{fixed for offset}  159 :   {23}24  =>  22108202    ;
{fixed for offset}  159 :   {24}25  =>  22108301    ;
    160 :   7   =>  43111802    ;
    160 :   8   =>  55115911    ;
    160 :   9   =>  55115914    ;
    160 :   10  =>  53114910    ;
    160 :   11  =>  72119503    ;
    160 :   12  =>  72419902    ;
    160 :   13  =>  72219601    ;
    160 :   14  =>  73320201    ;
```

```
    160 :   15  =>  73420301    ;
    160 :   16  =>  112026801   ;
    160 :   17  =>  73220102    ;
    160 :   26  =>  31209303    ;
    160 :   27  =>  83020905    ;
    160 :   28  =>  21308003    ;
    160 :   29  =>  21107611    ;
    160 :   30  =>  21308003    ;
    201 :   8   =>  91421604    ;
    201 :   9   =>  92121803    ;
    203 :   7   =>  43111603    ;
{fixed for offset}  204 :   {6}7    =>  91421604    ;
{fixed for offset}  206 :   {19}20  =>  123128101   ;
{fixed for offset}  206 :   {20}21  =>  32110301    ;
    207 :   9   =>  999999998   ;
    207 :   13  =>  32110301    ;
{fixed for offset}  208 :   {13}14  =>  32110301    ;
{fixed for offset}  208 :   {14}15  =>  32110301    ;
{fixed for offset}  208 :   {15}16  =>  32110301    ;
{fixed for offset}  209 :   {13}14  =>  72219602    ;
{fixed for offset}  209 :   {18}19  =>  999999998   ;
{fixed for offset}  209 :   {35}36  =>  32110301    ;
{fixed for offset}  209 :   {36}37  =>  32110301    ;
{fixed for offset}  209 :   {37}38  =>  32110301    ;
{fixed for offset}  210 :   {19}20  =>  95425416    ;
{fixed for offset}  210 :   {20}21  =>  95425601    ;
{fixed for offset}  210 :   {21}22  =>  105026203   ;
    211 :   12  =>  32110301    ;
{fixed for offset}  217 :   {16}17  =>  32110301    ;
{fixed for offset}  217 :   {21}22  =>  31209401    ;
{fixed for offset}  217 :   {22}23  =>  32110301    ;
{fixed for offset}  217 :   {24}25  =>  123128101   ;
{fixed for offset}  217 :   {25}26  =>  32110301    ;
    218 :   6   =>  53115111    ;
    218 :   8   =>  53215204    ;
    219 :   14  =>  55216411    ;
    219 :   21  =>  53115111    ;
    219 :   26  =>  53215204    ;
    219 :   39  =>  32110301    ;
    219 :   41  =>  32110301    ;
    219 :   42  =>  32110301    ;
    219 :   45  =>  32110301    ;
    219 :   46  =>  32110301    ;
    219 :   47  =>  32110301    ;
{fixed for offset}  222 :   {5}6    =>  112026801   ;
    223 :   17  =>  91121008    ;
    223 :   22  =>  92121803    ;
    223 :   25  =>  112026801   ;
    223 :   39  =>  61117801    ;
    223 :   40  =>  31209310    ;
    223 :   41  =>  31209412    ;
    223 :   42  =>  31209412    ;
    223 :   50  =>  32110301    ;
    223 :   51  =>  32110301    ;
    223 :   55  =>  123128101   ;
{fixed for offset}  224 :   {18}19  =>  112026801   ;
{fixed for offset}  224 :   {20}21  =>  72420001    ;
{fixed for offset}  224 :   {34}35  =>  95425416    ;
{fixed for offset}  224 :   {37}38  =>  95425416    ;
{fixed for offset}  224 :   {38}39  =>  62218501    ;
{fixed for offset}  224 :   {39}40  =>  61117801    ;
{fixed for offset}  224 :   {42}43  =>  32110301    ;
{fixed for offset}  224 :   {47}48  =>  32110301    ;
{fixed for offset}  224 :   {50}51  =>  121127208   ;
    226 :   10  =>  72219603    ;
    226 :   12  =>  112026801   ;
    226 :   22  =>  95425416    ;
    226 :   24  =>  32110301    ;
    227 :   22  =>  95425601    ;
    227 :   24  =>  32110301    ;
    227 :   26  =>  32110301    ;
```

```
232 :   9   =>  91421604    ;
233 :   5   =>  43111802    ;
233 :   6   =>  73620501    ;
233 :   10  =>  55216411    ;
233 :   20  =>  91421604    ;
233 :   23  =>  112026801   ;
233 :   26  =>  112026801   ;
234 :   22  =>  95425416    ;
236 :   11  =>  92121803    ;
236 :   12  =>  72219601    ;
237 :   19  =>  32110301    ;
238 :   26  =>  95425416    ;
238 :   33  =>  32110301    ;
238 :   44  =>  121127208   ;
239 :   36  =>  95425416    ;
239 :   37  =>  95425416    ;
239 :   39  =>  61117801    ;
239 :   48  =>  123128101   ;
239 :   50  =>  32110301    ;
239 :   51  =>  32110301    ;
239 :   52  =>  32110301    ;
239 :   53  =>  32110301    ;
240 :   10  =>  55216607    ;
240 :   13  =>  72219601    ;
240 :   30  =>  95425416    ;
240 :   31  =>  95425601    ;
240 :   43  =>  32110301    ;
240 :   46  =>  32110301    ;
241 :   6   =>  43111703    ;
241 :   9   =>  53115111    ;
249 :   10  =>  51214003    ;
249 :   29  =>  95425416    ;
249 :   30  =>  95425416    ;
249 :   31  =>  95425416    ;
249 :   35  =>  32110301    ;
249 :   38  =>  32110301    ;
249 :   39  =>  32110301    ;
250 :   12  =>  72219603    ;
250 :   25  =>  95425601    ;
250 :   32  =>  32110301    ;
256 :   14  =>  112026801   ;
256 :   16  =>  999999998   ;
256 :   26  =>  32110301    ;
256 :   28  =>  121127208   ;
258 :   30  =>  95425416    ;
258 :   31  =>  95425416    ;
301 :   6   =>  112026802   ;
301 :   12  =>  61117801    ;
301 :   13  =>  31209301    ;
301 :   14  =>  31209304    ;
301 :   15  =>  31209411    ;
301 :   16  =>  31209304    ;
301 :   17  =>  21308004    ;
302 :   8   =>  21308004    ;
304 :   7   =>  45212702    ;
304 :   8   =>  53215204    ;
305 :   8   =>  43111821    ;
305 :   9   =>  73620501    ;
305 :   10  =>  72219601    ;
306 :   7   =>  51113102    ;
306 :   9   =>  72219601    ;
306 :   11  =>  73420301    ;
306 :   12  =>  112026801   ;
306 :   13  =>  999999998   ;
307 :   4   =>  83020905    ;
307 :   8   =>  31209304    ;
307 :   9   =>  999999998   ;
307 :   10  =>  31309605    ;
307 :   11  =>  31209610    ;
309 :   12  =>  43111703    ;
309 :   16  =>  51113303    ;
```

```
    310 :   5   =>  43111811    ;
    310 :   8   =>  54015719    ;
    310 :   17  =>  31209307    ;
    310 :   18  =>  31209406    ;
    310 :   19  =>  121327716   ;
    310 :   20  =>  999999998   ;
    310 :   21  =>  999999998   ;
    310 :   22  =>  999999998   ;
    310 :   23  =>  31209310    ;
    310 :   24  =>  121327716   ;
    310 :   25  =>  83020905    ;
    310 :   26  =>  91321505    ;
    310 :   27  =>  82020701    ;
    310 :   28  =>  22108202    ;
    311 :   9   =>  112026801   ;
    311 :   15  =>  21308004    ;
    311 :   16  =>  21308004    ;
    311 :   17  =>  21308004    ;
{fixed for offset}  319 :   {4}5    =>  43111802    ;
{fixed for offset}  319 :   {5}6    =>  43111802    ;
{fixed for offset}  319 :   {6}7    =>  43111815    ;
{fixed for offset}  319 :   {8}9    =>  43111809    ;
{fixed for offset}  319 :   {9}10   =>  43111811    ;
{fixed for offset}  319 :   {10}11  =>  43211901    ;
    320 :   8   =>  53215204    ;
    320 :   10  =>  71219202    ;
    320 :   17  =>  32110301    ;
    322 :   8   =>  51113303    ;
    322 :   9   =>  51113201    ;
    322 :   14  =>  112026801   ;
    322 :   24  =>  95425416    ;
    323 :   6   =>  43111807    ;
    323 :   7   =>  43111811    ;
    323 :   8   =>  55216722    ;
    323 :   9   =>  43111703    ;
    323 :   11  =>  72219602    ;
    324 :   6   =>  45212702    ;
    324 :   21  =>  83020905    ;
    326 :   10  =>  53215204    ;
    326 :   15  =>  92121803    ;
    326 :   27  =>  95425416    ;
    326 :   28  =>  95425416    ;
    326 :   30  =>  31209310    ;
    327 :   8   =>  43111809    ;
    327 :   12  =>  53215204    ;
    327 :   15  =>  111126301   ;
    327 :   16  =>  999999998   ;
    327 :   18  =>  999999998   ;
    327 :   30  =>  31209310    ;
    328 :   6   =>  43111802    ;
    329 :   10  =>  112026801   ;
    332 :   6   =>  83020905    ;
    332 :   7   =>  51113802    ;
    332 :   13  =>  111126302   ;
    332 :   14  =>  999999998   ;
    332 :   20  =>  999999998   ;
    338 :   7   =>  43111821    ;
    338 :   21  =>  32110301    ;
    338 :   23  =>  32110301    ;
    339 :   8   =>  51113303    ;
    339 :   21  =>  121327813   ;
    340 :   5   =>  999999998   ;
    342 :   13  =>  112026801   ;
    342 :   15  =>  999999998   ;
    342 :   16  =>  72420001    ;
    342 :   26  =>  31209310    ;
    342 :   27  =>  31209310    ;
    343 :   13  =>  31209310    ;
    343 :   14  =>  31209412    ;
    343 :   15  =>  31209412    ;
    343 :   16  =>  123128202   ;
```

```
    343 :   17  =>  31209310    ;
    347 :   15  =>  53215204    ;
    347 :   24  =>  32110301    ;
    347 :   25  =>  31209411    ;
    347 :   26  =>  31209412    ;
    349 :   12  =>  999999998   ;
    350 :   6   =>  999999998   ;
    352 :   10  =>  53215204    ;
    352 :   13  =>  112026801   ;
    352 :   15  =>  72420001    ;
    352 :   18  =>  999999998   ;
    352 :   20  =>  112026801   ;
    352 :   32  =>  31209407    ;
    352 :   38  =>  31309605    ;
    352 :   39  =>  31309605    ;
    353 :   7   =>  43111802    ;
    354 :   9   =>  43111811    ;
    354 :   19  =>  72219601    ;
    354 :   20  =>  72219603    ;
    354 :   24  =>  999999998   ;
    354 :   25  =>  72420001    ;
    354 :   27  =>  112026801   ;
    356 :   5   =>  72219602    ;
endrecode;


recode  id07    :   M_LINE_NUMBER   =>  M_COICOP    ;
    204 :   6   =>  091421604   ;
//  209 :   13  =>  053114908   ;
//  224 :   25  =>  127029901   ;
    304 :   8   =>  045212701   ;
    305 :   10  =>  073320201   ;
    305 :   9   =>  072219601   ;
    305 :   7   =>  043111821   ;
    318 :   8   =>  052114501   ;
    342 :   28  =>  121327716   ;
endrecode;




if M_COICOP = 072319902 then
    M_COICOP = 072419902;
endif;
PROC M_DESCRIPTION
numeric tmpDura;
recode  M_COICOP    =>  tmpDura ;
    11100101-23109001   =>  1   ;   //  Food and drinks
    31109101    =>  4   ;   //  Pareu (printed or not)
    31109102    =>  4   ;   //  Clothing fabric, poplin material
    31109201    =>  4   ;   //  Pareu
    31109202    =>  4   ;   //  Material, Fabrics
    31109203    =>  4   ;   //  Poplin
    31209301    =>  4   ;   //  Men And Boys - Lavalava
    31209302    =>  4   ;   //  Men And Boys  - Shirt
    31209303    =>  4   ;   //  Men And Boys - Trousers
    31209304    =>  4   ;   //  Men And Boys  - T-Shirt
    31209305    =>  4   ;   //  Men And Boys - Underwear
    31209306    =>  4   ;   //  Men And Boys - School Uniform
    31209307    =>  4   ;   //  Men And Boys - Short
    31209308    =>  4   ;   //  Men And Boys - Socks
    31209309    =>  4   ;   //  Men And Boys - Other Clothes For Man.Boy Or Ned
    31209310    =>  4   ;   //  Men And Boys  - Clothes (general - not enough detail)
    31209401    =>  4   ;   //  Women And Girls - Shirt
    31209402    =>  4   ;   //  Women And Girls - Dress, Skirt
    31209403    =>  4   ;   //  Women And Girls - Trousers, Pants
    31209404    =>  4   ;   //  Women And Girls - T-Shirt, Top
    31209405    =>  4   ;   //  Women And Girls  - Suit
    31209406    =>  4   ;   //  Women And Girls - Underwear
    31209407    =>  4   ;   //  Women And Girls - School Uniform
    31209408    =>  4   ;   //  Women And Girls - Short
```

```
    31209409    =>  4   ;   //  Women And Girls - Socks
    31209410    =>  4   ;   //  Women And Girls - Other Clother For Women/Girl Or Ned
    31209411    =>  4   ;   //  Women And Girls - lava-lava
    31209412    =>  4   ;   //  Women And Girls - Clothes (general - not enough detail)
    31209501    =>  4   ;   //  Garments For Infant 0 To 2 Years Old
    31209601    =>  4   ;   //  Unspecifed / unisex clothes in general - shirts / tops
    31209602    =>  4   ;   //  Unspecifed / unisex clothes in general - pants, jeans,
    trousers
    31209603    =>  4   ;   //  Unspecifed / unisex clothes in general - dresses, slacks,
    and similar
    31209604    =>  4   ;   //  Unspecifed / unisex clothes in general - underwear, socks
    31209605    =>  4   ;   //  Unspecifed / unisex clothes in general - pajamas, robes
    31209606    =>  4   ;   //  Unspecifed / unisex clothes in general - school uniforms
    31209607    =>  4   ;   //  Unspecifed / unisex clothes in general - business, work attire
    31209608    =>  4   ;   //  Unspecifed / unisex clothes in general - Not enough detail
    31209609    =>  4   ;   //  Unspecifed / unisex clothes in general - Not specified
    elsewhere
    31209610    =>  4   ;   //  Unspecifed / unisex clothes in general - lava lava
    31309601    =>  4   ;   //  Cotton Gloves
    31309602    =>  4   ;   //  Tie
    31309603    =>  4   ;   //  Belt
    31309604    =>  4   ;   //  Cap
    31309605    =>  4   ;   //  Hat
    31309606    =>  4   ;   //  Other Clothing Accessories
    31309701    =>  4   ;   //  Cotton Thread
    31309702    =>  4   ;   //  Rayon Material
    31309703    =>  4   ;   //  Sewing Needles, Hand
    31309704    =>  4   ;   //  Other Sewing Materials
    31409801    =>  4   ;   //  Dyeing Of Garments
    31409901    =>  4   ;   //  Cleaning Of Garments, Laundry Shop
    31410001    =>  4   ;   //  Taylor
    31410101    =>  4   ;   //  Hire Of Garments
    32110201    =>  4   ;   //  Male - Beach Jandals
    32110202    =>  4   ;   //  Male - Sport Shoes
    32110203    =>  4   ;   //  Male - Casual Shoes
    32110204    =>  4   ;   //  Male - Other Shoes For Men And Boys Or Ned
    32110301    =>  4   ;   //  Female - Beach Jandals
    32110302    =>  4   ;   //  Female - Sport Shoes
    32110303    =>  4   ;   //  Female - Casual Shoes
    32110304    =>  4   ;   //  Female - Other Shoes For Woman And Girl Or Ned
    32110401    =>  4   ;   //  Shoes For Infant - 0 To 2 Years
    32110501    =>  4   ;   //  !!DO NOT USE !!
    32110502    =>  4   ;   //  Part Of Shoes (Heels, Soles, Lace…)
    32110503    =>  4   ;   //  Other shoes related articles
    32210601    =>  2   ;   //  Repair Of Footwear
    32210701    =>  2   ;   //  Shoe-Cleaning Service
    32210801    =>  2   ;   //  Hire of footwear
    41110901    =>  2   ;   //  Rent Paid By Tenants For Main House
    41211001    =>  2   ;   //  Rent Paid By Tenants For Land / Land Lease
    41211101    =>  2   ;   //  Rent Paid By Tenants For Another House
    42111201    =>  2   ;   //  Imputed Rentals Of Owner Occupying Their Main House
    42211301    =>  2   ;   //  Imputed Rental For Secondary Residence
    42211401    =>  2   ;   //  Imputed Rentals Of Households Paying A Reduced Rental Or
    Housed Free
    43111501    =>  4   ;   //  Pipe
    43111502    =>  4   ;   //  Buteline
    43111503    =>  4   ;   //  Elbow
    43111504    =>  4   ;   //  Tap
    43111505    =>  4   ;   //  Other Plumbing Materials
    43111601    =>  4   ;   //  Ceramic Floor Tile
    43111602    =>  4   ;   //  Floor Mosaic (Shower)
    43111603    =>  4   ;   //  Other Surfacing Materials (Primaflex…)
    43111701    =>  4   ;   //  Masking Tape For Painting
    43111702    =>  4   ;   //  Paint Brush
    43111703    =>  4   ;   //  Paint
    43111704    =>  4   ;   //  Other Painting Materials
    43111801    =>  4   ;   //  Blocks,Concrete Blocks
    43111802    =>  4   ;   //  Cement
    43111803    =>  4   ;   //  Door
    43111804    =>  4   ;   //  Window, Louvre
    43111805    =>  4   ;   //  Electric Cable inc network cable
```

```
43111806    =>  4   ;   //  Electric Materials (Plugs…)
43111807    =>  4   ;   //  Sink
43111808    =>  4   ;   //  Toilet Bowl, Sit, Shower
43111809    =>  4   ;   //  Plywood
43111810    =>  4   ;   //  Iron Roofing
43111811    =>  4   ;   //  Timber inc posts, planks, scotia, weatherboard, skirting
43111812    =>  4   ;   //  Hardboard
43111813    =>  4   ;   //  Inuslation foil, DPC
43111814    =>  4   ;   //  Tank iron
43111815    =>  4   ;   //  Rebar, reinforcing steel
43111816    =>  4   ;   //  Angle bar
43111817    =>  4   ;   //  Gib board
43111818    =>  4   ;   //  Cyclone straps
43111819    =>  4   ;   //  Ridge capping
43111820    =>  4   ;   //  Spouting, gutters
43111821    =>  4   ;   //  Other material for house maintenance ned
43211901    =>  2   ;   //  Contractor, Builder, Plumber, Electrician…
44112001    =>  2   ;   //  Water Bill, Water Meter
44112002    =>  2   ;   //  Water connection fees
44112003    =>  2   ;   //  Water truck
44112004    =>  2   ;   //  Other fee related to water supply
44212101    =>  2   ;   //  Garbage disposal
44312201    =>  2   ;   //  Sewer collection and disposal
44412301    =>  2   ;   //  Compound charge, co proprietor charge for caretaking of the
compound / building (cleaning….)
44412401    =>  2   ;   //  Security Services
45112501    =>  2   ;   //  Electricity Charge, Power Meter, Bill
45112502    =>  2   ;   //  Other Electricity Connection Fees
45212601    =>  2   ;   //  Do not use this commodity thank you - town gas and natural gas
45212701    =>  2   ;   //  Gas Cylinder
45212702    =>  1   ;   //  Gas Refill
45312801    =>  1   ;   //  Kerosene for domestic heating and lighting (fuel for vehicle
see 07.2.2_196.03)
45312802    =>  1   ;   //  Benzene
45312803    =>  1   ;   //  Petrol
45312804    =>  1   ;   //  Other Fuel For Generator Or Lights (Oil…)
45412901    =>  1   ;   //  Coconut Husk
45412902    =>  1   ;   //  Charcoal Imported
45412903    =>  1   ;   //  Other Solid Fuel (Firewood…)
45513001    =>  2   ;   //  Ice used for cooling and refrigeration purposes, ice cube,
ice blocks
51113101    =>  3   ;   //  Couche, Sofa
51113102    =>  3   ;   //  Other Lounge Furniture Or Ned
51113102    =>  3   ;   //  Bookshelf
51113201    =>  3   ;   //  Dining Table
51113202    =>  3   ;   //  Other Table
51113203    =>  3   ;   //  Chairs
51113204    =>  3   ;   //  Dining Hall Cabinet
51113205    =>  3   ;   //  Other Dining Hall Furniture Or Ned
51113301    =>  3   ;   //  Beds With Mattress
51113302    =>  3   ;   //  Beds Without Mattress
51113303    =>  3   ;   //  Mattress
51113304    =>  3   ;   //  Chest Of Drawers
51113305    =>  3   ;   //  Cupboards
51113306    =>  3   ;   //  Dressing Table
51113307    =>  3   ;   //  Wardrobe
51113308    =>  3   ;   //  Other Bedroom Furniture Or Ned
51113401    =>  3   ;   //  Foodsafe
51113402    =>  3   ;   //  Pantry Cupboard
51113403    =>  3   ;   //  Other Kitchen Furniture
51113501    =>  3   ;   //  Outdoor Chairs
51113502    =>  3   ;   //  Outdoor Table
51113503    =>  3   ;   //  Other Outdoor Furniture Or Ned
51113601    =>  3   ;   //  Cradles
51113602    =>  3   ;   //  High Chairs
51113603    =>  3   ;   //  Playpens
51113604    =>  3   ;   //  Other Baby Furniture Or Ned
51113701    =>  3   ;   //  Bathroom Cabinets
51113702    =>  3   ;   //  Other Bathroom Furniture Or Ned
51113801    =>  3   ;   //  Halogen Lamps
51113802    =>  3   ;   //  Other Lamp Or Ned (Exclude Torch)
```

```
51113901    =>  3   ;   //  Computer Table
51113902    =>  3   ;   //  Mirror
51113903    =>  3   ;   //  Bookcase
51113904    =>  3   ;   //  Decoration, Carving, Painting…
51113905    =>  3   ;   //  Other Furniture Or Ned
51214001    =>  3   ;   //  Carpet
51214002    =>  3   ;   //  Mat
51214003    =>  3   ;   //  Other Floor Cover
51314101    =>  2   ;   //  Repair Of Furniture
52114201    =>  4   ;   //  Awnings
52114202    =>  4   ;   //  Curtain, Curtain Material
52114203    =>  4   ;   //  Fabric Blinds
52114301    =>  4   ;   //  Bolsters
52114302    =>  4   ;   //  Futons
52114303    =>  4   ;   //  Hammocks
52114304    =>  4   ;   //  Pillows Including Cushions
52114401    =>  4   ;   //  Bed Sheets
52114402    =>  4   ;   //  Bed Spreads
52114403    =>  4   ;   //  Bedding Sets - Combination Eg Fitted, Flat And Pillow Cases
52114404    =>  4   ;   //  Blankets
52114405    =>  4   ;   //  Eiderdowns
52114406    =>  4   ;   //  Mosquito Nets
52114407    =>  4   ;   //  Pillow Cases
52114408    =>  4   ;   //  Other Bedding Or Ned
52114501    =>  4   ;   //  Table Clothes, Napkin
52114601    =>  4   ;   //  Face Towel
52114602    =>  4   ;   //  Hand Towel
52114603    =>  4   ;   //  Bath Towel
52114604    =>  4   ;   //  Beach Towel
52114605    =>  4   ;   //  Other Towel Or Ned
52114701    =>  4   ;   //  Shopping Plastic Bags
52114702    =>  4   ;   //  Shower Curtain
52114703    =>  4   ;   //  Covers For Clothes And Furniture
52114704    =>  4   ;   //  Doilies
52114705    =>  4   ;   //  Flags
52114706    =>  4   ;   //  Laundry Bags
52114707    =>  4   ;   //  Shoe Bags
52114708    =>  4   ;   //  Tea Towels
52114709    =>  4   ;   //  Other Household Textile
52114801    =>  2   ;   //  Repair Of Such Textiles
53114901    =>  3   ;   //  Stove, Kerosene, 2-Burner
53114902    =>  3   ;   //  Wicks, Kerosene Stove
53114903    =>  3   ;   //  Dishwasher
53114904    =>  3   ;   //  Freezer
53114905    =>  3   ;   //  Fridge-Freezer
53114906    =>  3   ;   //  Microwave
53114907    =>  3   ;   //  Oven Only
53114908    =>  3   ;   //  Refrigerator
53114909    =>  3   ;   //  Stove  – Electric
53114910    =>  3   ;   //  Stove – Gas
53114911    =>  3   ;   //  Stove - Kerosen
53114912    =>  3   ;   //  Wick For Kerosen Stove
53114913    =>  3   ;   //  Stove Ned
53114914    =>  3   ;   //  Other Major Kitchen Appliance Or Ned
53115001    =>  3   ;   //  Clothes Drier
53115002    =>  3   ;   //  Washing Machine
53115003    =>  3   ;   //  Other major laundry appliances
53115101    =>  3   ;   //  Air-Conditioner
53115102    =>  3   ;   //  Electric Water Heater (For Bathing)
53115103    =>  3   ;   //  Solar Power Water Heater
53115104    =>  3   ;   //  Extractor Hoods
53115105    =>  3   ;   //  Safes
53115106    =>  3   ;   //  Sewing Machine
53115107    =>  3   ;   //  Vacuum Cleaners, Steam Cleaning Machines, Carpet Shampooing
Machines And Machines For Scrubbing, Waxing And Polishing Floors
53115108    =>  3   ;   //  Other Major Household Appliances
53115109    =>  3   ;   //  Generator
53115110    =>  3   ;   //  Solar panel
53115111    =>  3   ;   //  Water tank
53215201    =>  4   ;   //  Jug, Electric, Steel
53215202    =>  4   ;   //  Fan For Ceiling
```

```
53215203    =>  4   ;   //  Fan For Table, Gie
53215204    =>  4   ;   //  Fan, Pedestal
53215205    =>  4   ;   //  Electric Toaster
53215206    =>  4   ;   //  Rice, Frc Rice Cooker
53215207    =>  4   ;   //  Blenders
53215208    =>  4   ;   //  Can-Openers
53215209    =>  4   ;   //  Coffee Maker
53215210    =>  4   ;   //  Deep Fryers
53215211    =>  4   ;   //  Food Processor
53215212    =>  4   ;   //  Frypans
53215213    =>  4   ;   //  Irons
53215214    =>  4   ;   //  Toaster
53215215    =>  4   ;   //  Other Small Electric Appliance
53315301    =>  2   ;   //  Repair Of Household Appliance
54015401    =>  4   ;   //  Ceremic Ware
54015402    =>  4   ;   //  China (Bone China, Royal Dalton, Lladro)
54015403    =>  4   ;   //  Crystal
54015404    =>  4   ;   //  Enamel
54015405    =>  4   ;   //  Glassware
54015406    =>  4   ;   //  Plastic Ware
54015407    =>  4   ;   //  Steel
54015408    =>  4   ;   //  Other Glassware Or Glassware Ned
54015501    =>  4   ;   //  Cutlery, Flatware And Silverware
54015601    =>  4   ;   //  Frying Pans
54015602    =>  4   ;   //  Hotplates
54015603    =>  4   ;   //  Household Scales
54015604    =>  4   ;   //  Kettle
54015605    =>  4   ;   //  Pots
54015606    =>  4   ;   //  Pressure Cookers
54015607    =>  4   ;   //  Saucepan
54015608    =>  4   ;   //  Tauaga (coconut strainer)
54015609    =>  4   ;   //  Other Non Electric Kitchen Ustensils Or Ned
54015701    =>  4   ;   //  Gas Lighter
54015702    =>  4   ;   //  Gas Lighter, Bic
54015703    =>  4   ;   //  Bottle Racks
54015704    =>  4   ;   //  Buckets
54015705    =>  4   ;   //  Containers For Bread, Coffee, Spices
54015706    =>  4   ;   //  Feeding Bottles
54015707    =>  4   ;   //  Iceboxes
54015708    =>  4   ;   //  Ironing Boards
54015709    =>  4   ;   //  Laundry Baskets
54015710    =>  4   ;   //  Letter Boxes
54015711    =>  4   ;   //  Portable Money Boxes
54015712    =>  4   ;   //  Thermos Flasks
54015713    =>  4   ;   //  Towel Rails
54015714    =>  4   ;   //  Waste Bins
54015715    =>  4   ;   //  Waste Paper Baskets
54015716    =>  4   ;   //  Scales, Counter 20Kg
54015717    =>  4   ;   //  Hand Basin With Brackets (For Toilets)
54015718    =>  4   ;   //  Jug, Plastic,
54015719    =>  4   ;   //  Plastic Jerry Cans, 10 Litres
54015720    =>  4   ;   //  Tarpaulin,
54015721    =>  4   ;   //  Tarpaulin Clear (Heavy Plastic)
54015722    =>  4   ;   //  Other Non Electric Hh Article Or Ned
54015801    =>  2   ;   //  Repair Of Household Article
55115901    =>  3   ;   //  Cement Mixer Petrol
55115902    =>  3   ;   //  Water Pump Electric Orange Sj400.
55115903    =>  3   ;   //  Compression
55115904    =>  3   ;   //  Engine Cleaning Gun
55115905    =>  3   ;   //  V Belts ( For Concrete Mixers)
55115906    =>  3   ;   //  Brush Cutters
55115907    =>  3   ;   //  Chainsaws
55115908    =>  3   ;   //  Cultivators
55115909    =>  3   ;   //  Electric Drills
55115910    =>  3   ;   //  Garden Tractors
55115911    =>  3   ;   //  Lawnmowers
55115912    =>  3   ;   //  Motor Mowers
55115913    =>  3   ;   //  Sanders And Hedge Cutters
55115914    =>  3   ;   //  Other Motorized Tools And Equipment
55116001    =>  2   ;   //  Repair Of Such Articles
55216101    =>  3   ;   //  Dustmask
```

```
55216102 => 3 ; // Hacksaws
55216103 => 3 ; // Hand Saw
55216104 => 3 ; // Petrol Pump
55216105 => 3 ; // Circular Saw
55216106 => 3 ; // Clamp Tool
55216107 => 3 ; // Air Duster Gun
55216108 => 3 ; // Coach Screws
55216109 => 3 ; // Dyna Bolts
55216110 => 3 ; // Crowbar
55216111 => 3 ; // Cut Off Wheel 230X2.5X22Mm
55216112 => 3 ; // Hacksaw
55216113 => 3 ; // Matic Guns, Standard # Skeleton
55216114 => 3 ; // Multimeter Pocket Digital
55216115 => 3 ; // Sandpaper
55216116 => 3 ; // Spray Gun Suction
55216117 => 3 ; // Trowel
55216118 => 3 ; // Wire Brush
55216119 => 3 ; // Files
55216120 => 3 ; // Hammers
55216121 => 3 ; // Pliers
55216122 => 3 ; // Rasps
55216123 => 3 ; // Saws
55216124 => 3 ; // Screwdrivers
55216125 => 3 ; // Spanners
55216126 => 3 ; // Trimming Knives
55216127 => 3 ; // Wrenches
55216128 => 3 ; // Other Hand Tools
55216201 => 3 ; // Hose 1With Fittings
55216202 => 3 ; // Cane Knives, Bush Knives
55216203 => 3 ; // Forks
55216204 => 3 ; // Hoses And Sprinklers
55216205 => 3 ; // Rakes
55216206 => 3 ; // Scythes
55216207 => 3 ; // Shovels
55216208 => 3 ; // Sickles
55216209 => 3 ; // Spades
55216210 => 3 ; // Watering Cans
55216211 => 3 ; // Wheelbarrows
55216212 => 3 ; // Other Garden Tools
55216301 => 3 ; // Ladders And Steps
55216401 => 3 ; // Entrance Door Lock Set
55216402 => 3 ; // Hasp & Staple
55216403 => 3 ; // Security Door Locks C/W Knob/Handles
55216404 => 3 ; // Security Entrance Door Locks/Knob Handles
55216405 => 3 ; // Toilet Door Lock Set
55216406 => 3 ; // Brass Padlocks
55216407 => 3 ; // Pad Bolt
55216408 => 3 ; // Padlock
55216409 => 3 ; // Handles
55216410 => 3 ; // Hinges
55216411 => 3 ; // Security mesh
55216412 => 3 ; // Other security materials ned
55216501 => 3 ; // Shower Curtain Rods
55216502 => 3 ; // Toilet Paper Holder
55216503 => 3 ; // Toilet Seat Covers
55216504 => 3 ; // Carpet Rods
55216505 => 3 ; // Curtain Rails
55216506 => 3 ; // Hooks
55216507 => 3 ; // Other Metal Article For The House
55216601 => 3 ; // Chains
55216602 => 3 ; // Grids
55216603 => 3 ; // Hoop Segments For Fencing And Bordering
55216604 => 3 ; // Stakes
55216605 => 3 ; // Tie wire (included chicken wire)
55216606 => 3 ; // Mesh, meshwire
55216607 => 3 ; // Other Metal Article For The Garden
55216701 => 3 ; // Battery (Sk-N100)
55216702 => 3 ; // Battery (Sk-N70Zl)
55216703 => 3 ; // Battery, Alkaline
55216704 => 3 ; // Battery, Energizer
55216705 => 3 ; // Battery, Eveready
```

```
55216706    =>  3   ;   //  Battery, Magnacell
55216707    =>  3   ;   //  Battery, Panasonic
55216708    =>  3   ;   //  Battery Ned
55216709    =>  3   ;   //  Energy Saver Lights
55216710    =>  3   ;   //  Extension Cords
55216711    =>  3   ;   //  Fluorescent Light
55216712    =>  3   ;   //  Fluorescent Tube
55216713    =>  3   ;   //  Light Bulb
55216714    =>  3   ;   //  Torches Powered Diving Waterproof
55216715    =>  3   ;   //  Spark Plugs
55216716    =>  3   ;   //  Alarms
55216717    =>  3   ;   //  Bells
55216718    =>  3   ;   //  Flashlights
55216719    =>  3   ;   //  Fluorescent Tubes starters
55216720    =>  3   ;   //  Hand Lamps
55216721    =>  3   ;   //  Power Sockets
55216722    =>  3   ;   //  Switches
55216723    =>  3   ;   //  Torches
55216724    =>  3   ;   //  Wiring Flex
55216725    =>  3   ;   //  Other Small Electric Appliance
55216801    =>  2   ;   //  Repair Of Such Articles
56116901    =>  1   ;   //  Air Freshener
56116902    =>  1   ;   //  Dish Washing Liquid
56116903    =>  1   ;   //  Dish Washing Liquid, Joy
56116904    =>  1   ;   //  Disinfectant, Dettol
56116905    =>  1   ;   //  Disinfectant: Pine-O-Cleen
56116906    =>  1   ;   //  Fly Spray Surface, Mortein
56116907    =>  1   ;   //  Fly Spray, Red, Mortein
56116908    =>  1   ;   //  Insect Repellent, Aeroguard
56116909    =>  1   ;   //  Rat Trap
56116910    =>  1   ;   //  Soap, Laundry, Pacific
56116911    =>  1   ;   //  Soap, Laundry, Sudso
56116912    =>  1   ;   //  Toilet Cleaner, Harpic
56116914    =>  1   ;   //  Washing Powder, Boom
56116915    =>  1   ;   //  Washing Powder, Gain
56116916    =>  1   ;   //  Washing Powder, Lerosa
56116917    =>  1   ;   //  Washing Powder, Surf
56116918    =>  1   ;   //  Washing Powder, Tasi
56116919    =>  1   ;   //  Cleaning Paste Axion
56116920    =>  1   ;   //  Cold Power
56116921    =>  1   ;   //  Fabric Softener, Downy
56116922    =>  1   ;   //  Moth Balls
56116923    =>  1   ;   //  Napisan Powder
56116924    =>  1   ;   //  Bleach, Janola
56116925    =>  1   ;   //  Clorox
56116926    =>  1   ;   //  Clorox, Janola,
56116927    =>  1   ;   //  Vaselin
56116928    =>  1   ;   //  Other Cleaning And Maintenance Products
56117001    =>  1   ;   //  Sponge
56117002    =>  1   ;   //  Sponge Mop With Handle
56117003    =>  1   ;   //  Dustpan And Brush (2 Piece)
56117004    =>  1   ;   //  Mop Complete
56117005    =>  1   ;   //  Mops & Buckets
56117006    =>  1   ;   //  Toilet Cleaning Brushes C/W Stand
56117008    =>  1   ;   //  Brooms
56117009    =>  1   ;   //  Duster
56117010    =>  1   ;   //  Wired steelo
56117010    =>  1   ;   //  Other Articles For Cleaning
56117101    =>  1   ;   //  Aluminium Foil
56117102    =>  1   ;   //  Foam Cups
56117103    =>  1   ;   //  Foam Plates
56117104    =>  1   ;   //  Napkins, Paper
56117105    =>  1   ;   //  Plastic Cups
56117106    =>  1   ;   //  Plastic Dinner Plates
56117107    =>  1   ;   //  Plastic Spoons
56117108    =>  1   ;   //  Table Cloth, Plastic,
56117109    =>  1   ;   //  Filter
56117110    =>  1   ;   //  Vaccum Cleaner Bags
56117111    =>  1   ;   //  Kitchen Paper, Tissue, Paper Towel
56117112    =>  1   ;   //  Plastic Bin Liner
56117113    =>  1   ;   //  Other Paper Product
```

```
56117201    =>  1   ;   //  Candle
56117202    =>  1   ;   //  Adhesive Tapes
56117203    =>  1   ;   //  Camphor
56117204    =>  1   ;   //  String rope
56117205    =>  1   ;   //  Clothes Hangers
56117206    =>  1   ;   //  Fire Extinguishers For Households
56117207    =>  1   ;   //  Glues
56117208    =>  1   ;   //  Incense Sticks
56117209    =>  1   ;   //  Knitting Needles
56117210    =>  1   ;   //  Lamp Wicks
56117211    =>  1   ;   //  Anchor Bolts M12 X 300
56117212    =>  1   ;   //  Rivet
56117213    =>  1   ;   //  Nail plate
56117214    =>  1   ;   //  Screws
56117215    =>  1   ;   //  Nail
56117216    =>  1   ;   //  Rubber gloves
56117222    =>  1   ;   //  PVC glue
56117224    =>  1   ;   //  Hinges
56117225    =>  1   ;   //  Matches
56117226    =>  1   ;   //  Measuring Tape
56117227    =>  1   ;   //  Mosquito Coil
56117228    =>  1   ;   //  Mosquito Coil, Zap
56117237    =>  1   ;   //  Packaging Tape
56117241    =>  1   ;   //  Cloth Pegs Hard Plastic
56117244    =>  1   ;   //  Other Non Durable Articles
56217301    =>  2   ;   //  Butlers
56217302    =>  2   ;   //  Cooks
56217303    =>  2   ;   //  Drivers
56217304    =>  2   ;   //  Gardeners
56217305    =>  2   ;   //  Governesses
56217306    =>  2   ;   //  Maids
56217307    =>  2   ;   //  Secretaries
56217308    =>  2   ;   //  Tutors
56217309    =>  2   ;   //  Baby sitting / child minding services provided at home
56217310    =>  2   ;   //  Elderly care services provided at home
56217398    =>  2   ;   //  Other Domestic Services - not elsewhere classified (nec)
56217399    =>  2   ;   //  Other Domestic Services - not enough detail (ned)
56217401    =>  2   ;   //  Domestic services, including baby-sitting and housework,
supplied by enterprises or self-employed persons
56217501    =>  2   ;   //  Pest control, fumigation
56217502    =>  2   ;   //  Lawn mowing services
56217503    =>  2   ;   //  Other household services provided by external company to the
household
56217601    =>  2   ;   //  Dry-cleaning, laundering and dyeing of household linen,
household textiles carpets
56217701    =>  2   ;   //  Hire of furniture, equipment (e.g. hire of cement mixer…)
61117801    =>  1   ;   //  Prescription medicine
61117901    =>  1   ;   //  Tablets, Panadol
61117902    =>  1   ;   //  Vicks Vaporub
61117903    =>  1   ;   //  Micreme
61117904    =>  1   ;   //  Panadol
61117905    =>  1   ;   //  Other Patent Medecine
61118001    =>  1   ;   //  Cod Liver Oil
61118002    =>  1   ;   //  Halibut Liver Oil
61118003    =>  1   ;   //  Oral Contraceptives
61118004    =>  1   ;   //  Serums And Vaccines
61118005    =>  1   ;   //  Vitamins And Minerals
61118006    =>  1   ;   //  Other Pharmaceutical Products Or Ned
61218101    =>  1   ;   //  Band Aids, Jj
61218102    =>  1   ;   //  Bandages
61218103    =>  1   ;   //  Clinical Thermometers
61218104    =>  1   ;   //  Condoms
61218105    =>  1   ;   //  First Aid Kits
61218106    =>  1   ;   //  Hot Water Bottles
61218107    =>  1   ;   //  Hypodermic Syringes
61218108    =>  1   ;   //  Ice Bags
61218109    =>  1   ;   //  Mechanical Contraceptive Devises
61218110    =>  1   ;   //  Medical Hosiery Items
61218111    =>  1   ;   //  Pregnancy Tests
61218112    =>  1   ;   //  Other Medical Product
61318201    =>  3   ;   //  Artificial Limbs And Other Prosthetic Devices
```

```
61318202    =>  3   ;   //  Corrective Eye Glasses And Contact Lenses
61318203    =>  3   ;   //  Crutches
61318204    =>  3   ;   //  Dentures Excluding Fitting Costs
61318205    =>  3   ;   //  Devices For Monitoring Blood Pressure
61318206    =>  3   ;   //  Glass Eyes
61318207    =>  3   ;   //  Health Lamps
61318208    =>  3   ;   //  Hearing Aids
61318209    =>  3   ;   //  Medical Massage Equipment
61318210    =>  3   ;   //  Neck Braces
61318211    =>  3   ;   //  Orthopaedic Braces And Supports
61318212    =>  3   ;   //  Orthopaedic Footwear
61318213    =>  3   ;   //  'Special Beds'
61318214    =>  3   ;   //  Surgical Belts, Trusses And Supports
61318215    =>  3   ;   //  Wheelchairs – Powered And Unpowered And Invalid Carriers
61318216    =>  3   ;   //  Other Therapeutic Appliances
61318301    =>  2   ;   //  Repair or hire of therapeutic appliances
62118401    =>  2   ;   //  Consultations Of Doctor, Specialist
62218501    =>  2   ;   //  Consultations Of Dentist
62318601    =>  2   ;   //  Blood Test, Xray, Analysis
62318602    =>  2   ;   //  Nurse And Midwife
62318603    =>  2   ;   //  Other Paramedical Doctor
62318604    =>  2   ;   //  Other paramedical services (chiropractor…)
62318701    =>  2   ;   //  Traditional Healers
63018801    =>  2   ;   //  Hosptial, Clinic Fee
63018901    =>  2   ;   //  Medical service eg services of physicians
63019001    =>  2   ;   //  Paramedical services eg services of chiropractors,
optometrists
71119101    =>  3   ;   //  Car
71119102    =>  3   ;   //  Purchase of pick up truck, utility vehicle
71119103    =>  3   ;   //  Purchase of truck, mini bus, van
71219201    =>  3   ;   //  Motor Cycle
71219202    =>  3   ;   //  Scooter
71319301    =>  3   ;   //  Bicycle
71419401    =>  3   ;   //  Animal drawn vehicle
72119501    =>  4   ;   //  Aluminium Round Tubes For Boat
72119502    =>  4   ;   //  Fan Belt
72119503    =>  4   ;   //  Tyre
72119504    =>  4   ;   //  Tyre Tubes
72119505    =>  4   ;   //  Batteries For Vehicle
72119506    =>  4   ;   //  Tyre inflator
72119507    =>  4   ;   //  Other Spare Part For Vehicle And Boat
72219601    =>  1   ;   //  Fuel / gas for cars and land vehicle (private use)
72219602    =>  1   ;   //  Fuel / gas for boats and other water crafts (private use)
72219603    =>  1   ;   //  Fuel / gas for other (chainsaw, tillers, tractor…)
72219604    =>  1   ;   //  Fuel / gas without any details
72219605    =>  1   ;   //  Other fuel / gas for vehicle n e s
72219606    =>  1   ;   //
72219607    =>  1   ;   //
72219608    =>  1   ;   //
72219701    =>  1   ;   //  Oil Outboard
72219702    =>  1   ;   //  Oil Outboard Super
72219703    =>  1   ;   //  Oil Outboard Super, Ob
72219704    =>  1   ;   //  Oil Sae15/40
72219705    =>  1   ;   //  Oil Sae30
72219706    =>  1   ;   //  Oil, Air Compresser
72219707    =>  1   ;   //  Oil, Hydraulic
72219708    =>  1   ;   //  Marine Spray 4 Crc/Wd40
72219709    =>  1   ;   //  Gear Oil
72219710    =>  1   ;   //  Grease Cartridges, Multipurpose
72219711    =>  1   ;   //  Oil Lubricant Ned
72319801    =>  2   ;   //  Service For Maintenance Of Car/Boat, Mechanic
72419901    =>  2   ;   //  Driving lesson
72419902    =>  2   ;   //  Driving license fee
72419903    =>  2   ;   //  Registration fee
72419904    =>  2   ;   //  Other fee in respect of vehicle
72420001    =>  2   ;   //  Car Rental
72420002    =>  2   ;   //  Truck rental, hire
72420003    =>  2   ;   //  Mini bus, van rental, hire
72420004    =>  2   ;   //  Motorcycle / scooter rental, hire
72420005    =>  2   ;   //  Other vehicle rental n e s
73220101    =>  2   ;   //  Bus Fare
```

```
73220102    =>  2   ;   //  Taxi Fare
73220103    =>  2   ;   //  Other Transport Fare Ned
73320201    =>  2   ;   //  Air Fare
73320202    =>  2   ;   //  Air fare - domestic
73320203    =>  2   ;   //  Air fare without any detail
73420301    =>  2   ;   //  Boat Fare
73420302    =>  2   ;   //  Boat fare (travel overseas)
73420303    =>  2   ;   //  Other fees related to transport by boat n e s
73520401    =>  2   ;   //  Air fare and boat fare together
73620501    =>  2   ;   //  Freight, Delivery Charge
73620502    =>  2   ;   //  Other services related to transport
81120601    =>  2   ;   //  Payments For Delivery Of Letter, Postcards And Parcels
81120602    =>  2   ;   //  Private Mail And Parcel Delivery
81120603    =>  2   ;   //  Purchases Of Postage Stamps
81120604    =>  2   ;   //  Purchase Of Pre□Franked Postcards And Aerogramme
81120605    =>  2   ;   //  Other Postal Fee
82020701    =>  3   ;   //  Purcahse Of Cell (Mobile) Phones, Smartphone
82020702    =>  3   ;   //  Purchase Of Land □Line Telephones
82020703    =>  3   ;   //  Radio Telephones
82020704    =>  3   ;   //  Telefax Machines
82020705    =>  3   ;   //  Other Telephone And Telefax Equipment
82020801    =>  2   ;   //  Repair Of Such Equipment
83020901    =>  2   ;   //  Connection And Subscription Costs Of Cell (Mobile) Phones
83020902    =>  2   ;   //  Purchase Of Recharge Cards For Cell (Mobile) Phones
83020903    =>  2   ;   //  Purchase Of Sim Card
83020904    =>  2   ;   //  Installation And Subscription Costs Of Land□Line Telephone
Equipment
83020905    =>  2   ;   //  Internet Services
83020906    =>  2   ;   //  Other Telephone Services
91121001    =>  3   ;   //  Car Radios
91121002    =>  3   ;   //  Cd-Players
91121003    =>  3   ;   //  Microphones And Earphones
91121004    =>  3   ;   //  Radio Clocks
91121005    =>  3   ;   //  Radio Sets
91121006    =>  3   ;   //  Stereos And Their Constituent Units Eg Tuners, Amplifiers Etc
91121007    =>  3   ;   //  Television Aerials Of All Types
91121008    =>  3   ;   //  Television Sets
91121009    =>  3   ;   //  Two-Way Radios
91121010    =>  3   ;   //  Video Player/Vcd/Dvd
91121011    =>  3   ;   //  Mp3 Players, Ipods
91121012    =>  3   ;   //  Other Equipment
91221101    =>  3   ;   //  Cameras
91221102    =>  3   ;   //  Video camera
91221201    =>  3   ;   //  Film And Slide Projectors
91221301    =>  3   ;   //  Accessories for camera (flash, objective, lenses…)
91221401    =>  3   ;   //  Binoculars
91221402    =>  3   ;   //  Compass
91221403    =>  3   ;   //  Other Optical Instrument
91221404    =>  3   ;   //  Telescopes
91221405    =>  3   ;   //  Gps devices
91321501    =>  3   ;   //  Applications
91321502    =>  3   ;   //  Calculators
91321503    =>  3   ;   //  Computer Software Packages, E.G. Operating Systems
91321504    =>  3   ;   //  Personal Computers Desktop
91321505    =>  3   ;   //  Personal Computers Laptop
91321506    =>  3   ;   //  Printers And Miscellaneous Accessories Accompanying Them
91321507    =>  3   ;   //  Tablet
91321508    =>  3   ;   //  Other It Equipment
91421601    =>  4   ;   //  Dvd, Blue Ray
91421602    =>  4   ;   //  Cd Compact Disc
91421603    =>  4   ;   //  Other Recording Media
91421604    =>  4   ;   //  portable harddrives, flash drives
91521701    =>  2   ;   //  Repair Of Audio-Visual, Photographic And Information
Processing Equipment
92121801    =>  3   ;   //  Boat without outboard motor
92121802    =>  3   ;   //  Canoe
92121803    =>  3   ;   //  Outboard Motor
92121805    =>  3   ;   //  Dive gear (tanks, bc, regulators, etc.)
92121807    =>  3   ;   //  Water craft trailer
92121808    =>  3   ;   //  Other major durables for outdoor recreation
92121809    =>  3   ;   //  Boat with outboard motor
```

```
92221901    =>  3   ;   //  Guitare
92221902    =>  3   ;   //  Guitar Strings, Steel
92221903    =>  3   ;   //  Other Musical Instrument
92221904    =>  3   ;   //  Other musical instrument n e s
92222001    =>  3   ;   //  Billiard tables
92222002    =>  3   ;   //  Gaming machines
92222003    =>  3   ;   //  Pinball machines
92222004    =>  3   ;   //  Pin-pong table (table-tennis table)
92322101    =>  2   ;   //  Maintenance And Repair Of Other Major Durables For
Recreation And Culture
93122201    =>  4   ;   //  Card Games
93122202    =>  4   ;   //  Chess
93122203    =>  4   ;   //  Parlour Games
93122204    =>  4   ;   //  Video Game Console And Accessories
93122205    =>  4   ;   //  Other Games
93122301    =>  4   ;   //  Barbie Dolls
93122302    =>  4   ;   //  Cars (Police) Remote Control
93122303    =>  4   ;   //  Toys of all kinds (dolls, fireworks, toy cars, etc.)
93122304    =>  4   ;   //  Water Guns
93122305    =>  4   ;   //  X-Mas Lights
93122306    =>  4   ;   //  X-Mas Tree Decoration
93122307    =>  4   ;   //  Baby Dolls With Accessories
93122308    =>  4   ;   //  Other Toys
93122401    =>  4   ;   //  Other items for collection e.g. coins, medals, zoological
and botanical specimens etc.)
93122402    =>  4   ;   //  Stamp collecting requisites (used or cancelled postage
stamps, stamp albums etc.)
93222501    =>  4   ;   //  Beach Volley Ball
93222502    =>  4   ;   //  Cricket Ball
93222503    =>  4   ;   //  Other Sport Equipment
93222504    =>  4   ;   //  Golf clubs
93222505    =>  4   ;   //  Nets
93222506    =>  4   ;   //  Rackets
93222507    =>  4   ;   //  Shuttlecocks
93222508    =>  4   ;   //  Weights
93222509    =>  4   ;   //  Other equipment for sport n e s
93222601    =>  4   ;   //  Parachute and other sky-diving equipment
93222701    =>  4   ;   //  Firearms (gun, bullet…)
93222801    =>  4   ;   //  Fishing Lines
93222802    =>  4   ;   //  Speargun
93222803    =>  4   ;   //  Mask, Flippers
93222804    =>  4   ;   //  Other Fishing Equipments
93222901    =>  4   ;   //  Frisbee
93222902    =>  4   ;   //  Inflatable Boats
93222903    =>  4   ;   //  Inflatable Swimming Pools
93222904    =>  4   ;   //  Volleyball
93222905    =>  4   ;   //  Other Beach And Open Air Games
93223001    =>  4   ;   //  Air Mattresses And Inflating Pumps
93223002    =>  4   ;   //  Camping Stoves And Barbecues
93223003    =>  4   ;   //  Sleeping Bags
93223004    =>  4   ;   //  Tents And Accessories
93223101    =>  4   ;   //  Football Boots
93223102    =>  4   ;   //  Golfing Shoes
93223103    =>  4   ;   //  Other Such Footwear Fitted With Ice Skates, Rollers, Spikes,
Studs, Etc.
93223104    =>  4   ;   //  Rugby Boots
93223201    =>  4   ;   //  Belts
93223202    =>  4   ;   //  Body Padding
93223203    =>  4   ;   //  Boxing Gloves
93223204    =>  4   ;   //  Goggles
93223205    =>  4   ;   //  Headgear
93223206    =>  4   ;   //  Life Jackets
93223207    =>  4   ;   //  Shin Guards
93223208    =>  4   ;   //  Supports
93223301    =>  2   ;   //  Repair Of Such Articles
93323401    =>  1   ;   //  Fertilisers, Composts And Garden Peat
93323402    =>  1   ;   //  Horticultural Preparations
93323403    =>  1   ;   //  Other Garden Plants And Flower Stuff
93323404    =>  1   ;   //  Pots and plant holders
93323405    =>  1   ;   //  Other plants related items n e s
93423501    =>  1   ;   //  Purchase of pets (cat, dog,….)
```

```
93423601    =>  1   ;   //  Animal Feed
93423602    =>  1   ;   //  Pig feed, pig pellet, grower
93423603    =>  1   ;   //  Chicken feed
93423604    =>  1   ;   //  Other animal feed (including expired food purchased for
animal feed)
93423605    =>  1   ;   //  Other pet-related products (feed, leashes, vet fee, etc.)
93523701    =>  2   ;   //  Veterinary
93523801    =>  2   ;   //  Other Services Eg Training
94123901    =>  2   ;   //  Bowling Alleys
94123902    =>  2   ;   //  Fairgrounds And Amusement Parks
94123903    =>  2   ;   //  Fitness Centres
94123904    =>  2   ;   //  Golf Courses
94123905    =>  2   ;   //  Gymnasia
94123906    =>  2   ;   //  Horse Racing Courses
94123907    =>  2   ;   //  Motor Racing Circuits
94123908    =>  2   ;   //  Pin□Ball Machines (Excluding Game Of Chance)
94123909    =>  2   ;   //  Roundabouts, See□Saws And Other Playground
94123910    =>  2   ;   //  Facilities For Children
94123911    =>  2   ;   //  See□Saws And Playground Facilities For Children
94123912    =>  2   ;   //  Skating Rinks
94123913    =>  2   ;   //  Sports Stadium
94123914    =>  2   ;   //  Squash Courts
94123915    =>  2   ;   //  Swimming Pools
94123916    =>  2   ;   //  Tennis Courts
94123917    =>  2   ;   //  Velodromes
94124001    =>  2   ;   //  Hire of boat
94124002    =>  2   ;   //  Hire of other accessories for sport and recreation
94124101    =>  2   ;   //  Aerobics
94124102    =>  2   ;   //  Chess
94124103    =>  2   ;   //  Dancing
94124104    =>  2   ;   //  Music
94124105    =>  2   ;   //  Swimming
94124106    =>  2   ;   //  Other Group Lessons
94124201    =>  2   ;   //  Services Of Mountain Guides, Tour Guides Etc
94124301    =>  2   ;   //  Navigational Aid Services For Boating
94124401    =>  2   ;   //  Hire of game-specific footwear (excludes purchase (932))
94224501    =>  2   ;   //  Art Galleries And Exhibitions
94224502    =>  2   ;   //  Cinemas
94224503    =>  2   ;   //  Concert And Music Halls
94224504    =>  2   ;   //  Historical Monuments, National Parks, Zoological And
Botanical Gardens
94224505    =>  2   ;   //  And Aquaria
94224506    =>  2   ;   //  Libraries
94224507    =>  2   ;   //  Theatre
94224508    =>  2   ;   //  Cinema
94224509    =>  2   ;   //  Game At The Stadium
94224510    =>  2   ;   //  Other Services
94224601    =>  2   ;   //  Dvd rental, hire
94224701    =>  2   ;   //  Cable Tv Fee
94224702    =>  2   ;   //  Other Fee Related To Cable Tv
94224801    =>  2   ;   //  Services Of Photographers Eg Film Developing , Print
Processing, Enlarging, Portrait Photography, Wedding Photography
94324901    =>  2   ;   //  Bingo Card
94324902    =>  2   ;   //  Gambling Machine
94324903    =>  2   ;   //  Other Gamble Fee
95125001    =>  4   ;   //  Atlases
95125002    =>  4   ;   //  Dictionaries
95125003    =>  4   ;   //  Encyclopaedias
95125004    =>  4   ;   //  Guidebooks
95125005    =>  4   ;   //  Musical Scores
95125006    =>  4   ;   //  Novels, Story Books
95125007    =>  4   ;   //  Scrapbooks And Albums For Children
95125008    =>  4   ;   //  Textbooks, School Books
95125009    =>  4   ;   //  Other Book Related Expense
95125101    =>  2   ;   //  Bookbinding
95225201    =>  1   ;   //  Magazines
95225202    =>  1   ;   //  Newspapers
95225203    =>  1   ;   //  Periodicals
95325301    =>  1   ;   //  Greeting Cards, Visiting Card
95325302    =>  1   ;   //  Maps And Globes
95325303    =>  1   ;   //  Poster, Post Card…
```

```
95425401    =>  1   ;   //  Envelopes
95425402    =>  1   ;   //  Marking Pen
95425403    =>  1   ;   //  Pencil Case
95425404    =>  1   ;   //  Pencil Sharpener
95425405    =>  1   ;   //  Pens, Blue/Black, Ball Point
95425406    =>  1   ;   //  Pens, Blue/Blk/Red, Metal Point, Bic
95425407    =>  1   ;   //  Rubber (Eraser)
95425408    =>  1   ;   //  Carpenter’S Pencil
95425409    =>  1   ;   //  Carpenter'S Folding Ruler
95425410    =>  1   ;   //  Chalk Line Refill Blue Johnson
95425411    =>  1   ;   //  Crepe Paper
95425412    =>  1   ;   //  Dual Metric/Imperial 8/10M
95425413    =>  1   ;   //  Gift Wrapper
95425414    =>  1   ;   //  Sellotape, 2" Clear
95425415    =>  1   ;   //  Tialeniu School Badges-Jan. 2011
95425416    =>  1   ;   //  Other Stationery Or Ned
95425501    =>  1   ;   //  Ink Cartridge Brother
95425601    =>  1   ;   //  Exercise book
96025701    =>  2   ;   //  All Inclusive Holidays Or Tours Which Provide For Travel,
Food, Accommodation, Guides Etc
101025801   =>  2   ;   //  Kindergarden Fees
101025802   =>  2   ;   //  Primary School Fees
102025901   =>  2   ;   //  Secondary School Fees
103026001   =>  2   ;   //  Technical Education School Fees
103026002   =>  2   ;   //  Vocational Education School Fees
104026101   =>  2   ;   //  University Fees
105026201   =>  2   ;   //  Other Educational Program Fees
105026202   =>  2   ;   //  Tutoring for academic subjects (science….), private lesson
for school purposes
105026203   =>  2   ;   //  Other expenditure related to education (teacher day, sport
day, school events…)
111126301   =>  2   ;   //  Restaurant, Café
111126302   =>  2   ;   //  Bar Night Clubs
111126401   =>  2   ;   //  Ham, Sandwich, Local
111126402   =>  2   ;   //  Ham, Sandwich, Us
111126403   =>  2   ;   //  Other Take Away Food
111126501   =>  2   ;   //  Take away ice cream
111126502   =>  2   ;   //  Take away pizza
111126503   =>  2   ;   //  Cooked poultry, chicken take away
111126504   =>  2   ;   //  Fish and chips take away
111126505   =>  2   ;   //  Burger, cheese burger take away
111126506   =>  2   ;   //  Fried chicken take away
111126507   =>  2   ;   //  Meat pies take away
111126508   =>  2   ;   //  Sandwich take away
111126598   =>  2   ;   //  Other take away plate, take away food (chinese food,
chicken, barbecue, cassava, fish, sashimi ….)
111126599   =>  2   ;   //  Take away food without any detail
111126601   =>  2   ;   //  The sale of cooked dishes by catering contractors whether
collected by the customer or delivered to the customer’s home
111226701   =>  2   ;   //  School restaurant
112026801   =>  2   ;   //  Hotel, Motel, Guest House
112026802   =>  2   ;   //  Packaged accomidation with food (hotel)
112026901   =>  2   ;   //  Boarding Schools, Univ Compound
112027001   =>  2   ;   //  Accommodation services provided by others
121127101   =>  2   ;   //  Hairdresser
121127201   =>  2   ;   //  Body Care, Depilation (Waxing, Threading, Etc.) And The Like
121127202   =>  2   ;   //  Facials
121127203   =>  2   ;   //  Manicure
121127204   =>  2   ;   //  Non□Medical Massage
121127205   =>  2   ;   //  Pedicure
121127206   =>  2   ;   //  Turkish Baths, Saunas
121127207   =>  2   ;   //  Other Personal Grooming
121127208   =>  2   ;   //  Tattoo, piercings and other body art
121227301   =>  4   ;   //  Curling Tongs And Styling Combs
121227302   =>  4   ;   //  Electric Razors And Hair Trimmers
121227303   =>  4   ;   //  Electric Toothbrushes
121227304   =>  4   ;   //  Hand□Held And Hood Hair Dryers
121227305   =>  4   ;   //  Sunlamps
121227306   =>  4   ;   //  Other Electric Appliance For Personal Care
121227401   =>  2   ;   //  Repair Of Such Appliances
121327501   =>  1   ;   //  Razor Blades
```

```
121327502   =>  1   ;   //  Razor Refill, Gillette
121327503   =>  1   ;   //  Razor, Contour
121327504   =>  1   ;   //  Tooth Brush
121327505   =>  1   ;   //  Hair Brush
121327506   =>  1   ;   //  Nail Brush
121327507   =>  1   ;   //  Nail Files
121327508   =>  1   ;   //  Personal Scale, Baby Scale
121327509   =>  1   ;   //  Combs
121327510   =>  1   ;   //  Other Non Electric Appliance For Personal Care
121327601   =>  1   ;   //  Shampoo, Baby Shampoo
121327602   =>  1   ;   //  Hair Conditioner, Palmolive
121327603   =>  1   ;   //  Hair Gel, (Mousse)
121327604   =>  1   ;   //  Hair Shampoo, Palmolive
121327605   =>  1   ;   //  Shaving Cream, Nivea
121327606   =>  1   ;   //  Soap, Lux
121327607   =>  1   ;   //  Soap, Palmolive
121327608   =>  1   ;   //  Soap, Protex
121327609   =>  1   ;   //  Body Wash
121327610   =>  1   ;   //  Body Wash, Lux
121327611   =>  1   ;   //  Body Wash, Protex
121327612   =>  1   ;   //  Body Wash, Suave
121327613   =>  1   ;   //  Toothpaste, Colgate
121327614   =>  1   ;   //  Cleaning Milk
121327615   =>  1   ;   //  Soap Ned
121327616   =>  1   ;   //  Other Articles Of Personal Hygiene
121327701   =>  1   ;   //  Baby Oil, 200Mls, Johnson
121327702   =>  1   ;   //  Baby Powder
121327703   =>  1   ;   //  Baby Powder, Johnsons
121327704   =>  1   ;   //  Body Lotion, Nivea
121327705   =>  1   ;   //  Deodorant Lynx Spray, Men
121327706   =>  1   ;   //  Deodorant Lynx Spray, Women
121327707   =>  1   ;   //  Deodorant Mum Roll-On, Dove (For Ladies)
121327708   =>  1   ;   //  Deodorant Mum Roll-On, Rexona, (For Ladies)
121327709   =>  1   ;   //  Deodorant Mum Roll-On, Rexona,(For Men)
121327710   =>  1   ;   //  Deodorant Spray, (Men), Rexona
121327711   =>  1   ;   //  Deodorant Spray, (Women), Rexona
121327712   =>  1   ;   //  Deodorant Spray, For Men, Nivea
121327713   =>  1   ;   //  Deodorant Spray, For Women, Nivea
121327714   =>  1   ;   //  Deodorant, Impulse
121327715   =>  1   ;   //  Deodorant, Lynx Roll-On, Men
121327716   =>  1   ;   //  Perfume
121327717   =>  1   ;   //  Make Up, Lipstick
121327718   =>  1   ;   //  Other Beauty Product
121327801   =>  1   ;   //  Baby Wipes, Huggies
121327802   =>  1   ;   //  Cotton Buds, Ear Buds
121327803   =>  1   ;   //  Ladies Pads, Johnsons
121327804   =>  1   ;   //  Ladies Pads, Libra
121327805   =>  1   ;   //  Ladies Pads, Modess
121327806   =>  1   ;   //  Ladies Pads, Stay Free
121327807   =>  1   ;   //  Nappies, Huggies
121327808   =>  1   ;   //  Nappies, Soft Love
121327809   =>  1   ;   //  Toilet Paper, Sofan
121327810   =>  1   ;   //  Toilet Paper, Swan
121327811   =>  1   ;   //  Toilet Paper, Wondersoft
121327812   =>  1   ;   //  Cotton
121327813   =>  1   ;   //  Nappies for adults, Depends
121327814   =>  1   ;   //  Other Product
122027901   =>  2   ;   //  Services provided by prostitutes and the like
123128001   =>  3   ;   //  Diamon, gold, platnum, and similar precious metals and stones
123128101   =>  3   ;   //  Jewellery (local or not): earing, necklace, bracelet
123128102   =>  3   ;   //  Artificial flowers, artificial tiare
123128103   =>  3   ;   //  Jewellery without any detail
123128104   =>  3   ;   //  Fresh flower arrangments in bouquet, lei, ei, head ea, ea
katu, poe tiare
123128201   =>  3   ;   //  Clocks
123128202   =>  3   ;   //  Watch
123128203   =>  3   ;   //  Alarm Clock
123128204   =>  3   ;   //  Other Clock And Watch Related Article
123128301   =>  2   ;   //  Repair Of Such Articles
123228401   =>  4   ;   //  Handbag
123228402   =>  4   ;   //  Suit Case
```

```
    123228403   =>  4   ;   //  School Backpacks, (Boys)
    123228404   =>  4   ;   //  School Backpacks, (Girls)
    123228405   =>  4   ;   //  Other Travel Goods
    123228501   =>  4   ;   //  Baby Carriages, Carrycots
    123228502   =>  4   ;   //  Car Beds And Seats
    123228503   =>  4   ;   //  Pushchairs
    123228504   =>  4   ;   //  Reins And Harnesses
    123228505   =>  4   ;   //  Other Articles For Babies
    123228601   =>  4   ;   //  Lighter
    123228602   =>  4   ;   //  Pipe
    123228603   =>  4   ;   //  Ashtray
    123228604   =>  4   ;   //  Other Article For Smokers
    123228701   =>  4   ;   //  Umbrella
    123228702   =>  4   ;   //  Umbrella (Beach)
    123228703   =>  4   ;   //  Fan
    123228704   =>  4   ;   //  Sunglasses
    123228705   =>  4   ;   //  Fey Ring
    123228706   =>  4   ;   //  Other Miscellaneous Personal Article
    123228707   =>  4   ;   //  traditional crafts - fan
    123228708   =>  4   ;   //  traditional crafts - handbags, bags, pouches
    123228709   =>  4   ;   //  traditional crafts - hats
    123228710   =>  4   ;   //  traditional crafts - mats
    123228801   =>  4   ;   //  Funerary Articles Eg Coffins, Gravestones, Urns
    123228901   =>  2   ;   //  Repair Of Such Articles
    124029001   =>  2   ;   //  Retirement Homes For Elderly Persons And Residences For
    Disabled Persons
    124029101   =>  2   ;   //  Home help, day care and rehabilitation
    125129201   =>  2   ;   //  Life Assurance
    125129202   =>  2   ;   //  Education Assurance
    125229301   =>  2   ;   //  Home Insurance
    125329401   =>  2   ;   //  Health Insurance
    125429501   =>  2   ;   //  Car Insurance
    125429502   =>  2   ;   //  Boat Insurance
    125429503   =>  2   ;   //  Other Insurance Related To Transport
    125529601   =>  2   ;   //  Personal Insurance
    126129701   =>  2   ;   //  Financial intermediation services indirectly measured
    126229801   =>  2   ;   //  Banking Fees
    126229802   =>  2   ;   //  Money Transfer Fees
    126229803   =>  2   ;   //  Credit Card Fee
    126229804   =>  2   ;   //  Other Financial Fees
    127029901   =>  2   ;   //  Passport Fees
    127029902   =>  2   ;   //  Administrative Document, Certificate (Birth, Death…)
    127029903   =>  2   ;   //  Photocopies
    127029904   =>  2   ;   //  Fees related to the purchase of a house (appraisal, realty,
    etc.)
    127029905   =>  2   ;   //  Legal expenses
    127029907   =>  2   ;   //  Other fees related to personal services
    901100101-999999999 =>  5   ;   //  Current cash transfers: birthdays
endrecode;


if M_COICOP in 1:970000000 then
    numeric tmp2;
    numeric tmp = noccurs(EXPENDITURE_EDT)+1;
    insert(EXPENDITURE_EDT(tmp));

    E_LINE_NUMBER(tmp) = tmp;
    E_HIES_CODE(tmp) = M_HIES_CODE;
    E_COICOP(tmp) = M_COICOP;
    E_AMOUNT(tmp) = M_AMOUNT;
    E_AN_AMT(tmp) = M_AN_AMT;
    E_WGHT_AN_AMT(tmp) = M_WGHT_AN_AMT;
    E_MULTIPLIER(tmp) = M_MULTIPLIER;
    E_LOCATION(tmp) = M_LOCATION;
    E_PROD_TYPE(tmp) = M_PROD_TYPE;
    E_NUMBER(tmp) = M_QUANTITY;
    E_UNIT(tmp) = M_UNIT;
    E_CATEGORY(tmp) = notappl;
    E_BENEFICIARY(tmp) = M_BENEFICIARY;
    E_EXP_TYPE(tmp) = M_EXP_TYPE;
    E_DURABILITY(tmp) = tmpDura;
```

```
    E_DESCRIPTION(tmp) = tolower(M_DESCRIPTION);




    //income side
    if M_EXP_TYPE in 3 then
        numeric tmpCOICOP = M_COICOP;
        numeric tmpPACCOI = notappl; // using code below
        numeric tmpHIEScode = M_HIES_CODE; // using code below
        numeric tmpMulti = M_MULTIPLIER;
        numeric tmpISCO = notappl; // not necessary for this section
        numeric tmpISIC = notappl; // not necessary for this seciton
        numeric tmpValue = M_AMOUNT;
        numeric tmpIncomeType = M_EXP_TYPE; //[1]cash  [2]home  [3]in-kind bought
        [4]in-kind home  [5]rent  [6]intermediate  [7]in-kind income
        numeric tmpLocation = M_LOCATION;
        numeric tmpBeneficiary = M_BENEFICIARY;
        string tmpDescription = tolower(M_DESCRIPTION);
        numeric tmpProductType = M_PROD_TYPE;
        numeric tmpQuantity = notappl;
        numeric tmpNumber = M_QUANTITY;
        numeric tmpUnit = M_UNIT;
        numeric tmpCategory = notappl;
        numeric tmpDurability = tmpDura;

        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));

        I_LINE_NUMBER(tmp) = tmp;
        I_HIES_CODE(tmp) = tmpHIEScode;
        I_PACCOI(tmp) = tmpPACCOI;
        I_COICOP(tmp) = tmpCOICOP;
        I_GROSS(tmp) = tmpValue;
        I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
        I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
        I_MULTIPLIER(tmp) = tmpMulti;
        I_LOCATION(tmp) = tmpLocation;
        I_INCOME_TYPE(tmp) = tmpIncomeType;
        I_PROD_TYPE(tmp) = tmpProductType;
        I_QUANTITY(tmp) = tmpQuantity;
        I_NUMBER(tmp) = tmpNumber;
        I_UNIT(tmp) = tmpUnit;
        I_CATEGORY(tmp) = tmpCategory;
        I_BENEFICIARY(tmp) = tmpBeneficiary;
        I_DURABILITY(tmp) = tmpDurability;
        I_INDUSTRY(tmp) = tmpISIC;
        I_OCCUPATION(tmp) = tmpISCO;
        I_DESCRIPTION(tmp) = tmpDescription;
    endif;
endif;
PROC P_LINE_NUMBER

// first pass coicop correction on 21/3/2016
if ID07 in  102 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'sky f' then
impute(P_COICOP,   11100518    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'custa' then
impute(P_COICOP,   11403301    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  102 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'noddl' then
impute(P_COICOP,   11100906    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'sonic' then
impute(P_COICOP,   83020905    );  endif;
if ID07 in  102 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  103 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'donat' then
```

```
impute(P_COICOP,   901100501   );  endif;
if ID07 in  103 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504003    );  endif;
if ID07 in  103 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  104 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  104 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  104 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  105 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'nappy' then
impute(P_COICOP,   56116923    );  endif;
if ID07 in  105 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'ice b' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  106 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'donny' then
impute(P_COICOP,   56116921    );  endif;
if ID07 in  106 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'musty' then
impute(P_COICOP,   11906305    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'seiko' then
impute(P_COICOP,   11100204    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'gas b' then
impute(P_COICOP,   45212702    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  108 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'commu' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  108 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'plane' then
impute(P_COICOP,   73320201    );  endif;
if ID07 in  108 and P_LINE_NUMBER in    52  and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'confl' then
impute(P_COICOP,   11101004    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'cold'  then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'gift'  then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  109 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'chops' then
impute(P_COICOP,   11100919    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  109 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  109 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  111 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'detol' then
impute(P_COICOP,   56116904    );  endif;
if ID07 in  111 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'janol' then
impute(P_COICOP,   56116924    );  endif;
if ID07 in  111 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'soap'  then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'tooth' then
impute(P_COICOP,   121327613   );  endif;
```

```
if ID07 in  111 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  111 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  111 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  113 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100202    );  endif;
if ID07 in  113 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'crc'   then
impute(P_COICOP,   72219708    );  endif;
if ID07 in  113 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'nappy' then
impute(P_COICOP,   56116923    );  endif;
if ID07 in  113 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  113 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  114 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  114 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504003    );  endif;
if ID07 in  114 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  114 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'baki'  then
impute(P_COICOP,   11906603    );  endif;
if ID07 in  114 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = '1x bo' then
impute(P_COICOP,   12207101    );  endif;
if ID07 in  116 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = '1x pa' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  116 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'cocoa' then
impute(P_COICOP,   12106901    );  endif;
if ID07 in  116 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'custa' then
impute(P_COICOP,   11403301    );  endif;
if ID07 in  116 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  116 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  116 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  116 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  116 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  116 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  117 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'gift'  then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  117 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  118 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  118 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  119 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  119 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  119 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'chill' then
impute(P_COICOP,   11906302    );  endif;
if ID07 in  119 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'custa' then
impute(P_COICOP,   11403301    );  endif;
if ID07 in  119 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  120 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  120 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  120 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  120 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = '1x sp' then
impute(P_COICOP,   21107611    );  endif;
if ID07 in  120 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'donat' then
```

```
impute(P_COICOP,   901100601   );  endif;
if ID07 in  120 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  120 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  121 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'tined' then
impute(P_COICOP,   11302501    );  endif;
if ID07 in  121 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = '1x sp' then
impute(P_COICOP,   21107611    );  endif;
if ID07 in  121 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  121 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805802    );  endif;
if ID07 in  121 and P_LINE_NUMBER in    44  and strip(P_DESCRIPTION[1:5]) = 'dnill' then
impute(P_COICOP,   11906302    );  endif;
if ID07 in  121 and P_LINE_NUMBER in    50  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  124 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  124 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  124 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'zap 6' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  124 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'beef'  then
impute(P_COICOP,   11201107    );  endif;
if ID07 in  124 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  124 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504002    );  endif;
if ID07 in  124 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'funur' then
impute(P_COICOP,   123228801   );  endif;
if ID07 in  125 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  125 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  125 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  126 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  126 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'chip'  then
impute(P_COICOP,   43111817    );  endif;
if ID07 in  126 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'famil' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  126 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  126 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  127 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'zap 8' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'zap 8' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'highl' then
impute(P_COICOP,   11402801    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'zap 8' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'zap 6' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'week'  then
impute(P_COICOP,   11101005    );  endif;
if ID07 in  127 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  127 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  132 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'roll'  then
impute(P_COICOP,   121327708   );  endif;
if ID07 in  132 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'perto' then
impute(P_COICOP,   72219602    );  endif;
```

```
if ID07 in  132 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'dishw' then
impute(P_COICOP,   56116903    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'sausa' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'detto' then
impute(P_COICOP,   56116904    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805802    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  132 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  134 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'butte' then
impute(P_COICOP,   11503601    );  endif;
if ID07 in  134 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805802    );  endif;
if ID07 in  134 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'raro'  then
impute(P_COICOP,   12207501    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116906    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'axion' then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'soap'  then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  135 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'weet'  then
impute(P_COICOP,   11101005    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'tooth' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  135 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'match' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'boat'  then
impute(P_COICOP,   73420301    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'spiri' then
impute(P_COICOP,   21107611    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  135 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'slice' then
impute(P_COICOP,   11403404    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'butte' then
```

```
impute(P_COICOP,   11503601    );  endif;
if ID07 in  135 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'cigs'  then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  136 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  138 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'jam'   then
impute(P_COICOP,   11805002    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'cigs'  then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'irish' then
impute(P_COICOP,   11202003    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'potat' then
impute(P_COICOP,   11704601    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'tooth' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  138 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'tinne' then
impute(P_COICOP,   11302501    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'tomat' then
impute(P_COICOP,   11906310    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  138 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  138 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'butte' then
impute(P_COICOP,   11503601    );  endif;
if ID07 in  138 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'chill' then
impute(P_COICOP,   11906302    );  endif;
if ID07 in  139 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  140 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'crack' then
impute(P_COICOP,   11100514    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'cocoa' then
impute(P_COICOP,   12106901    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'napis' then
impute(P_COICOP,   56116923    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'noodl' then
impute(P_COICOP,   11100919    );  endif;
```

```
if ID07 in  140 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'softl' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  140 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'janda' then
impute(P_COICOP,   32110201    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  140 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'condi' then
impute(P_COICOP,   121327602   );  endif;
if ID07 in  140 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  140 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  140 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'vermi' then
impute(P_COICOP,   11100917    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'jiff'  then
impute(P_COICOP,   56116928    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'vienn' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'sweet' then
impute(P_COICOP,   11100506    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'chewi' then
impute(P_COICOP,   11805603    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'm&m'   then
impute(P_COICOP,   11805501    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'tea b' then
impute(P_COICOP,   12106801    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'peanu' then
impute(P_COICOP,   11503901    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  141 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  141 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  142 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  143 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
```

```
impute(P_COICOP,   11100102    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'salte' then
impute(P_COICOP,   11201103    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  143 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'corn'  then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11403004    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  143 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'biscu' then
impute(P_COICOP,   11100518    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'chips' then
impute(P_COICOP,   11704804    );  endif;
if ID07 in  143 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'bottl' then
impute(P_COICOP,   12207101    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'cold'  then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'sweet' then
impute(P_COICOP,   11906303    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'cold'  then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'paste' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'bongo' then
impute(P_COICOP,   11906606    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'prote' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
```

```
if ID07 in  147 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11402605    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'baby'  then
impute(P_COICOP,   121327701   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'axio'  then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11402605    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'cold'  then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11402605    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'coco'  then
impute(P_COICOP,   11101003    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'soap'  then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  147 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    44  and strip(P_DESCRIPTION[1:5]) = 'axion' then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    46  and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    48  and strip(P_DESCRIPTION[1:5]) = 'cornb' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    50  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    51  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    52  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    53  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  147 and P_LINE_NUMBER in    54  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'cerea' then
```

```
impute(P_COICOP,   11101006    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'axion' then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'axio'  then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'condi' then
impute(P_COICOP,   121327602   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'bongo' then
impute(P_COICOP,   11906606    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'sao b' then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'noodl' then
impute(P_COICOP,   11100919    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  148 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  148 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'lamp'  then
impute(P_COICOP,   11201301    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402604    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'cold'  then
impute(P_COICOP,   56116920    );  endif;
```

```
if ID07 in  150 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327615   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'axion' then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'chips' then
impute(P_COICOP,   11704804    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'tinne' then
impute(P_COICOP,   11302501    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'vermi' then
impute(P_COICOP,   11100917    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'yeast' then
impute(P_COICOP,   11906614    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'bakin' then
impute(P_COICOP,   11906603    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'nutel' then
impute(P_COICOP,   11805901    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'corn'  then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'liali' then
impute(P_COICOP,   11100917    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'prote' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  150 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'chops' then
impute(P_COICOP,   11201301    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'sausa' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'bongo' then
impute(P_COICOP,   11906606    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  150 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'donat' then
```

```
impute(P_COICOP,   901100501   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'pinea' then
impute(P_COICOP,   11604408    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'downy' then
impute(P_COICOP,   56116921    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'rexon' then
impute(P_COICOP,   121327711   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'ancho' then
impute(P_COICOP,   11402604    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'chill' then
impute(P_COICOP,   11906302    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'wipes' then
impute(P_COICOP,   121327801   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'tooth' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'detto' then
impute(P_COICOP,   56116904    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'match' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'cloro' then
impute(P_COICOP,   56116925    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
```

```
if ID07 in  151 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'paste' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  151 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'napis' then
impute(P_COICOP,   56116923    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    44  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'cloro' then
impute(P_COICOP,   56116925    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    46  and strip(P_DESCRIPTION[1:5]) = 'axio'  then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'fly s' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  151 and P_LINE_NUMBER in    48  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  152 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'elect' then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'lamb'  then
impute(P_COICOP,   11201301    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'water' then
impute(P_COICOP,   12207101    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'butte' then
impute(P_COICOP,   11503601    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'snack' then
impute(P_COICOP,   11100518    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'fuel'  then
impute(P_COICOP,   45312803    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  152 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  152 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'dishw' then
impute(P_COICOP,   56116903    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'tinne' then
impute(P_COICOP,   11302501    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'sweet' then
impute(P_COICOP,   11100501    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'paste' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'axion' then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'prote' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302507    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'janda' then
impute(P_COICOP,   32110201    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'vienn' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'curry' then
impute(P_COICOP,   11906103    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'donat' then
```

```
impute(P_COICOP,   901100501   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cabba' then
impute(P_COICOP,   11704517    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'mutto' then
impute(P_COICOP,   11201303    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'tea b' then
impute(P_COICOP,   12106801    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'sausa' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'butte' then
impute(P_COICOP,   11503601    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'nappy' then
impute(P_COICOP,   121327807   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  154 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'cornf' then
impute(P_COICOP,   11101004    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402604    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    44  and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'sao b' then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    46  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  154 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
```

```
if ID07 in  155 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'mayon' then
impute(P_COICOP,   11906402    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'noodl' then
impute(P_COICOP,   11100919    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'soya'  then
impute(P_COICOP,   11906306    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'cornf' then
impute(P_COICOP,   11101004    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'ancho' then
impute(P_COICOP,   11402603    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'coffe' then
impute(P_COICOP,   12106701    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'raro'  then
impute(P_COICOP,   12207501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'raro'  then
impute(P_COICOP,   12207501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'spagh' then
impute(P_COICOP,   11906612    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'soap'  then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116902    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'telep' then
impute(P_COICOP,   83020904    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'cornf' then
impute(P_COICOP,   11101004    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'tinne' then
impute(P_COICOP,   11302501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    33  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    36  and strip(P_DESCRIPTION[1:5]) = 'mayon' then
impute(P_COICOP,   11906402    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'cerea' then
impute(P_COICOP,   11101006    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'flour' then
```

```
impute(P_COICOP,   11100203    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11402603    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'sweet' then
impute(P_COICOP,   11100501    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'paste' then
impute(P_COICOP,   121327613   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'tooth' then
impute(P_COICOP,   121327504   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    44  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    46  and strip(P_DESCRIPTION[1:5]) = 'potat' then
impute(P_COICOP,   11704601    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    48  and strip(P_DESCRIPTION[1:5]) = 'cornb' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    50  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    51  and strip(P_DESCRIPTION[1:5]) = 'bakin' then
impute(P_COICOP,   11906603    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    52  and strip(P_DESCRIPTION[1:5]) = 'axio'  then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  155 and P_LINE_NUMBER in    53  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  155 and P_LINE_NUMBER in    54  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'cornb' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'shamp' then
impute(P_COICOP,   121327604   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'batte' then
impute(P_COICOP,   55216704    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'vienn' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'sao b' then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'butte' then
impute(P_COICOP,   11503601    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'chops' then
impute(P_COICOP,   11201301    );  endif;
```

```
if ID07 in  158 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'tea b' then
impute(P_COICOP,   12106801    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'ancho' then
impute(P_COICOP,   11402604    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'gas l' then
impute(P_COICOP,   123228601   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  158 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'cocoa' then
impute(P_COICOP,   12106901    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  158 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'match' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'milk'  then
impute(P_COICOP,   11403004    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'cocoa' then
impute(P_COICOP,   12106901    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  159 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'gift'  then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  159 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'cooki' then
impute(P_COICOP,   11504001    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'patet' then
impute(P_COICOP,   11704601    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'milo'  then
impute(P_COICOP,   12107001    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'sao'   then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'raro'  then
impute(P_COICOP,   12207501    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'coco'  then
impute(P_COICOP,   11101003    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'gas c' then
impute(P_COICOP,   45212702    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'coco'  then
impute(P_COICOP,   11101003    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'cornf' then
impute(P_COICOP,   11101004    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'tea b' then
```

```
impute(P_COICOP,   12106801    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'bongo' then
impute(P_COICOP,   11906606    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'downy' then
impute(P_COICOP,   56116921    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'axio'  then
impute(P_COICOP,   56116919    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  159 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  160 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'tin f' then
impute(P_COICOP,   11302509    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'flour' then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'sugar' then
impute(P_COICOP,   11804902    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'chops' then
impute(P_COICOP,   11201301    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'juice' then
impute(P_COICOP,   12207301    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'vienn' then
impute(P_COICOP,   11201906    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'tuna'  then
impute(P_COICOP,   11302508    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'cup n' then
impute(P_COICOP,   11100905    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'cigs'  then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  160 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'curry' then
impute(P_COICOP,   11906103    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402604    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'soaps' then
impute(P_COICOP,   121327608   );  endif;
if ID07 in  160 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'sack'  then
impute(P_COICOP,   11100203    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'chick' then
impute(P_COICOP,   11201502    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'morte' then
impute(P_COICOP,   56116907    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'cocon' then
impute(P_COICOP,   11403201    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'sao b' then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'downy' then
impute(P_COICOP,   56116921    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'match' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'pine'  then
impute(P_COICOP,   56116905    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  160 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'corn'  then
impute(P_COICOP,   11101004    );  endif;
```

```
if ID07 in  160 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'long'  then
impute(P_COICOP,   11402607    );  endif;
if ID07 in  160 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'yeast' then
impute(P_COICOP,   11906614    );  endif;
if ID07 in  203 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  203 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  204 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'cabin' then
impute(P_COICOP,   11100514    );  endif;
if ID07 in  206 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'rices' then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  206 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100401   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'zap-f' then
impute(P_COICOP,   12207506    );  endif;
if ID07 in  206 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  206 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  207 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  207 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  208 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  209 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  209 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  209 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  209 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  209 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'janol' then
impute(P_COICOP,   56116924    );  endif;
if ID07 in  209 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  210 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  210 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  210 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'sunsh' then
impute(P_COICOP,   11403005    );  endif;
if ID07 in  210 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'break' then
impute(P_COICOP,   11100514    );  endif;
if ID07 in  210 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  211 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'stell' then
impute(P_COICOP,   56117010    );  endif;
if ID07 in  211 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'janol' then
impute(P_COICOP,   56116924    );  endif;
if ID07 in  211 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  211 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  213 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  217 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  218 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  218 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'donat' then
```

```
impute(P_COICOP,   901100501   );  endif;
if ID07 in  219 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100401    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'zap x' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    38  and strip(P_DESCRIPTION[1:5]) = 'phone' then
impute(P_COICOP,   82020801    );  endif;
if ID07 in  219 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100401    );  endif;
if ID07 in  220 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  220 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  220 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'hotdo' then
impute(P_COICOP,   111126402   );  endif;
if ID07 in  220 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  220 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100601   );  endif;
if ID07 in  221 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  221 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  221 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'tv sk' then
impute(P_COICOP,   94224701    );  endif;
if ID07 in  221 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  222 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  222 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'fried' then
impute(P_COICOP,   11100302    );  endif;
if ID07 in  223 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'phone' then
impute(P_COICOP,   83020904    );  endif;
if ID07 in  223 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'luau'  then
impute(P_COICOP,   11704533    );  endif;
if ID07 in  223 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  223 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'sunsh' then
impute(P_COICOP,   11403005    );  endif;
if ID07 in  223 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  223 and P_LINE_NUMBER in    45  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  223 and P_LINE_NUMBER in    47  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  224 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'sunsh' then
impute(P_COICOP,   11403005    );  endif;
if ID07 in  224 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'ice b' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  224 and P_LINE_NUMBER in    46  and strip(P_DESCRIPTION[1:5]) = 'nutri' then
impute(P_COICOP,   11101006    );  endif;
if ID07 in  224 and P_LINE_NUMBER in    65  and strip(P_DESCRIPTION[1:5]) = 'nutri' then
impute(P_COICOP,   11101006    );  endif;
if ID07 in  224 and P_LINE_NUMBER in    69  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  226 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'donti' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  226 and P_LINE_NUMBER in    43  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  227 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'sunsh' then
impute(P_COICOP,   11403005    );  endif;
if ID07 in  227 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  232 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
```

```
if ID07 in  232 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'doant' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  235 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  236 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'zap x' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  236 and P_LINE_NUMBER in    48  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  236 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'soup'  then
impute(P_COICOP,   11100919    );  endif;
if ID07 in  236 and P_LINE_NUMBER in    75  and strip(P_DESCRIPTION[1:5]) = '2 bot' then
impute(P_COICOP,   56116904    );  endif;
if ID07 in  237 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'telep' then
impute(P_COICOP,   83020904    );  endif;
if ID07 in  237 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  237 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'light' then
impute(P_COICOP,   123228601   );  endif;
if ID07 in  238 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'twist' then
impute(P_COICOP,   11906613    );  endif;
if ID07 in  238 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  238 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'twist' then
impute(P_COICOP,   11906613    );  endif;
if ID07 in  238 and P_LINE_NUMBER in    49  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  239 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'paper' then
impute(P_COICOP,   56117106    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'dobie' then
impute(P_COICOP,   56116921    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  239 and P_LINE_NUMBER in    48  and strip(P_DESCRIPTION[1:5]) = 'stir'  then
impute(P_COICOP,   11704533    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    55  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    59  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    62  and strip(P_DESCRIPTION[1:5]) = 'nutri' then
impute(P_COICOP,   11101006    );  endif;
if ID07 in  239 and P_LINE_NUMBER in    73  and strip(P_DESCRIPTION[1:5]) = 'donti' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  240 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'pall'  then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  240 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'zap'   then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  240 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  240 and P_LINE_NUMBER in    51  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  241 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  241 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'doant' then
impute(P_COICOP,   901100201   );  endif;
if ID07 in  241 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  241 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'curry' then
impute(P_COICOP,   11906103    );  endif;
if ID07 in  241 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  301 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'sausa' then
impute(P_COICOP,   11201904    );  endif;
if ID07 in  301 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  301 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  301 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  301 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'foil'  then
```

```
impute(P_COICOP,   56117101    );  endif;
if ID07 in  301 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'baby'  then
impute(P_COICOP,   121327807   );  endif;
if ID07 in  301 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'baby'  then
impute(P_COICOP,   121327801   );  endif;
if ID07 in  301 and P_LINE_NUMBER in    32  and strip(P_DESCRIPTION[1:5]) = 'cigar' then
impute(P_COICOP,   22108202    );  endif;
if ID07 in  302 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'vaili' then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  302 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   901100101   );  endif;
if ID07 in  304 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'match' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  305 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  305 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  305 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'chair' then
impute(P_COICOP,   51113203    );  endif;
if ID07 in  305 and P_LINE_NUMBER in    10  and strip(P_DESCRIPTION[1:5]) = 'elect' then
impute(P_COICOP,   43211901    );  endif;
if ID07 in  306 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'taro'  then
impute(P_COICOP,   11704603    );  endif;
if ID07 in  306 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = '40 li' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  306 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  306 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  306 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'inter' then
impute(P_COICOP,   83020905    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'morti' then
impute(P_COICOP,   56116906    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'shopp' then
impute(P_COICOP,   73420301    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'meat'  then
impute(P_COICOP,   11201301    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'rice'  then
impute(P_COICOP,   11100102    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'morti' then
impute(P_COICOP,   56116906    );  endif;
if ID07 in  307 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'sonic' then
impute(P_COICOP,   83020905    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'can t' then
impute(P_COICOP,   11906309    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'corn'  then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'salt'  then
impute(P_COICOP,   11906002    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'neck'  then
impute(P_COICOP,   11201302    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  308 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'vaili' then
impute(P_COICOP,   21308003    );  endif;
if ID07 in  309 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'water' then
impute(P_COICOP,   12207101    );  endif;
if ID07 in  309 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'packe' then
impute(P_COICOP,   121327807   );  endif;
if ID07 in  309 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  309 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'keros' then
impute(P_COICOP,   45312801    );  endif;
if ID07 in  309 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'panad' then
impute(P_COICOP,   61117901    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'adult' then
impute(P_COICOP,   121327813   );  endif;
```

```
if ID07 in  310 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'telep' then
impute(P_COICOP,   83020904    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    11  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  310 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'raro'  then
impute(P_COICOP,   12207501    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'packe' then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'tin b' then
impute(P_COICOP,   11100518    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'bowl'  then
impute(P_COICOP,   11100901    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'nouil' then
impute(P_COICOP,   11100901    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'potat' then
impute(P_COICOP,   11704801    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'twist' then
impute(P_COICOP,   11906613    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'vaili' then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    28  and strip(P_DESCRIPTION[1:5]) = 'fishi' then
impute(P_COICOP,   93222804    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  310 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  311 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'crtn'  then
impute(P_COICOP,   56117225    );  endif;
if ID07 in  311 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  311 and P_LINE_NUMBER in    21  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  311 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'pacif' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  318 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  318 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'fruit' then
impute(P_COICOP,   11604401    );  endif;
if ID07 in  318 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'churc' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  319 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'detto' then
impute(P_COICOP,   56116904    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'ice b' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'sand'  then
impute(P_COICOP,   72420001    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    54  and strip(P_DESCRIPTION[1:5]) = 'timbe' then
impute(P_COICOP,   43111811    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    55  and strip(P_DESCRIPTION[1:5]) = 'cemen' then
impute(P_COICOP,   43111802    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    56  and strip(P_DESCRIPTION[1:5]) = 'plywo' then
impute(P_COICOP,   43111809    );  endif;
if ID07 in  319 and P_LINE_NUMBER in    57  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  320 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  320 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  320 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308003    );  endif;
if ID07 in  320 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = 'vaili' then
impute(P_COICOP,   21308003    );  endif;
if ID07 in  322 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'pure'  then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  322 and P_LINE_NUMBER in    13  and strip(P_DESCRIPTION[1:5]) = 'pure'  then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  322 and P_LINE_NUMBER in    15  and strip(P_DESCRIPTION[1:5]) = 'cash'  then
impute(P_COICOP,   45112501    );  endif;
if ID07 in  322 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = '20l p' then
```

```
impute(P_COICOP,   72219602    );  endif;
if ID07 in  323 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'telev' then
impute(P_COICOP,   91121008    );  endif;
if ID07 in  323 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'telep' then
impute(P_COICOP,   83020904    );  endif;
if ID07 in  323 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'tin o' then
impute(P_COICOP,   11201904    );  endif;
if ID07 in  323 and P_LINE_NUMBER in    17  and strip(P_DESCRIPTION[1:5]) = 'corne' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  324 and P_LINE_NUMBER in    2   and strip(P_DESCRIPTION[1:5]) = 'fruit' then
impute(P_COICOP,   11604401    );  endif;
if ID07 in  324 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'hot d' then
impute(P_COICOP,   111126402   );  endif;
if ID07 in  324 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'buns'  then
impute(P_COICOP,   11100304    );  endif;
if ID07 in  324 and P_LINE_NUMBER in    27  and strip(P_DESCRIPTION[1:5]) = 'fruit' then
impute(P_COICOP,   11604401    );  endif;
if ID07 in  324 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'puta'  then
impute(P_COICOP,   11100304    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    22  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'donat' then
impute(P_COICOP,   901100501   );  endif;
if ID07 in  326 and P_LINE_NUMBER in    29  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    34  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    35  and strip(P_DESCRIPTION[1:5]) = '2 bot' then
impute(P_COICOP,   12207101    );  endif;
if ID07 in  326 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  327 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  152 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'bough' then
impute(P_COICOP,   999999999   );  endif;
if ID07 in  327 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'cocon' then
impute(P_COICOP,   11403201    );  endif;
if ID07 in  327 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = '2 can' then
impute(P_COICOP,   11202001    );  endif;
if ID07 in  327 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = '2 coo' then
impute(P_COICOP,   11504002    );  endif;
if ID07 in  327 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'car p' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  327 and P_LINE_NUMBER in    41  and strip(P_DESCRIPTION[1:5]) = '40 li' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    1   and strip(P_DESCRIPTION[1:5]) = 'can c' then
impute(P_COICOP,   11403201    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = '20l p' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'chees' then
impute(P_COICOP,   11906607    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'onion' then
impute(P_COICOP,   11704521    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'bottl' then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    24  and strip(P_DESCRIPTION[1:5]) = 'mixed' then
impute(P_COICOP,   11704519    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    30  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'beer'  then
impute(P_COICOP,   21308004    );  endif;
if ID07 in  328 and P_LINE_NUMBER in    42  and strip(P_DESCRIPTION[1:5]) = 'cakes' then
impute(P_COICOP,   11100601    );  endif;
if ID07 in  332 and P_LINE_NUMBER in    4   and strip(P_DESCRIPTION[1:5]) = 'loaf'  then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  332 and P_LINE_NUMBER in    7   and strip(P_DESCRIPTION[1:5]) = 'sack'  then
impute(P_COICOP,   11704601    );  endif;
```

```
if ID07 in  332 and P_LINE_NUMBER in    12  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  332 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'bread' then
impute(P_COICOP,   11100301    );  endif;
if ID07 in  337 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'loade' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  337 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'loade' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  337 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'loade' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  339 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'petro' then
impute(P_COICOP,   72219602    );  endif;
if ID07 in  341 and P_LINE_NUMBER in    6   and strip(P_DESCRIPTION[1:5]) = 'sport' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  341 and P_LINE_NUMBER in    16  and strip(P_DESCRIPTION[1:5]) = 'sport' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  341 and P_LINE_NUMBER in    25  and strip(P_DESCRIPTION[1:5]) = 'washi' then
impute(P_COICOP,   56116920    );  endif;
if ID07 in  341 and P_LINE_NUMBER in    40  and strip(P_DESCRIPTION[1:5]) = 'sport' then
impute(P_COICOP,   12207205    );  endif;
if ID07 in  342 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'fruit' then
impute(P_COICOP,   11906616    );  endif;
if ID07 in  342 and P_LINE_NUMBER in    26  and strip(P_DESCRIPTION[1:5]) = 'zaps'  then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  342 and P_LINE_NUMBER in    37  and strip(P_DESCRIPTION[1:5]) = 'fruit' then
impute(P_COICOP,   11906616    );  endif;
if ID07 in  343 and P_LINE_NUMBER in    5   and strip(P_DESCRIPTION[1:5]) = 'baked' then
impute(P_COICOP,   11100302    );  endif;
if ID07 in  346 and P_LINE_NUMBER in    8   and strip(P_DESCRIPTION[1:5]) = 'drill' then
impute(P_COICOP,   55216128    );  endif;
if ID07 in  346 and P_LINE_NUMBER in    18  and strip(P_DESCRIPTION[1:5]) = 'truck' then
impute(P_COICOP,   72420001    );  endif;
if ID07 in  347 and P_LINE_NUMBER in    39  and strip(P_DESCRIPTION[1:5]) = 'janol' then
impute(P_COICOP,   56116924    );  endif;
if ID07 in  348 and P_LINE_NUMBER in    3   and strip(P_DESCRIPTION[1:5]) = 'sao'   then
impute(P_COICOP,   11100512    );  endif;
if ID07 in  349 and P_LINE_NUMBER in    14  and strip(P_DESCRIPTION[1:5]) = 'writi' then
impute(P_COICOP,   95425416    );  endif;
if ID07 in  349 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'orang' then
impute(P_COICOP,   11604203    );  endif;
if ID07 in  352 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'mosqu' then
impute(P_COICOP,   56117228    );  endif;
if ID07 in  353 and P_LINE_NUMBER in    23  and strip(P_DESCRIPTION[1:5]) = 'ice c' then
impute(P_COICOP,   11805803    );  endif;
if ID07 in  354 and P_LINE_NUMBER in    9   and strip(P_DESCRIPTION[1:5]) = 'napis' then
impute(P_COICOP,   56116923    );  endif;
if ID07 in  354 and P_LINE_NUMBER in    19  and strip(P_DESCRIPTION[1:5]) = 'toile' then
impute(P_COICOP,   121327809   );  endif;
if ID07 in  354 and P_LINE_NUMBER in    20  and strip(P_DESCRIPTION[1:5]) = 'gas l' then
impute(P_COICOP,   123228601   );  endif;
if ID07 in  354 and P_LINE_NUMBER in    31  and strip(P_DESCRIPTION[1:5]) = 'plate' then
impute(P_COICOP,   111126403   );  endif;

PROC P_HIES_CODE
// second pass coicop
recode  ID07    :   P_LINE_NUMBER   =>  P_COICOP    ;
    102 :   1   =>  11804903    ;
    102 :   3   =>  11804903    ;
    102 :   8   =>  11403302    ;
    102 :   12  =>  11403302    ;
    102 :   25  =>  11100101    ;
    102 :   29  =>  901100302   ;
    103 :   6   =>  11804903    ;
    103 :   19  =>  901100301   ;
    103 :   35  =>  11804903    ;
    104 :   8   =>  11403302    ;
    104 :   27  =>  901100301   ;
    104 :   28  =>  11804903    ;
    104 :   31  =>  901100302   ;
    105 :   6   =>  11804903    ;
    106 :   1   =>  11804903    ;
```

```
106 :   35  =>  11804903    ;
106 :   44  =>  11100101    ;
106 :   49  =>  11804903    ;
108 :   2   =>  11804903    ;
108 :   8   =>  11403302    ;
108 :   16  =>  999999998   ;
108 :   17  =>  901100301   ;
108 :   20  =>  11403302    ;
108 :   28  =>  901100601   ;
108 :   42  =>  999999998   ;
108 :   49  =>  999999998   ;
108 :   52  =>  11403302    ;
109 :   12  =>  11403302    ;
109 :   21  =>  11804903    ;
109 :   30  =>  11100101    ;
109 :   37  =>  11403302    ;
109 :   45  =>  901100302   ;
109 :   49  =>  901100302   ;
111 :   3   =>  999999998   ;
111 :   9   =>  901100301   ;
111 :   10  =>  901100301   ;
111 :   13  =>  901100110   ;
111 :   17  =>  11403302    ;
111 :   24  =>  11403302    ;
111 :   31  =>  901100110   ;
111 :   37  =>  999999998   ;
113 :   10  =>  11804903    ;
113 :   21  =>  901100301   ;
113 :   24  =>  999999998   ;
114 :   7   =>  901100301   ;
114 :   8   =>  11804903    ;
114 :   14  =>  11100101    ;
116 :   19  =>  901100301   ;
116 :   21  =>  901100102   ;
116 :   22  =>  11804903    ;
116 :   27  =>  901100302   ;
116 :   28  =>  901100302   ;
116 :   29  =>  901100302   ;
117 :   2   =>  901100110   ;
117 :   6   =>  901100302   ;
118 :   4   =>  901100302   ;
118 :   10  =>  901100302   ;
119 :   4   =>  11403302    ;
119 :   16  =>  901100302   ;
119 :   30  =>  11804903    ;
119 :   32  =>  901100302   ;
120 :   2   =>  901100301   ;
120 :   7   =>  901100301   ;
120 :   8   =>  11403302    ;
120 :   9   =>  11804903    ;
120 :   20  =>  901100301   ;
121 :   6   =>  11302509    ;
121 :   9   =>  11403501    ;
121 :   10  =>  999999998   ;
121 :   18  =>  11804903    ;
121 :   20  =>  11906002    ;
121 :   26  =>  901100301   ;
121 :   50  =>  901100301   ;
124 :   6   =>  901100601   ;
124 :   8   =>  11804903    ;
124 :   9   =>  11906002    ;
124 :   12  =>  901100601   ;
124 :   14  =>  11403302    ;
124 :   22  =>  901100301   ;
124 :   36  =>  11804903    ;
125 :   28  =>  901100102   ;
125 :   31  =>  901100301   ;
126 :   3   =>  901100302   ;
126 :   6   =>  901100110   ;
126 :   9   =>  11804903    ;
126 :   15  =>  901100301   ;
```

```
126 :   22  =>  11804903    ;
126 :   26  =>  901100302   ;
127 :   2   =>  11804903    ;
127 :   6   =>  11403302    ;
127 :   14  =>  11403302    ;
127 :   21  =>  11804903    ;
127 :   23  =>  11403302    ;
127 :   29  =>  11403302    ;
127 :   32  =>  11403302    ;
127 :   36  =>  11403302    ;
127 :   39  =>  11101005    ;
127 :   47  =>  901100110   ;
127 :   49  =>  901100302   ;
132 :   6   =>  11804903    ;
132 :   13  =>  901100302   ;
132 :   17  =>  11100101    ;
132 :   28  =>  901100302   ;
134 :   14  =>  11804903    ;
134 :   42  =>  11804903    ;
134 :   47  =>  901100302   ;
134 :   50  =>  901100302   ;
135 :   8   =>  11804903    ;
135 :   23  =>  11804903    ;
136 :   7   =>  11804903    ;
136 :   16  =>  901100301   ;
136 :   22  =>  901100301   ;
138 :   1   =>  11804903    ;
138 :   16  =>  11100101    ;
138 :   26  =>  11804903    ;
138 :   30  =>  901100302   ;
139 :   2   =>  11804903    ;
140 :   16  =>  11100101    ;
140 :   21  =>  11804903    ;
141 :   17  =>  11804903    ;
142 :   3   =>  11504005    ;
142 :   11  =>  11804903    ;
142 :   13  =>  901100301   ;
142 :   23  =>  901100301   ;
143 :   5   =>  901100301   ;
143 :   6   =>  11804903    ;
143 :   8   =>  11100101    ;
147 :   1   =>  11804903    ;
147 :   9   =>  901100110   ;
147 :   15  =>  901100110   ;
147 :   20  =>  11403302    ;
147 :   25  =>  11100101    ;
147 :   30  =>  11100101    ;
147 :   31  =>  11804903    ;
147 :   35  =>  901100110   ;
147 :   36  =>  901100301   ;
147 :   38  =>  11804903    ;
147 :   39  =>  11100101    ;
147 :   51  =>  11100101    ;
147 :   54  =>  901100302   ;
148 :   3   =>  11100101    ;
148 :   13  =>  11804903    ;
148 :   15  =>  11100101    ;
148 :   17  =>  901100302   ;
148 :   19  =>  11403302    ;
148 :   30  =>  11804903    ;
148 :   36  =>  901100110   ;
148 :   37  =>  901100302   ;
150 :   6   =>  11403302    ;
150 :   11  =>  11804903    ;
150 :   15  =>  11100101    ;
150 :   20  =>  901100302   ;
150 :   32  =>  11403302    ;
150 :   33  =>  11100101    ;
150 :   35  =>  11403501    ;
150 :   40  =>  901100302   ;
151 :   2   =>  11100101    ;
```

```
151 :   19  =>  901100301   ;
151 :   20  =>  11804903    ;
151 :   22  =>  11100101    ;
151 :   30  =>  11403501    ;
151 :   34  =>  901100110   ;
151 :   35  =>  11100101    ;
151 :   38  =>  901100110   ;
151 :   39  =>  11804903    ;
151 :   48  =>  901100301   ;
154 :   14  =>  901100302   ;
154 :   16  =>  11403501    ;
154 :   33  =>  11804903    ;
154 :   40  =>  901100302   ;
155 :   5   =>  11804903    ;
155 :   12  =>  11100101    ;
155 :   17  =>  11804903    ;
155 :   20  =>  901100302   ;
155 :   26  =>  901100110   ;
155 :   37  =>  11804903    ;
155 :   45  =>  901100110   ;
155 :   47  =>  11804903    ;
155 :   53  =>  901100110   ;
155 :   54  =>  901100302   ;
158 :   2   =>  11804903    ;
158 :   6   =>  11100101    ;
158 :   12  =>  901100301   ;
158 :   27  =>  901100110   ;
158 :   28  =>  901100301   ;
159 :   4   =>  11403302    ;
159 :   5   =>  901100301   ;
159 :   6   =>  901100110   ;
159 :   14  =>  11100101    ;
159 :   20  =>  11403302    ;
159 :   21  =>  11804903    ;
159 :   30  =>  901100302   ;
160 :   3   =>  11804903    ;
160 :   5   =>  11100101    ;
160 :   16  =>  901100301   ;
160 :   24  =>  11504005    ;
160 :   25  =>  11100512    ;
160 :   29  =>  901100302   ;
201 :   2   =>  11804903    ;
203 :   1   =>  901100102   ;
203 :   3   =>  901100102   ;
204 :   3   =>  11804903    ;
204 :   20  =>  11804903    ;
206 :   2   =>  11100101    ;
206 :   6   =>  901100102   ;
206 :   7   =>  901100602   ;
206 :   19  =>  901100102   ;
206 :   22  =>  901100301   ;
206 :   29  =>  901100301   ;
207 :   1   =>  901100102   ;
207 :   2   =>  901100102   ;
208 :   18  =>  901100301   ;
209 :   5   =>  901100102   ;
209 :   17  =>  11804903    ;
209 :   23  =>  901100301   ;
209 :   29  =>  11403302    ;
209 :   42  =>  901100301   ;
210 :   10  =>  901100102   ;
210 :   12  =>  901100301   ;
210 :   23  =>  901100110   ;
210 :   28  =>  11804903    ;
210 :   40  =>  901100302   ;
211 :   29  =>  11504005    ;
211 :   32  =>  901100301   ;
213 :   14  =>  11804903    ;
218 :   2   =>  11804903    ;
218 :   23  =>  901100302   ;
219 :   19  =>  45513001    ;
```

```
219 :   22  =>  45513001    ;
219 :   27  =>  11403302    ;
219 :   32  =>  45513001    ;
219 :   38  =>  83020904    ;
221 :   27  =>  901100302   ;
222 :   9   =>  901100301   ;
222 :   13  =>  11100402    ;
223 :   18  =>  901100302   ;
223 :   30  =>  901100301   ;
223 :   45  =>  901100302   ;
223 :   47  =>  901100301   ;
224 :   30  =>  45513001    ;
224 :   39  =>  11804903    ;
224 :   69  =>  901100302   ;
226 :   24  =>  901100302   ;
226 :   36  =>  11804903    ;
226 :   43  =>  901100302   ;
227 :   15  =>  11403302    ;
232 :   17  =>  901100301   ;
232 :   20  =>  11804903    ;
232 :   25  =>  901100301   ;
233 :   2   =>  11804903    ;
235 :   8   =>  11504005    ;
235 :   20  =>  901100302   ;
236 :   29  =>  11403302    ;
236 :   48  =>  901100302   ;
236 :   57  =>  11804903    ;
237 :   5   =>  11403302    ;
237 :   11  =>  11804903    ;
238 :   25  =>  901100302   ;
238 :   49  =>  901100302   ;
239 :   20  =>  11403302    ;
239 :   29  =>  11504005    ;
239 :   35  =>  901100302   ;
239 :   55  =>  11403302    ;
239 :   59  =>  11403302    ;
239 :   68  =>  11804903    ;
239 :   73  =>  901100302   ;
240 :   21  =>  11804903    ;
240 :   23  =>  11403302    ;
240 :   29  =>  901100301   ;
240 :   49  =>  11804903    ;
240 :   51  =>  901100301   ;
241 :   12  =>  901100102   ;
241 :   14  =>  11804903    ;
241 :   23  =>  901100301   ;
241 :   35  =>  901100301   ;
246 :   4   =>  11504005    ;
246 :   7   =>  901100301   ;
246 :   8   =>  901100302   ;
246 :   9   =>  901100302   ;
246 :   10  =>  11504005    ;
246 :   11  =>  901100102   ;
246 :   12  =>  901100102   ;
246 :   13  =>  901100301   ;
247 :   16  =>  11804903    ;
249 :   9   =>  901100301   ;
249 :   30  =>  901100302   ;
250 :   14  =>  901100601   ;
250 :   15  =>  901100302   ;
250 :   27  =>  901100601   ;
250 :   28  =>  901100302   ;
250 :   34  =>  11201512    ;
250 :   43  =>  45513001    ;
250 :   44  =>  999999998   ;
250 :   45  =>  901100301   ;
250 :   46  =>  11804903    ;
250 :   56  =>  901100301   ;
252 :   14  =>  901100301   ;
252 :   21  =>  11804903    ;
252 :   24  =>  901100301   ;
```

```
253 :   13  =>  901100301   ;
253 :   16  =>  11804903    ;
253 :   19  =>  901100110   ;
253 :   20  =>  901100302   ;
253 :   21  =>  901100302   ;
254 :   12  =>  11804903    ;
254 :   21  =>  11804903    ;
254 :   41  =>  11403302    ;
254 :   58  =>  121327808   ;
255 :   14  =>  11804903    ;
255 :   20  =>  11101004    ;
256 :   27  =>  11804903    ;
256 :   35  =>  11804903    ;
256 :   44  =>  999999998   ;
256 :   45  =>  11201512    ;
256 :   60  =>  11201512    ;
256 :   75  =>  11704704    ;
256 :   82  =>  901100301   ;
258 :   17  =>  11804903    ;
258 :   25  =>  901100301   ;
302 :   4   =>  901100110   ;
302 :   7   =>  901100301   ;
304 :   1   =>  56117225    ;
304 :   20  =>  901100301   ;
304 :   21  =>  11504005    ;
305 :   5   =>  901100301   ;
305 :   14  =>  901100301   ;
306 :   1   =>  11201103    ;
306 :   23  =>  11804903    ;
307 :   9   =>  999999998   ;
307 :   10  =>  901100301   ;
307 :   13  =>  901100602   ;
307 :   16  =>  11100101    ;
308 :   7   =>  11804903    ;
309 :   18  =>  901100302   ;
309 :   23  =>  11804903    ;
309 :   36  =>  901100301   ;
310 :   11  =>  901100302   ;
310 :   31  =>  901100301   ;
311 :   12  =>  901100302   ;
311 :   16  =>  11804903    ;
311 :   35  =>  901100302   ;
318 :   10  =>  11804903    ;
318 :   21  =>  83020904    ;
318 :   22  =>  901100302   ;
319 :   13  =>  901100302   ;
319 :   36  =>  11804903    ;
319 :   59  =>  901100302   ;
320 :   9   =>  11804903    ;
320 :   16  =>  901100302   ;
320 :   17  =>  901100301   ;
320 :   26  =>  901100301   ;
322 :   7   =>  901100302   ;
322 :   8   =>  11804903    ;
322 :   16  =>  901100302   ;
323 :   5   =>  11804903    ;
323 :   7   =>  901100302   ;
324 :   19  =>  901100302   ;
324 :   20  =>  901100302   ;
324 :   22  =>  11100401    ;
324 :   30  =>  11100401    ;
324 :   31  =>  901100302   ;
326 :   22  =>  21308004    ;
326 :   25  =>  901100302   ;
326 :   34  =>  21308004    ;
326 :   42  =>  901100302   ;
327 :   15  =>  901100301   ;
327 :   16  =>  21308004    ;
327 :   30  =>  11504005    ;
327 :   42  =>  901100302   ;
328 :   1   =>  11504005    ;
```

```
    328 :   13  =>  901100301   ;
    328 :   16  =>  11804903    ;
    328 :   33  =>  11804903    ;
    328 :   44  =>  901100301   ;
    329 :   5   =>  901100602   ;
    329 :   8   =>  901100302   ;
    332 :   15  =>  901100302   ;
    332 :   31  =>  11804903    ;
    332 :   39  =>  901100302   ;
    333 :   7   =>  901100301   ;
    333 :   11  =>  11804903    ;
    333 :   24  =>  901100301   ;
    336 :   11  =>  901100301   ;
    336 :   21  =>  11804903    ;
    336 :   22  =>  901100301   ;
    337 :   4   =>  901100301   ;
    337 :   15  =>  901100301   ;
    338 :   41  =>  901100301   ;
    339 :   19  =>  901100301   ;
    339 :   23  =>  11804903    ;
    339 :   40  =>  901100301   ;
    340 :   3   =>  11804903    ;
    340 :   29  =>  901100302   ;
    340 :   48  =>  901100301   ;
    341 :   17  =>  901100301   ;
    341 :   35  =>  11804903    ;
    341 :   41  =>  901100301   ;
    342 :   26  =>  11403302    ;
    342 :   38  =>  901100301   ;
    343 :   5   =>  11100301    ;
    343 :   16  =>  901100302   ;
    343 :   44  =>  901100301   ;
    346 :   19  =>  901100301   ;
    346 :   23  =>  11804903    ;
    346 :   42  =>  11804903    ;
    347 :   11  =>  11201512    ;
    347 :   21  =>  901100301   ;
    347 :   24  =>  11201512    ;
    347 :   43  =>  901100302   ;
    348 :   4   =>  901100301   ;
    348 :   12  =>  901100301   ;
    349 :   10  =>  11201512    ;
    349 :   15  =>  901100301   ;
    350 :   6   =>  901100301   ;
    350 :   11  =>  901100301   ;
    352 :   4   =>  11804903    ;
    352 :   25  =>  11804903    ;
    352 :   35  =>  901100302   ;
    353 :   19  =>  901100301   ;
    353 :   34  =>  901100302   ;
    354 :   14  =>  901100301   ;
    354 :   35  =>  901100301   ;
    126 :   46  =>  53114903    ;
endrecode;

$ = 510 + P_DAY / 1000;
PROC P_AMOUNT

i = 0;
recode  ID07    :   P_LINE_NUMBER   =>  i;
    140 :   7   =>  8   ;
    143 :   11  =>  43  ;
    148 :   10  =>  15  ;
    232 :   24  =>  28.5    ;
    236 :   28  =>  10.2    ;
    236 :   71  =>  28.5    ;
    241 :   2   =>  10.2    ;
        :       => 0;
endrecode;

if i > 0 then
```

```
    impute($, i);
endif;

PROC P_MULTIPLIER
{
HIES ID Items week 1    Items week 2    Total both weeks    W1/W2 ratio Source  Multiplier
308     1               18              19                  0.1         Week 2  52
318     2               22              24                  0.1         Week 2  52
118     19              6               25                  3.2         Week 1  52
326     42              13              55                  3.2         Week 1  52
250     45              11              56                  4.1         Week 1  52
134     55              3               58                  18.3        Week 1  52
105     31                              31                  0.0         Week 1  52

week1 = P_DAY in 1:7
week2 = P_DAY in 8:14
}

if ID07 in 308,318,118,326,250,134,105 then
    if ID07 in 308 then if P_DAY in 1:7 then P_MULTIPLIER = 0; else P_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 318 then if P_DAY in 1:7 then P_MULTIPLIER = 0; else P_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 118 then if P_DAY in 1:7 then P_MULTIPLIER = 52; else P_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 326 then if P_DAY in 1:7 then P_MULTIPLIER = 52; else P_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 250 then if P_DAY in 1:7 then P_MULTIPLIER = 52; else P_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 134 then if P_DAY in 1:7 then P_MULTIPLIER = 52; else P_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 105 then if P_DAY in 1:7 then P_MULTIPLIER = 52; else P_MULTIPLIER = 0;
    endif; endif;
else
    P_MULTIPLIER = 26;
endif;

P_AN_AMT = P_AMOUNT * P_MULTIPLIER;
P_WGHT_AN_AMT = P_AN_AMT * WEIGHT;


PROC P_LOCATION
$ = 1;
PROC P_EXP_TYPE
if P_PURPOSE in notappl, 4 then
    P_PURPOSE = 1;
elseif P_PURPOSE in 3 then //change non consumption to personal as a beneficiary
    if P_COICOP > 900000000 then
        P_PURPOSE = 1;
    endif;
endif;

if P_PURPOSE in 1 then
    $ = 1;
elseif P_PURPOSE in 2 then
    $ = 3;
elseif P_PURPOSE in 3 then
    $ = 6;
endif;


PROC P_DAY_NAME
numeric tmpDura;
recode  P_COICOP    =>  tmpDura ;
    11100101-23109001   =>  1   ;   //  Food and drinks
    31109101    =>  4   ;   //  Pareu (printed or not)
    31109102    =>  4   ;   //  Clothing fabric, poplin material
    31109201    =>  4   ;   //  Pareu
    31109202    =>  4   ;   //  Material, Fabrics
    31109203    =>  4   ;   //  Poplin
    31209301    =>  4   ;   //  Men And Boys - Lavalava
```

```
31209302    =>  4   ;   //  Men And Boys  - Shirt
31209303    =>  4   ;   //  Men And Boys - Trousers
31209304    =>  4   ;   //  Men And Boys  - T-Shirt
31209305    =>  4   ;   //  Men And Boys - Underwear
31209306    =>  4   ;   //  Men And Boys - School Uniform
31209307    =>  4   ;   //  Men And Boys - Short
31209308    =>  4   ;   //  Men And Boys - Socks
31209309    =>  4   ;   //  Men And Boys - Other Clothes For Man.Boy Or Ned
31209310    =>  4   ;   //  Men And Boys  - Clothes (general - not enough detail)
31209401    =>  4   ;   //  Women And Girls - Shirt
31209402    =>  4   ;   //  Women And Girls - Dress, Skirt
31209403    =>  4   ;   //  Women And Girls - Trousers, Pants
31209404    =>  4   ;   //  Women And Girls - T-Shirt, Top
31209405    =>  4   ;   //  Women And Girls  - Suit
31209406    =>  4   ;   //  Women And Girls - Underwear
31209407    =>  4   ;   //  Women And Girls - School Uniform
31209408    =>  4   ;   //  Women And Girls - Short
31209409    =>  4   ;   //  Women And Girls - Socks
31209410    =>  4   ;   //  Women And Girls - Other Clother For Women/Girl Or Ned
31209411    =>  4   ;   //  Women And Girls - lava-lava
31209412    =>  4   ;   //  Women And Girls - Clothes (general - not enough detail)
31209501    =>  4   ;   //  Garments For Infant 0 To 2 Years Old
31209601    =>  4   ;   //  Unspecifed / unisex clothes in general - shirts / tops
31209602    =>  4   ;   //  Unspecifed / unisex clothes in general - pants, jeans,
trousers
31209603    =>  4   ;   //  Unspecifed / unisex clothes in general - dresses, slacks,
and similar
31209604    =>  4   ;   //  Unspecifed / unisex clothes in general - underwear, socks
31209605    =>  4   ;   //  Unspecifed / unisex clothes in general - pajamas, robes
31209606    =>  4   ;   //  Unspecifed / unisex clothes in general - school uniforms
31209607    =>  4   ;   //  Unspecifed / unisex clothes in general - business, work attire
31209608    =>  4   ;   //  Unspecifed / unisex clothes in general - Not enough detail
31209609    =>  4   ;   //  Unspecifed / unisex clothes in general - Not specified
elsewhere
31209610    =>  4   ;   //  Unspecifed / unisex clothes in general - lava lava
31309601    =>  4   ;   //  Cotton Gloves
31309602    =>  4   ;   //  Tie
31309603    =>  4   ;   //  Belt
31309604    =>  4   ;   //  Cap
31309605    =>  4   ;   //  Hat
31309606    =>  4   ;   //  Other Clothing Accessories
31309701    =>  4   ;   //  Cotton Thread
31309702    =>  4   ;   //  Rayon Material
31309703    =>  4   ;   //  Sewing Needles, Hand
31309704    =>  4   ;   //  Other Sewing Materials
31409801    =>  4   ;   //  Dyeing Of Garments
31409901    =>  4   ;   //  Cleaning Of Garments, Laundry Shop
31410001    =>  4   ;   //  Taylor
31410101    =>  4   ;   //  Hire Of Garments
32110201    =>  4   ;   //  Male - Beach Jandals
32110202    =>  4   ;   //  Male - Sport Shoes
32110203    =>  4   ;   //  Male - Casual Shoes
32110204    =>  4   ;   //  Male - Other Shoes For Men And Boys Or Ned
32110301    =>  4   ;   //  Female - Beach Jandals
32110302    =>  4   ;   //  Female - Sport Shoes
32110303    =>  4   ;   //  Female - Casual Shoes
32110304    =>  4   ;   //  Female - Other Shoes For Woman And Girl Or Ned
32110401    =>  4   ;   //  Shoes For Infant - 0 To 2 Years
32110501    =>  4   ;   //  !!DO NOT USE !!
32110502    =>  4   ;   //  Part Of Shoes (Heels, Soles, Lace…)
32110503    =>  4   ;   //  Other shoes related articles
32210601    =>  2   ;   //  Repair Of Footwear
32210701    =>  2   ;   //  Shoe-Cleaning Service
32210801    =>  2   ;   //  Hire of footwear
41110901    =>  2   ;   //  Rent Paid By Tenants For Main House
41211001    =>  2   ;   //  Rent Paid By Tenants For Land / Land Lease
41211101    =>  2   ;   //  Rent Paid By Tenants For Another House
42111201    =>  2   ;   //  Imputed Rentals Of Owner Occupying Their Main House
42211301    =>  2   ;   //  Imputed Rental For Secondary Residence
42211401    =>  2   ;   //  Imputed Rentals Of Households Paying A Reduced Rental Or
Housed Free
```

```
43111501    =>  4   ;   //  Pipe
43111502    =>  4   ;   //  Buteline
43111503    =>  4   ;   //  Elbow
43111504    =>  4   ;   //  Tap
43111505    =>  4   ;   //  Other Plumbing Materials
43111601    =>  4   ;   //  Ceramic Floor Tile
43111602    =>  4   ;   //  Floor Mosaic (Shower)
43111603    =>  4   ;   //  Other Surfacing Materials (Primaflex…)
43111701    =>  4   ;   //  Masking Tape For Painting
43111702    =>  4   ;   //  Paint Brush
43111703    =>  4   ;   //  Paint
43111704    =>  4   ;   //  Other Painting Materials
43111801    =>  4   ;   //  Blocks,Concrete Blocks
43111802    =>  4   ;   //  Cement
43111803    =>  4   ;   //  Door
43111804    =>  4   ;   //  Window, Louvre
43111805    =>  4   ;   //  Electric Cable inc network cable
43111806    =>  4   ;   //  Electric Materials (Plugs…)
43111807    =>  4   ;   //  Sink
43111808    =>  4   ;   //  Toilet Bowl, Sit, Shower
43111809    =>  4   ;   //  Plywood
43111810    =>  4   ;   //  Iron Roofing
43111811    =>  4   ;   //  Timber inc posts, planks, scotia, weatherboard, skirting
43111812    =>  4   ;   //  Hardboard
43111813    =>  4   ;   //  Inuslation foil, DPC
43111814    =>  4   ;   //  Tank iron
43111815    =>  4   ;   //  Rebar, reinforcing steel
43111816    =>  4   ;   //  Angle bar
43111817    =>  4   ;   //  Gib board
43111818    =>  4   ;   //  Cyclone straps
43111819    =>  4   ;   //  Ridge capping
43111820    =>  4   ;   //  Spouting, gutters
43111821    =>  4   ;   //  Other material for house maintenance ned
43211901    =>  2   ;   //  Contractor, Builder, Plumber, Electrician…
44112001    =>  2   ;   //  Water Bill, Water Meter
44112002    =>  2   ;   //  Water connection fees
44112003    =>  2   ;   //  Water truck
44112004    =>  2   ;   //  Other fee related to water supply
44212101    =>  2   ;   //  Garbage disposal
44312201    =>  2   ;   //  Sewer collection and disposal
44412301    =>  2   ;   //  Compound charge, co proprietor charge for caretaking of the
compound / building (cleaning….)
44412401    =>  2   ;   //  Security Services
45112501    =>  2   ;   //  Electricity Charge, Power Meter, Bill
45112502    =>  2   ;   //  Other Electricity Connection Fees
45212601    =>  2   ;   //  Do not use this commodity thank you - town gas and natural gas
45212701    =>  2   ;   //  Gas Cylinder
45212702    =>  1   ;   //  Gas Refill
45312801    =>  1   ;   //  Kerosene for domestic heating and lighting (fuel for vehicle
see 07.2.2_196.03)
45312802    =>  1   ;   //  Benzene
45312803    =>  1   ;   //  Petrol
45312804    =>  1   ;   //  Other Fuel For Generator Or Lights (Oil…)
45412901    =>  1   ;   //  Coconut Husk
45412902    =>  1   ;   //  Charcoal Imported
45412903    =>  1   ;   //  Other Solid Fuel (Firewood…)
45513001    =>  2   ;   //  Ice used for cooling and refrigeration purposes, ice cube,
ice blocks
51113101    =>  3   ;   //  Couche, Sofa
51113102    =>  3   ;   //  Other Lounge Furniture Or Ned
51113102    =>  3   ;   //  Bookshelf
51113201    =>  3   ;   //  Dining Table
51113202    =>  3   ;   //  Other Table
51113203    =>  3   ;   //  Chairs
51113204    =>  3   ;   //  Dining Hall Cabinet
51113205    =>  3   ;   //  Other Dining Hall Furniture Or Ned
51113301    =>  3   ;   //  Beds With Mattress
51113302    =>  3   ;   //  Beds Without Mattress
51113303    =>  3   ;   //  Mattress
51113304    =>  3   ;   //  Chest Of Drawers
51113305    =>  3   ;   //  Cupboards
```

```
51113306 => 3 ; // Dressing Table
51113307 => 3 ; // Wardrobe
51113308 => 3 ; // Other Bedroom Furniture Or Ned
51113401 => 3 ; // Foodsafe
51113402 => 3 ; // Pantry Cupboard
51113403 => 3 ; // Other Kitchen Furniture
51113501 => 3 ; // Outdoor Chairs
51113502 => 3 ; // Outdoor Table
51113503 => 3 ; // Other Outdoor Furniture Or Ned
51113601 => 3 ; // Cradles
51113602 => 3 ; // High Chairs
51113603 => 3 ; // Playpens
51113604 => 3 ; // Other Baby Furniture Or Ned
51113701 => 3 ; // Bathroom Cabinets
51113702 => 3 ; // Other Bathroom Furniture Or Ned
51113801 => 3 ; // Halogen Lamps
51113802 => 3 ; // Other Lamp Or Ned (Exclude Torch)
51113901 => 3 ; // Computer Table
51113902 => 3 ; // Mirror
51113903 => 3 ; // Bookcase
51113904 => 3 ; // Decoration, Carving, Painting…
51113905 => 3 ; // Other Furniture Or Ned
51214001 => 3 ; // Carpet
51214002 => 3 ; // Mat
51214003 => 3 ; // Other Floor Cover
51314101 => 2 ; // Repair Of Furniture
52114201 => 4 ; // Awnings
52114202 => 4 ; // Curtain, Curtain Material
52114203 => 4 ; // Fabric Blinds
52114301 => 4 ; // Bolsters
52114302 => 4 ; // Futons
52114303 => 4 ; // Hammocks
52114304 => 4 ; // Pillows Including Cushions
52114401 => 4 ; // Bed Sheets
52114402 => 4 ; // Bed Spreads
52114403 => 4 ; // Bedding Sets - Combination Eg Fitted, Flat And Pillow Cases
52114404 => 4 ; // Blankets
52114405 => 4 ; // Eiderdowns
52114406 => 4 ; // Mosquito Nets
52114407 => 4 ; // Pillow Cases
52114408 => 4 ; // Other Bedding Or Ned
52114501 => 4 ; // Table Clothes, Napkin
52114601 => 4 ; // Face Towel
52114602 => 4 ; // Hand Towel
52114603 => 4 ; // Bath Towel
52114604 => 4 ; // Beach Towel
52114605 => 4 ; // Other Towel Or Ned
52114701 => 4 ; // Shopping Plastic Bags
52114702 => 4 ; // Shower Curtain
52114703 => 4 ; // Covers For Clothes And Furniture
52114704 => 4 ; // Doilies
52114705 => 4 ; // Flags
52114706 => 4 ; // Laundry Bags
52114707 => 4 ; // Shoe Bags
52114708 => 4 ; // Tea Towels
52114709 => 4 ; // Other Household Textile
52114801 => 2 ; // Repair Of Such Textiles
53114901 => 3 ; // Stove, Kerosene, 2-Burner
53114902 => 3 ; // Wicks, Kerosene Stove
53114903 => 3 ; // Dishwasher
53114904 => 3 ; // Freezer
53114905 => 3 ; // Fridge-Freezer
53114906 => 3 ; // Microwave
53114907 => 3 ; // Oven Only
53114908 => 3 ; // Refrigerator
53114909 => 3 ; // Stove  – Electric
53114910 => 3 ; // Stove – Gas
53114911 => 3 ; // Stove - Kerosen
53114912 => 3 ; // Wick For Kerosen Stove
53114913 => 3 ; // Stove Ned
53114914 => 3 ; // Other Major Kitchen Appliance Or Ned
```

```
53115001    =>  3   ;   //  Clothes Drier
53115002    =>  3   ;   //  Washing Machine
53115003    =>  3   ;   //  Other major laundry appliances
53115101    =>  3   ;   //  Air-Conditioner
53115102    =>  3   ;   //  Electric Water Heater (For Bathing)
53115103    =>  3   ;   //  Solar Power Water Heater
53115104    =>  3   ;   //  Extractor Hoods
53115105    =>  3   ;   //  Safes
53115106    =>  3   ;   //  Sewing Machine
53115107    =>  3   ;   //  Vacuum Cleaners, Steam Cleaning Machines, Carpet Shampooing
Machines And Machines For Scrubbing, Waxing And Polishing Floors
53115108    =>  3   ;   //  Other Major Household Appliances
53115109    =>  3   ;   //  Generator
53115110    =>  3   ;   //  Solar panel
53115111    =>  3   ;   //  Water tank
53215201    =>  4   ;   //  Jug, Electric, Steel
53215202    =>  4   ;   //  Fan For Ceiling
53215203    =>  4   ;   //  Fan For Table, Gie
53215204    =>  4   ;   //  Fan, Pedestal
53215205    =>  4   ;   //  Electric Toaster
53215206    =>  4   ;   //  Rice, Frc Rice Cooker
53215207    =>  4   ;   //  Blenders
53215208    =>  4   ;   //  Can-Openers
53215209    =>  4   ;   //  Coffee Maker
53215210    =>  4   ;   //  Deep Fryers
53215211    =>  4   ;   //  Food Processor
53215212    =>  4   ;   //  Frypans
53215213    =>  4   ;   //  Irons
53215214    =>  4   ;   //  Toaster
53215215    =>  4   ;   //  Other Small Electric Appliance
53315301    =>  2   ;   //  Repair Of Household Appliance
54015401    =>  4   ;   //  Ceremic Ware
54015402    =>  4   ;   //  China (Bone China, Royal Dalton, Lladro)
54015403    =>  4   ;   //  Crystal
54015404    =>  4   ;   //  Enamel
54015405    =>  4   ;   //  Glassware
54015406    =>  4   ;   //  Plastic Ware
54015407    =>  4   ;   //  Steel
54015408    =>  4   ;   //  Other Glassware Or Glassware Ned
54015501    =>  4   ;   //  Cutlery, Flatware And Silverware
54015601    =>  4   ;   //  Frying Pans
54015602    =>  4   ;   //  Hotplates
54015603    =>  4   ;   //  Household Scales
54015604    =>  4   ;   //  Kettle
54015605    =>  4   ;   //  Pots
54015606    =>  4   ;   //  Pressure Cookers
54015607    =>  4   ;   //  Saucepan
54015608    =>  4   ;   //  Tauaga (coconut strainer)
54015609    =>  4   ;   //  Other Non Electric Kitchen Ustensils Or Ned
54015701    =>  4   ;   //  Gas Lighter
54015702    =>  4   ;   //  Gas Lighter, Bic
54015703    =>  4   ;   //  Bottle Racks
54015704    =>  4   ;   //  Buckets
54015705    =>  4   ;   //  Containers For Bread, Coffee, Spices
54015706    =>  4   ;   //  Feeding Bottles
54015707    =>  4   ;   //  Iceboxes
54015708    =>  4   ;   //  Ironing Boards
54015709    =>  4   ;   //  Laundry Baskets
54015710    =>  4   ;   //  Letter Boxes
54015711    =>  4   ;   //  Portable Money Boxes
54015712    =>  4   ;   //  Thermos Flasks
54015713    =>  4   ;   //  Towel Rails
54015714    =>  4   ;   //  Waste Bins
54015715    =>  4   ;   //  Waste Paper Baskets
54015716    =>  4   ;   //  Scales, Counter 20Kg
54015717    =>  4   ;   //  Hand Basin With Brackets (For Toilets)
54015718    =>  4   ;   //  Jug, Plastic,
54015719    =>  4   ;   //  Plastic Jerry Cans, 10 Litres
54015720    =>  4   ;   //  Tarpaulin,
54015721    =>  4   ;   //  Tarpaulin Clear (Heavy Plastic)
54015722    =>  4   ;   //  Other Non Electric Hh Article Or Ned
```

```
54015801    =>  2   ;   //  Repair Of Household Article
55115901    =>  3   ;   //  Cement Mixer Petrol
55115902    =>  3   ;   //  Water Pump Electric Orange Sj400.
55115903    =>  3   ;   //  Compression
55115904    =>  3   ;   //  Engine Cleaning Gun
55115905    =>  3   ;   //  V Belts ( For Concrete Mixers)
55115906    =>  3   ;   //  Brush Cutters
55115907    =>  3   ;   //  Chainsaws
55115908    =>  3   ;   //  Cultivators
55115909    =>  3   ;   //  Electric Drills
55115910    =>  3   ;   //  Garden Tractors
55115911    =>  3   ;   //  Lawnmowers
55115912    =>  3   ;   //  Motor Mowers
55115913    =>  3   ;   //  Sanders And Hedge Cutters
55115914    =>  3   ;   //  Other Motorized Tools And Equipment
55116001    =>  2   ;   //  Repair Of Such Articles
55216101    =>  3   ;   //  Dustmask
55216102    =>  3   ;   //  Hacksaws
55216103    =>  3   ;   //  Hand Saw
55216104    =>  3   ;   //  Petrol Pump
55216105    =>  3   ;   //  Circular Saw
55216106    =>  3   ;   //  Clamp Tool
55216107    =>  3   ;   //  Air Duster Gun
55216108    =>  3   ;   //  Coach Screws
55216109    =>  3   ;   //  Dyna Bolts
55216110    =>  3   ;   //  Crowbar
55216111    =>  3   ;   //  Cut Off Wheel 230X2.5X22Mm
55216112    =>  3   ;   //  Hacksaw
55216113    =>  3   ;   //  Matic Guns, Standard # Skeleton
55216114    =>  3   ;   //  Multimeter Pocket Digital
55216115    =>  3   ;   //  Sandpaper
55216116    =>  3   ;   //  Spray Gun Suction
55216117    =>  3   ;   //  Trowel
55216118    =>  3   ;   //  Wire Brush
55216119    =>  3   ;   //  Files
55216120    =>  3   ;   //  Hammers
55216121    =>  3   ;   //  Pliers
55216122    =>  3   ;   //  Rasps
55216123    =>  3   ;   //  Saws
55216124    =>  3   ;   //  Screwdrivers
55216125    =>  3   ;   //  Spanners
55216126    =>  3   ;   //  Trimming Knives
55216127    =>  3   ;   //  Wrenches
55216128    =>  3   ;   //  Other Hand Tools
55216201    =>  3   ;   //  Hose 1With Fittings
55216202    =>  3   ;   //  Cane Knives, Bush Knives
55216203    =>  3   ;   //  Forks
55216204    =>  3   ;   //  Hoses And Sprinklers
55216205    =>  3   ;   //  Rakes
55216206    =>  3   ;   //  Scythes
55216207    =>  3   ;   //  Shovels
55216208    =>  3   ;   //  Sickles
55216209    =>  3   ;   //  Spades
55216210    =>  3   ;   //  Watering Cans
55216211    =>  3   ;   //  Wheelbarrows
55216212    =>  3   ;   //  Other Garden Tools
55216301    =>  3   ;   //  Ladders And Steps
55216401    =>  3   ;   //  Entrance Door Lock Set
55216402    =>  3   ;   //  Hasp & Staple
55216403    =>  3   ;   //  Security Door Locks C/W Knob/Handles
55216404    =>  3   ;   //  Security Entrance Door Locks/Knob Handles
55216405    =>  3   ;   //  Toilet Door Lock Set
55216406    =>  3   ;   //  Brass Padlocks
55216407    =>  3   ;   //  Pad Bolt
55216408    =>  3   ;   //  Padlock
55216409    =>  3   ;   //  Handles
55216410    =>  3   ;   //  Hinges
55216411    =>  3   ;   //  Security mesh
55216412    =>  3   ;   //  Other security materials ned
55216501    =>  3   ;   //  Shower Curtain Rods
55216502    =>  3   ;   //  Toilet Paper Holder
```

```
55216503 => 3 ; // Toilet Seat Covers
55216504 => 3 ; // Carpet Rods
55216505 => 3 ; // Curtain Rails
55216506 => 3 ; // Hooks
55216507 => 3 ; // Other Metal Article For The House
55216601 => 3 ; // Chains
55216602 => 3 ; // Grids
55216603 => 3 ; // Hoop Segments For Fencing And Bordering
55216604 => 3 ; // Stakes
55216605 => 3 ; // Tie wire (included chicken wire)
55216606 => 3 ; // Mesh, meshwire
55216607 => 3 ; // Other Metal Article For The Garden
55216701 => 3 ; // Battery (Sk-N100)
55216702 => 3 ; // Battery (Sk-N70Zl)
55216703 => 3 ; // Battery, Alkaline
55216704 => 3 ; // Battery, Energizer
55216705 => 3 ; // Battery, Eveready
55216706 => 3 ; // Battery, Magnacell
55216707 => 3 ; // Battery, Panasonic
55216708 => 3 ; // Battery Ned
55216709 => 3 ; // Energy Saver Lights
55216710 => 3 ; // Extension Cords
55216711 => 3 ; // Fluorescent Light
55216712 => 3 ; // Fluorescent Tube
55216713 => 3 ; // Light Bulb
55216714 => 3 ; // Torches Powered Diving Waterproof
55216715 => 3 ; // Spark Plugs
55216716 => 3 ; // Alarms
55216717 => 3 ; // Bells
55216718 => 3 ; // Flashlights
55216719 => 3 ; // Fluorescent Tubes starters
55216720 => 3 ; // Hand Lamps
55216721 => 3 ; // Power Sockets
55216722 => 3 ; // Switches
55216723 => 3 ; // Torches
55216724 => 3 ; // Wiring Flex
55216725 => 3 ; // Other Small Electric Appliance
55216801 => 2 ; // Repair Of Such Articles
56116901 => 1 ; // Air Freshener
56116902 => 1 ; // Dish Washing Liquid
56116903 => 1 ; // Dish Washing Liquid, Joy
56116904 => 1 ; // Disinfectant, Dettol
56116905 => 1 ; // Disinfectant: Pine-O-Cleen
56116906 => 1 ; // Fly Spray Surface, Mortein
56116907 => 1 ; // Fly Spray, Red, Mortein
56116908 => 1 ; // Insect Repellent, Aeroguard
56116909 => 1 ; // Rat Trap
56116910 => 1 ; // Soap, Laundry, Pacific
56116911 => 1 ; // Soap, Laundry, Sudso
56116912 => 1 ; // Toilet Cleaner, Harpic
56116914 => 1 ; // Washing Powder, Boom
56116915 => 1 ; // Washing Powder, Gain
56116916 => 1 ; // Washing Powder, Lerosa
56116917 => 1 ; // Washing Powder, Surf
56116918 => 1 ; // Washing Powder, Tasi
56116919 => 1 ; // Cleaning Paste Axion
56116920 => 1 ; // Cold Power
56116921 => 1 ; // Fabric Softener, Downy
56116922 => 1 ; // Moth Balls
56116923 => 1 ; // Napisan Powder
56116924 => 1 ; // Bleach, Janola
56116925 => 1 ; // Clorox
56116926 => 1 ; // Clorox, Janola,
56116927 => 1 ; // Vaselin
56116928 => 1 ; // Other Cleaning And Maintenance Products
56117001 => 1 ; // Sponge
56117002 => 1 ; // Sponge Mop With Handle
56117003 => 1 ; // Dustpan And Brush (2 Piece)
56117004 => 1 ; // Mop Complete
56117005 => 1 ; // Mops & Buckets
56117006 => 1 ; // Toilet Cleaning Brushes C/W Stand
```

```
56117008    =>  1   ;   //  Brooms
56117009    =>  1   ;   //  Duster
56117010    =>  1   ;   //  Wired steelo
56117010    =>  1   ;   //  Other Articles For Cleaning
56117101    =>  1   ;   //  Aluminium Foil
56117102    =>  1   ;   //  Foam Cups
56117103    =>  1   ;   //  Foam Plates
56117104    =>  1   ;   //  Napkins, Paper
56117105    =>  1   ;   //  Plastic Cups
56117106    =>  1   ;   //  Plastic Dinner Plates
56117107    =>  1   ;   //  Plastic Spoons
56117108    =>  1   ;   //  Table Cloth, Plastic,
56117109    =>  1   ;   //  Filter
56117110    =>  1   ;   //  Vaccum Cleaner Bags
56117111    =>  1   ;   //  Kitchen Paper, Tissue, Paper Towel
56117112    =>  1   ;   //  Plastic Bin Liner
56117113    =>  1   ;   //  Other Paper Product
56117201    =>  1   ;   //  Candle
56117202    =>  1   ;   //  Adhesive Tapes
56117203    =>  1   ;   //  Camphor
56117204    =>  1   ;   //  String rope
56117205    =>  1   ;   //  Clothes□Hangers
56117206    =>  1   ;   //  Fire Extinguishers For Households
56117207    =>  1   ;   //  Glues
56117208    =>  1   ;   //  Incense Sticks
56117209    =>  1   ;   //  Knitting Needles
56117210    =>  1   ;   //  Lamp Wicks
56117211    =>  1   ;   //  Anchor Bolts M12 X 300
56117212    =>  1   ;   //  Rivet
56117213    =>  1   ;   //  Nail plate
56117214    =>  1   ;   //  Screws
56117215    =>  1   ;   //  Nail
56117216    =>  1   ;   //  Rubber gloves
56117222    =>  1   ;   //  PVC glue
56117224    =>  1   ;   //  Hinges
56117225    =>  1   ;   //  Matches
56117226    =>  1   ;   //  Measuring Tape
56117227    =>  1   ;   //  Mosquito Coil
56117228    =>  1   ;   //  Mosquito Coil, Zap
56117237    =>  1   ;   //  Packaging Tape
56117241    =>  1   ;   //  Cloth Pegs Hard Plastic
56117244    =>  1   ;   //  Other Non Durable Articles
56217301    =>  2   ;   //  Butlers
56217302    =>  2   ;   //  Cooks
56217303    =>  2   ;   //  Drivers
56217304    =>  2   ;   //  Gardeners
56217305    =>  2   ;   //  Governesses
56217306    =>  2   ;   //  Maids
56217307    =>  2   ;   //  Secretaries
56217308    =>  2   ;   //  Tutors
56217309    =>  2   ;   //  Baby sitting / child minding services provided at home
56217310    =>  2   ;   //  Elderly care services provided at home
56217398    =>  2   ;   //  Other Domestic Services - not elsewhere classified (nec)
56217399    =>  2   ;   //  Other Domestic Services - not enough detail (ned)
56217401    =>  2   ;   //  Domestic services, including baby-sitting and housework,
supplied by enterprises or self-employed persons
56217501    =>  2   ;   //  Pest control, fumigation
56217502    =>  2   ;   //  Lawn mowing services
56217503    =>  2   ;   //  Other household services provided by external company to the
household
56217601    =>  2   ;   //  Dry-cleaning, laundering and dyeing of household linen,
household textiles carpets
56217701    =>  2   ;   //  Hire of furniture, equipment (e.g. hire of cement mixer…)
61117801    =>  1   ;   //  Prescription medicine
61117901    =>  1   ;   //  Tablets, Panadol
61117902    =>  1   ;   //  Vicks Vaporub
61117903    =>  1   ;   //  Micreme
61117904    =>  1   ;   //  Panadol
61117905    =>  1   ;   //  Other Patent Medecine
61118001    =>  1   ;   //  Cod Liver Oil
61118002    =>  1   ;   //  Halibut Liver Oil
```

```
61118003    =>  1   ;   //  Oral Contraceptives
61118004    =>  1   ;   //  Serums And Vaccines
61118005    =>  1   ;   //  Vitamins And Minerals
61118006    =>  1   ;   //  Other Pharmaceutical Products Or Ned
61218101    =>  1   ;   //  Band Aids, Jj
61218102    =>  1   ;   //  Bandages
61218103    =>  1   ;   //  Clinical Thermometers
61218104    =>  1   ;   //  Condoms
61218105    =>  1   ;   //  First□Aid Kits
61218106    =>  1   ;   //  Hot Water Bottles
61218107    =>  1   ;   //  Hypodermic Syringes
61218108    =>  1   ;   //  Ice Bags
61218109    =>  1   ;   //  Mechanical Contraceptive Devises
61218110    =>  1   ;   //  Medical Hosiery Items
61218111    =>  1   ;   //  Pregnancy Tests
61218112    =>  1   ;   //  Other Medical Product
61318201    =>  3   ;   //  Artificial Limbs And Other Prosthetic Devices
61318202    =>  3   ;   //  Corrective Eye Glasses And Contact Lenses
61318203    =>  3   ;   //  Crutches
61318204    =>  3   ;   //  Dentures Excluding Fitting Costs
61318205    =>  3   ;   //  Devices For Monitoring Blood Pressure
61318206    =>  3   ;   //  Glass Eyes
61318207    =>  3   ;   //  Health Lamps
61318208    =>  3   ;   //  Hearing Aids
61318209    =>  3   ;   //  Medical Massage Equipment
61318210    =>  3   ;   //  Neck Braces
61318211    =>  3   ;   //  Orthopaedic Braces And Supports
61318212    =>  3   ;   //  Orthopaedic Footwear
61318213    =>  3   ;   //  'Special Beds'
61318214    =>  3   ;   //  Surgical Belts, Trusses And Supports
61318215    =>  3   ;   //  Wheelchairs – Powered And Unpowered And Invalid Carriers
61318216    =>  3   ;   //  Other Therapeutic Appliances
61318301    =>  2   ;   //  Repair or hire of therapeutic appliances
62118401    =>  2   ;   //  Consultations Of Doctor, Specialist
62218501    =>  2   ;   //  Consultations Of Dentist
62318601    =>  2   ;   //  Blood Test, Xray, Analysis
62318602    =>  2   ;   //  Nurse And Midwife
62318603    =>  2   ;   //  Other Paramedical Doctor
62318604    =>  2   ;   //  Other paramedical services (chiropractor…)
62318701    =>  2   ;   //  Traditional Healers
63018801    =>  2   ;   //  Hosptial, Clinic Fee
63018901    =>  2   ;   //  Medical service eg services of physicians
63019001    =>  2   ;   //  Paramedical services eg services of chiropractors,
optometrists
71119101    =>  3   ;   //  Car
71119102    =>  3   ;   //  Purchase of pick up truck, utility vehicle
71119103    =>  3   ;   //  Purchase of truck, mini bus, van
71219201    =>  3   ;   //  Motor Cycle
71219202    =>  3   ;   //  Scooter
71319301    =>  3   ;   //  Bicycle
71419401    =>  3   ;   //  Animal drawn vehicle
72119501    =>  4   ;   //  Aluminium Round Tubes For Boat
72119502    =>  4   ;   //  Fan Belt
72119503    =>  4   ;   //  Tyre
72119504    =>  4   ;   //  Tyre Tubes
72119505    =>  4   ;   //  Batteries For Vehicle
72119506    =>  4   ;   //  Tyre inflator
72119507    =>  4   ;   //  Other Spare Part For Vehicle And Boat
72219601    =>  1   ;   //  Fuel / gas for cars and land vehicle (private use)
72219602    =>  1   ;   //  Fuel / gas for boats and other water crafts (private use)
72219603    =>  1   ;   //  Fuel / gas for other (chainsaw, tillers, tractor…)
72219604    =>  1   ;   //  Fuel / gas without any details
72219605    =>  1   ;   //  Other fuel / gas for vehicle n e s
72219606    =>  1   ;   //
72219607    =>  1   ;   //
72219608    =>  1   ;   //
72219701    =>  1   ;   //  Oil Outboard
72219702    =>  1   ;   //  Oil Outboard Super
72219703    =>  1   ;   //  Oil Outboard Super, Ob
72219704    =>  1   ;   //  Oil Sae15/40
72219705    =>  1   ;   //  Oil Sae30
```

```
72219706    =>  1   ;   //  Oil, Air Compresser
72219707    =>  1   ;   //  Oil, Hydraulic
72219708    =>  1   ;   //  Marine Spray 4 Crc/Wd40
72219709    =>  1   ;   //  Gear Oil
72219710    =>  1   ;   //  Grease Cartridges, Multipurpose
72219711    =>  1   ;   //  Oil Lubricant Ned
72319801    =>  2   ;   //  Service For Maintenance Of Car/Boat, Mechanic
72419901    =>  2   ;   //  Driving lesson
72419902    =>  2   ;   //  Driving license fee
72419903    =>  2   ;   //  Registration fee
72419904    =>  2   ;   //  Other fee in respect of vehicle
72420001    =>  2   ;   //  Car Rental
72420002    =>  2   ;   //  Truck rental, hire
72420003    =>  2   ;   //  Mini bus, van rental, hire
72420004    =>  2   ;   //  Motorcycle / scooter rental, hire
72420005    =>  2   ;   //  Other vehicle rental n e s
73220101    =>  2   ;   //  Bus Fare
73220102    =>  2   ;   //  Taxi Fare
73220103    =>  2   ;   //  Other Transport Fare Ned
73320201    =>  2   ;   //  Air Fare
73320202    =>  2   ;   //  Air fare - domestic
73320203    =>  2   ;   //  Air fare without any detail
73420301    =>  2   ;   //  Boat Fare
73420302    =>  2   ;   //  Boat fare (travel overseas)
73420303    =>  2   ;   //  Other fees related to transport by boat n e s
73520401    =>  2   ;   //  Air fare and boat fare together
73620501    =>  2   ;   //  Freight, Delivery Charge
73620502    =>  2   ;   //  Other services related to transport
81120601    =>  2   ;   //  Payments For Delivery Of Letter, Postcards And Parcels
81120602    =>  2   ;   //  Private Mail And Parcel Delivery
81120603    =>  2   ;   //  Purchases Of Postage Stamps
81120604    =>  2   ;   //  Purchase Of Pre□Franked Postcards And Aerogramme
81120605    =>  2   ;   //  Other Postal Fee
82020701    =>  3   ;   //  Purcahse Of Cell (Mobile) Phones, Smartphone
82020702    =>  3   ;   //  Purchase Of Land □Line Telephones
82020703    =>  3   ;   //  Radio Telephones
82020704    =>  3   ;   //  Telefax Machines
82020705    =>  3   ;   //  Other Telephone And Telefax Equipment
82020801    =>  2   ;   //  Repair Of Such Equipment
83020901    =>  2   ;   //  Connection And Subscription Costs Of Cell (Mobile) Phones
83020902    =>  2   ;   //  Purchase Of Recharge Cards For Cell (Mobile) Phones
83020903    =>  2   ;   //  Purchase Of Sim Card
83020904    =>  2   ;   //  Installation And Subscription Costs Of Land□Line Telephone
Equipment
83020905    =>  2   ;   //  Internet Services
83020906    =>  2   ;   //  Other Telephone Services
91121001    =>  3   ;   //  Car Radios
91121002    =>  3   ;   //  Cd-Players
91121003    =>  3   ;   //  Microphones And Earphones
91121004    =>  3   ;   //  Radio Clocks
91121005    =>  3   ;   //  Radio Sets
91121006    =>  3   ;   //  Stereos And Their Constituent Units Eg Tuners, Amplifiers Etc
91121007    =>  3   ;   //  Television Aerials Of All Types
91121008    =>  3   ;   //  Television Sets
91121009    =>  3   ;   //  Two-Way Radios
91121010    =>  3   ;   //  Video Player/Vcd/Dvd
91121011    =>  3   ;   //  Mp3 Players, Ipods
91121012    =>  3   ;   //  Other Equipment
91221101    =>  3   ;   //  Cameras
91221102    =>  3   ;   //  Video camera
91221201    =>  3   ;   //  Film And Slide Projectors
91221301    =>  3   ;   //  Accessories for camera (flash, objective, lenses…)
91221401    =>  3   ;   //  Binoculars
91221402    =>  3   ;   //  Compass
91221403    =>  3   ;   //  Other Optical Instrument
91221404    =>  3   ;   //  Telescopes
91221405    =>  3   ;   //  Gps devices
91321501    =>  3   ;   //  Applications
91321502    =>  3   ;   //  Calculators
91321503    =>  3   ;   //  Computer Software Packages, E.G. Operating Systems
91321504    =>  3   ;   //  Personal Computers Desktop
```

```
91321505    =>  3   ;   //  Personal Computers Laptop
91321506    =>  3   ;   //  Printers And Miscellaneous Accessories Accompanying Them
91321507    =>  3   ;   //  Tablet
91321508    =>  3   ;   //  Other It Equipment
91421601    =>  4   ;   //  Dvd, Blue Ray
91421602    =>  4   ;   //  Cd Compact Disc
91421603    =>  4   ;   //  Other Recording Media
91421604    =>  4   ;   //  portable harddrives, flash drives
91521701    =>  2   ;   //  Repair Of Audio-Visual, Photographic And Information
Processing Equipment
92121801    =>  3   ;   //  Boat without outboard motor
92121802    =>  3   ;   //  Canoe
92121803    =>  3   ;   //  Outboard Motor
92121805    =>  3   ;   //  Dive gear (tanks, bc, regulators, etc.)
92121807    =>  3   ;   //  Water craft trailer
92121808    =>  3   ;   //  Other major durables for outdoor recreation
92121809    =>  3   ;   //  Boat with outboard motor
92221901    =>  3   ;   //  Guitare
92221902    =>  3   ;   //  Guitar Strings, Steel
92221903    =>  3   ;   //  Other Musical Instrument
92221904    =>  3   ;   //  Other musical instrument n e s
92222001    =>  3   ;   //  Billiard tables
92222002    =>  3   ;   //  Gaming machines
92222003    =>  3   ;   //  Pinball machines
92222004    =>  3   ;   //  Pin-pong table (table-tennis table)
92322101    =>  2   ;   //  Maintenance And Repair Of Other Major Durables For
Recreation And Culture
93122201    =>  4   ;   //  Card Games
93122202    =>  4   ;   //  Chess
93122203    =>  4   ;   //  Parlour Games
93122204    =>  4   ;   //  Video Game Console And Accessories
93122205    =>  4   ;   //  Other Games
93122301    =>  4   ;   //  Barbie Dolls
93122302    =>  4   ;   //  Cars (Police) Remote Control
93122303    =>  4   ;   //  Toys of all kinds (dolls, fireworks, toy cars, etc.)
93122304    =>  4   ;   //  Water Guns
93122305    =>  4   ;   //  X-Mas Lights
93122306    =>  4   ;   //  X-Mas Tree Decoration
93122307    =>  4   ;   //  Baby Dolls With Accessories
93122308    =>  4   ;   //  Other Toys
93122401    =>  4   ;   //  Other items for collection e.g. coins, medals, zoological
and botanical specimens etc.)
93122402    =>  4   ;   //  Stamp collecting requisites (used or cancelled postage
stamps, stamp albums etc.)
93222501    =>  4   ;   //  Beach Volley Ball
93222502    =>  4   ;   //  Cricket Ball
93222503    =>  4   ;   //  Other Sport Equipment
93222504    =>  4   ;   //  Golf clubs
93222505    =>  4   ;   //  Nets
93222506    =>  4   ;   //  Rackets
93222507    =>  4   ;   //  Shuttlecocks
93222508    =>  4   ;   //  Weights
93222509    =>  4   ;   //  Other equipment for sport n e s
93222601    =>  4   ;   //  Parachute and other sky-diving equipment
93222701    =>  4   ;   //  Firearms (gun, bullet…)
93222801    =>  4   ;   //  Fishing Lines
93222802    =>  4   ;   //  Speargun
93222803    =>  4   ;   //  Mask, Flippers
93222804    =>  4   ;   //  Other Fishing Equipments
93222901    =>  4   ;   //  Frisbee
93222902    =>  4   ;   //  Inflatable Boats
93222903    =>  4   ;   //  Inflatable Swimming Pools
93222904    =>  4   ;   //  Volleyball
93222905    =>  4   ;   //  Other Beach And Open Air Games
93223001    =>  4   ;   //  Air Mattresses And Inflating Pumps
93223002    =>  4   ;   //  Camping Stoves And Barbecues
93223003    =>  4   ;   //  Sleeping Bags
93223004    =>  4   ;   //  Tents And Accessories
93223101    =>  4   ;   //  Football Boots
93223102    =>  4   ;   //  Golfing Shoes
93223103    =>  4   ;   //  Other Such Footwear Fitted With Ice□Skates, Rollers, Spikes,
```

```
Studs, Etc.
93223104    =>  4   ;   //  Rugby Boots
93223201    =>  4   ;   //  Belts
93223202    =>  4   ;   //  Body Padding
93223203    =>  4   ;   //  Boxing Gloves
93223204    =>  4   ;   //  Goggles
93223205    =>  4   ;   //  Headgear
93223206    =>  4   ;   //  Life Jackets
93223207    =>  4   ;   //  Shin Guards
93223208    =>  4   ;   //  Supports
93223301    =>  2   ;   //  Repair Of Such Articles
93323401    =>  1   ;   //  Fertilisers, Composts And Garden Peat
93323402    =>  1   ;   //  Horticultural Preparations
93323403    =>  1   ;   //  Other Garden Plants And Flower Stuff
93323404    =>  1   ;   //  Pots and plant holders
93323405    =>  1   ;   //  Other plants related items n e s
93423501    =>  1   ;   //  Purchase of pets (cat, dog,….)
93423601    =>  1   ;   //  Animal Feed
93423602    =>  1   ;   //  Pig feed, pig pellet, grower
93423603    =>  1   ;   //  Chicken feed
93423604    =>  1   ;   //  Other animal feed (including expired food purchased for
animal feed)
93423605    =>  1   ;   //  Other pet-related products (feed, leashes, vet fee, etc.)
93523701    =>  2   ;   //  Veterinary
93523801    =>  2   ;   //  Other Services Eg Training
94123901    =>  2   ;   //  Bowling Alleys
94123902    =>  2   ;   //  Fairgrounds And Amusement Parks
94123903    =>  2   ;   //  Fitness Centres
94123904    =>  2   ;   //  Golf Courses
94123905    =>  2   ;   //  Gymnasia
94123906    =>  2   ;   //  Horse Racing Courses
94123907    =>  2   ;   //  Motor Racing Circuits
94123908    =>  2   ;   //  Pin□Ball Machines (Excluding Game Of Chance)
94123909    =>  2   ;   //  Roundabouts, See□Saws And Other Playground
94123910    =>  2   ;   //  Facilities For Children
94123911    =>  2   ;   //  See□Saws And Playground Facilities For Children
94123912    =>  2   ;   //  Skating Rinks
94123913    =>  2   ;   //  Sports Stadium
94123914    =>  2   ;   //  Squash Courts
94123915    =>  2   ;   //  Swimming Pools
94123916    =>  2   ;   //  Tennis Courts
94123917    =>  2   ;   //  Velodromes
94124001    =>  2   ;   //  Hire of boat
94124002    =>  2   ;   //  Hire of other accessories for sport and recreation
94124101    =>  2   ;   //  Aerobics
94124102    =>  2   ;   //  Chess
94124103    =>  2   ;   //  Dancing
94124104    =>  2   ;   //  Music
94124105    =>  2   ;   //  Swimming
94124106    =>  2   ;   //  Other Group Lessons
94124201    =>  2   ;   //  Services Of Mountain Guides, Tour Guides Etc
94124301    =>  2   ;   //  Navigational Aid Services For Boating
94124401    =>  2   ;   //  Hire of game-specific footwear (excludes purchase (932))
94224501    =>  2   ;   //  Art Galleries And Exhibitions
94224502    =>  2   ;   //  Cinemas
94224503    =>  2   ;   //  Concert And Music Halls
94224504    =>  2   ;   //  Historical Monuments, National Parks, Zoological And
Botanical Gardens
94224505    =>  2   ;   //  And Aquaria
94224506    =>  2   ;   //  Libraries
94224507    =>  2   ;   //  Theatre
94224508    =>  2   ;   //  Cinema
94224509    =>  2   ;   //  Game At The Stadium
94224510    =>  2   ;   //  Other Services
94224601    =>  2   ;   //  Dvd rental, hire
94224701    =>  2   ;   //  Cable Tv Fee
94224702    =>  2   ;   //  Other Fee Related To Cable Tv
94224801    =>  2   ;   //  Services Of Photographers Eg Film Developing , Print
Processing, Enlarging, Portrait Photography, Wedding Photography
94324901    =>  2   ;   //  Bingo Card
94324902    =>  2   ;   //  Gambling Machine
```

```
94324903    =>  2   ;   //  Other Gamble Fee
95125001    =>  4   ;   //  Atlases
95125002    =>  4   ;   //  Dictionaries
95125003    =>  4   ;   //  Encyclopaedias
95125004    =>  4   ;   //  Guidebooks
95125005    =>  4   ;   //  Musical Scores
95125006    =>  4   ;   //  Novels, Story Books
95125007    =>  4   ;   //  Scrapbooks And Albums For Children
95125008    =>  4   ;   //  Textbooks, School Books
95125009    =>  4   ;   //  Other Book Related Expense
95125101    =>  2   ;   //  Bookbinding
95225201    =>  1   ;   //  Magazines
95225202    =>  1   ;   //  Newspapers
95225203    =>  1   ;   //  Periodicals
95325301    =>  1   ;   //  Greeting Cards, Visiting Card
95325302    =>  1   ;   //  Maps And Globes
95325303    =>  1   ;   //  Poster, Post Card…
95425401    =>  1   ;   //  Envelopes
95425402    =>  1   ;   //  Marking Pen
95425403    =>  1   ;   //  Pencil Case
95425404    =>  1   ;   //  Pencil Sharpener
95425405    =>  1   ;   //  Pens, Blue/Black, Ball Point
95425406    =>  1   ;   //  Pens, Blue/Blk/Red, Metal Point, Bic
95425407    =>  1   ;   //  Rubber (Eraser)
95425408    =>  1   ;   //  Carpenter’S Pencil
95425409    =>  1   ;   //  Carpenter'S Folding Ruler
95425410    =>  1   ;   //  Chalk Line Refill Blue Johnson
95425411    =>  1   ;   //  Crepe Paper
95425412    =>  1   ;   //  Dual Metric/Imperial 8/10M
95425413    =>  1   ;   //  Gift Wrapper
95425414    =>  1   ;   //  Sellotape, 2" Clear
95425415    =>  1   ;   //  Tialeniu School Badges-Jan. 2011
95425416    =>  1   ;   //  Other Stationery Or Ned
95425501    =>  1   ;   //  Ink Cartridge Brother
95425601    =>  1   ;   //  Exercise book
96025701    =>  2   ;   //  All Inclusive Holidays Or Tours Which Provide For Travel,
Food, Accommodation, Guides Etc
101025801   =>  2   ;   //  Kindergarden Fees
101025802   =>  2   ;   //  Primary School Fees
102025901   =>  2   ;   //  Secondary School Fees
103026001   =>  2   ;   //  Technical Education School Fees
103026002   =>  2   ;   //  Vocational Education School Fees
104026101   =>  2   ;   //  University Fees
105026201   =>  2   ;   //  Other Educational Program Fees
105026202   =>  2   ;   //  Tutoring for academic subjects (science….), private lesson
for school purposes
105026203   =>  2   ;   //  Other expenditure related to education (teacher day, sport
day, school events…)
111126301   =>  2   ;   //  Restaurant, Café
111126302   =>  2   ;   //  Bar Night Clubs
111126401   =>  2   ;   //  Ham, Sandwich, Local
111126402   =>  2   ;   //  Ham, Sandwich, Us
111126403   =>  2   ;   //  Other Take Away Food
111126501   =>  2   ;   //  Take away ice cream
111126502   =>  2   ;   //  Take away pizza
111126503   =>  2   ;   //  Cooked poultry, chicken take away
111126504   =>  2   ;   //  Fish and chips take away
111126505   =>  2   ;   //  Burger, cheese burger take away
111126506   =>  2   ;   //  Fried chicken take away
111126507   =>  2   ;   //  Meat pies take away
111126508   =>  2   ;   //  Sandwich take away
111126598   =>  2   ;   //  Other take away plate, take away food (chinese food,
chicken, barbecue, cassava, fish, sashimi ….)
111126599   =>  2   ;   //  Take away food without any detail
111126601   =>  2   ;   //  The sale of cooked dishes by catering contractors whether
collected by the customer or delivered to the customer’s home
111226701   =>  2   ;   //  School restaurant
112026801   =>  2   ;   //  Hotel, Motel, Guest House
112026802   =>  2   ;   //  Packaged accomidation with food (hotel)
112026901   =>  2   ;   //  Boarding Schools, Univ Compound
112027001   =>  2   ;   //  Accommodation services provided by others
```

```
121127101   =>  2   ;   //  Hairdresser
121127201   =>  2   ;   //  Body Care, Depilation (Waxing, Threading, Etc.) And The Like
121127202   =>  2   ;   //  Facials
121127203   =>  2   ;   //  Manicure
121127204   =>  2   ;   //  Non□Medical Massage
121127205   =>  2   ;   //  Pedicure
121127206   =>  2   ;   //  Turkish Baths, Saunas
121127207   =>  2   ;   //  Other Personal Grooming
121127208   =>  2   ;   //  Tattoo, piercings and other body art
121227301   =>  4   ;   //  Curling Tongs And Styling Combs
121227302   =>  4   ;   //  Electric Razors And Hair Trimmers
121227303   =>  4   ;   //  Electric Toothbrushes
121227304   =>  4   ;   //  Hand□Held And Hood Hair Dryers
121227305   =>  4   ;   //  Sunlamps
121227306   =>  4   ;   //  Other Electric Appliance For Personal Care
121227401   =>  2   ;   //  Repair Of Such Appliances
121327501   =>  1   ;   //  Razor Blades
121327502   =>  1   ;   //  Razor Refill, Gillette
121327503   =>  1   ;   //  Razor, Contour
121327504   =>  1   ;   //  Tooth Brush
121327505   =>  1   ;   //  Hair Brush
121327506   =>  1   ;   //  Nail Brush
121327507   =>  1   ;   //  Nail Files
121327508   =>  1   ;   //  Personal Scale, Baby Scale
121327509   =>  1   ;   //  Combs
121327510   =>  1   ;   //  Other Non Electric Appliance For Personal Care
121327601   =>  1   ;   //  Shampoo, Baby Shampoo
121327602   =>  1   ;   //  Hair Conditioner, Palmolive
121327603   =>  1   ;   //  Hair Gel, (Mousse)
121327604   =>  1   ;   //  Hair Shampoo, Palmolive
121327605   =>  1   ;   //  Shaving Cream, Nivea
121327606   =>  1   ;   //  Soap, Lux
121327607   =>  1   ;   //  Soap, Palmolive
121327608   =>  1   ;   //  Soap, Protex
121327609   =>  1   ;   //  Body Wash
121327610   =>  1   ;   //  Body Wash, Lux
121327611   =>  1   ;   //  Body Wash, Protex
121327612   =>  1   ;   //  Body Wash, Suave
121327613   =>  1   ;   //  Toothpaste, Colgate
121327614   =>  1   ;   //  Cleaning Milk
121327615   =>  1   ;   //  Soap Ned
121327616   =>  1   ;   //  Other Articles Of Personal Hygiene
121327701   =>  1   ;   //  Baby Oil, 200Mls, Johnson
121327702   =>  1   ;   //  Baby Powder
121327703   =>  1   ;   //  Baby Powder, Johnsons
121327704   =>  1   ;   //  Body Lotion, Nivea
121327705   =>  1   ;   //  Deodorant Lynx Spray, Men
121327706   =>  1   ;   //  Deodorant Lynx Spray, Women
121327707   =>  1   ;   //  Deodorant Mum Roll-On, Dove (For Ladies)
121327708   =>  1   ;   //  Deodorant Mum Roll-On, Rexona, (For Ladies)
121327709   =>  1   ;   //  Deodorant Mum Roll-On, Rexona,(For Men)
121327710   =>  1   ;   //  Deodorant Spray, (Men), Rexona
121327711   =>  1   ;   //  Deodorant Spray, (Women), Rexona
121327712   =>  1   ;   //  Deodorant Spray, For Men, Nivea
121327713   =>  1   ;   //  Deodorant Spray, For Women, Nivea
121327714   =>  1   ;   //  Deodorant, Impulse
121327715   =>  1   ;   //  Deodorant, Lynx Roll-On, Men
121327716   =>  1   ;   //  Perfume
121327717   =>  1   ;   //  Make Up, Lipstick
121327718   =>  1   ;   //  Other Beauty Product
121327801   =>  1   ;   //  Baby Wipes, Huggies
121327802   =>  1   ;   //  Cotton Buds, Ear Buds
121327803   =>  1   ;   //  Ladies Pads, Johnsons
121327804   =>  1   ;   //  Ladies Pads, Libra
121327805   =>  1   ;   //  Ladies Pads, Modess
121327806   =>  1   ;   //  Ladies Pads, Stay Free
121327807   =>  1   ;   //  Nappies, Huggies
121327808   =>  1   ;   //  Nappies, Soft Love
121327809   =>  1   ;   //  Toilet Paper, Sofan
121327810   =>  1   ;   //  Toilet Paper, Swan
121327811   =>  1   ;   //  Toilet Paper, Wondersoft
```

```
    121327812   =>  1   ;   //  Cotton
    121327813   =>  1   ;   //  Nappies for adults, Depends
    121327814   =>  1   ;   //  Other Product
    122027901   =>  2   ;   //  Services provided by prostitutes and the like
    123128001   =>  3   ;   //  Diamon, gold, platnum, and similar precious metals and stones
    123128101   =>  3   ;   //  Jewellery (local or not): earing, necklace, bracelet
    123128102   =>  3   ;   //  Artificial flowers, artificial tiare
    123128103   =>  3   ;   //  Jewellery without any detail
    123128104   =>  3   ;   //  Fresh flower arrangments in bouquet, lei, ei, head ea, ea
    katu, poe tiare
    123128201   =>  3   ;   //  Clocks
    123128202   =>  3   ;   //  Watch
    123128203   =>  3   ;   //  Alarm Clock
    123128204   =>  3   ;   //  Other Clock And Watch Related Article
    123128301   =>  2   ;   //  Repair Of Such Articles
    123228401   =>  4   ;   //  Handbag
    123228402   =>  4   ;   //  Suit Case
    123228403   =>  4   ;   //  School Backpacks, (Boys)
    123228404   =>  4   ;   //  School Backpacks, (Girls)
    123228405   =>  4   ;   //  Other Travel Goods
    123228501   =>  4   ;   //  Baby Carriages, Carrycots
    123228502   =>  4   ;   //  Car Beds And Seats
    123228503   =>  4   ;   //  Pushchairs
    123228504   =>  4   ;   //  Reins And Harnesses
    123228505   =>  4   ;   //  Other Articles For Babies
    123228601   =>  4   ;   //  Lighter
    123228602   =>  4   ;   //  Pipe
    123228603   =>  4   ;   //  Ashtray
    123228604   =>  4   ;   //  Other Article For Smokers
    123228701   =>  4   ;   //  Umbrella
    123228702   =>  4   ;   //  Umbrella (Beach)
    123228703   =>  4   ;   //  Fan
    123228704   =>  4   ;   //  Sunglasses
    123228705   =>  4   ;   //  Fey Ring
    123228706   =>  4   ;   //  Other Miscellaneous Personal Article
    123228707   =>  4   ;   //  traditional crafts - fan
    123228708   =>  4   ;   //  traditional crafts - handbags, bags, pouches
    123228709   =>  4   ;   //  traditional crafts - hats
    123228710   =>  4   ;   //  traditional crafts - mats
    123228801   =>  4   ;   //  Funerary Articles Eg Coffins, Gravestones, Urns
    123228901   =>  2   ;   //  Repair Of Such Articles
    124029001   =>  2   ;   //  Retirement Homes For Elderly Persons And Residences For
    Disabled Persons
    124029101   =>  2   ;   //  Home help, day care and rehabilitation
    125129201   =>  2   ;   //  Life Assurance
    125129202   =>  2   ;   //  Education Assurance
    125229301   =>  2   ;   //  Home Insurance
    125329401   =>  2   ;   //  Health Insurance
    125429501   =>  2   ;   //  Car Insurance
    125429502   =>  2   ;   //  Boat Insurance
    125429503   =>  2   ;   //  Other Insurance Related To Transport
    125529601   =>  2   ;   //  Personal Insurance
    126129701   =>  2   ;   //  Financial intermediation services indirectly measured
    126229801   =>  2   ;   //  Banking Fees
    126229802   =>  2   ;   //  Money Transfer Fees
    126229803   =>  2   ;   //  Credit Card Fee
    126229804   =>  2   ;   //  Other Financial Fees
    127029901   =>  2   ;   //  Passport Fees
    127029902   =>  2   ;   //  Administrative Document, Certificate (Birth, Death…)
    127029903   =>  2   ;   //  Photocopies
    127029904   =>  2   ;   //  Fees related to the purchase of a house (appraisal, realty,
    etc.)
    127029905   =>  2   ;   //  Legal expenses
    127029907   =>  2   ;   //  Other fees related to personal services
    901100101-999999999 =>  5   ;   //  Current cash transfers: birthdays
endrecode;


numeric tmp2 = notappl;
numeric tmp = notappl;
```

```
if P_PURPOSE in 2 then
    if P_COICOP in 1:130000000 and P_MULTIPLIER > 1 then
        tmp = noccurs(INCOME_EDT)+1;
        insert(INCOME_EDT(tmp));

        I_LINE_NUMBER(tmp) = tmp;
        I_HIES_CODE(tmp) = P_HIES_CODE;
        I_COICOP(tmp) = P_COICOP;
        I_GROSS(tmp) = P_AMOUNT;
        I_AN_GROSS(tmp) = P_AN_AMT;
        I_WGHT_AN_GROSS(tmp) = P_WGHT_AN_AMT;
        I_MULTIPLIER(tmp) = P_MULTIPLIER;
        I_LOCATION(tmp) = P_LOCATION;
        I_INCOME_TYPE(tmp) = P_EXP_TYPE;
        I_PROD_TYPE(tmp) = P_PROD_TYPE;
        I_NUMBER(tmp) = P_QUANTITY;
        I_UNIT(tmp) = P_UNIT;
        I_CATEGORY(tmp) = notappl;

        recode P_PURPOSE => tmp2;
            1 => 51;
            2 => 52;
            3 => 58;
        endrecode;

        I_BENEFICIARY(tmp) = tmp2;
        I_DURABILITY(tmp) = tmpDura;
        I_DESCRIPTION(tmp) = "[diary purchase]"+tolower(P_DESCRIPTION);
    endif;
endif;

if P_COICOP in 1:130000000  and P_MULTIPLIER > 1 then
    tmp = noccurs(EXPENDITURE_EDT)+1;
    insert(EXPENDITURE_EDT(tmp));

    E_LINE_NUMBER(tmp) = tmp;
    E_HIES_CODE(tmp) = P_HIES_CODE;
    E_COICOP(tmp) = P_COICOP;
    E_AMOUNT(tmp) = P_AMOUNT;
    E_AN_AMT(tmp) = P_AN_AMT;
    E_WGHT_AN_AMT(tmp) = P_WGHT_AN_AMT;
    E_MULTIPLIER(tmp) = P_MULTIPLIER;
    E_LOCATION(tmp) = P_LOCATION;
    E_EXP_TYPE(tmp) = P_EXP_TYPE;
    E_PROD_TYPE(tmp) = P_PROD_TYPE;
    E_NUMBER(tmp) = P_QUANTITY;
    E_UNIT(tmp) = P_UNIT;
    E_CATEGORY(tmp) = notappl;

    recode P_PURPOSE => tmp2;
        1 => 51;
        2 => 52;
        3 => 58;
    endrecode;

    E_BENEFICIARY(tmp) = tmp2;
    E_DURABILITY(tmp) = tmpDura;
    E_DESCRIPTION(tmp) = "[diary purchase]"+tolower(P_DESCRIPTION);
endif;
PROC F_LINE_NUMBER

// first pass coicop correction on 21/3/2016
if ID07 in  102 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'toast' then
impute(F_COICOP,   11100301    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  102 and F_LINE_NUMBER in    17  and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  102 and F_LINE_NUMBER in    30  and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  102 and F_LINE_NUMBER in    37  and strip(F_DESCRIPTION[1:5]) = 'green' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  103 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
```

```
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  104 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  104 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = '4x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  105 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = '1x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  105 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'curry' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  105 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'tukit' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  105 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'pulak' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  106 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = '1x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  106 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'cornb' then
impute(F_COICOP,   11202001    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  106 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = '2x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'scoop' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'tukit' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = '1x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'puta'  then
impute(F_COICOP,   11100304    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'luau'  then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'tukit' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    18  and strip(F_DESCRIPTION[1:5]) = 'chow'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    20  and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  108 and F_LINE_NUMBER in    22  and strip(F_DESCRIPTION[1:5]) = '2x sc' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  111 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = '6x hu' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  111 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = '6x hu' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  111 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  111 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'uto t' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'baked' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  114 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'banba' then
impute(F_COICOP,   11604202    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  116 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  116 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'bbq c' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  117 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'fried' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
```

```
if ID07 in  117 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'scone' then
impute(F_COICOP,   11100701    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  117 and F_LINE_NUMBER in    22  and strip(F_DESCRIPTION[1:5]) = 'raw f' then
impute(F_COICOP,   11302232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  117 and F_LINE_NUMBER in    24  and strip(F_DESCRIPTION[1:5]) = 'salad' then
impute(F_COICOP,   11704518    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  118 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  119 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'laupe' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  119 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'soup'  then
impute(F_COICOP,   11604533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  120 and F_LINE_NUMBER in    24  and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  120 and F_LINE_NUMBER in    29  and strip(F_DESCRIPTION[1:5]) = 'coco'  then
impute(F_COICOP,   11100102    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  120 and F_LINE_NUMBER in    31  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  120 and F_LINE_NUMBER in    36  and strip(F_DESCRIPTION[1:5]) = 'cabba' then
impute(F_COICOP,   11704517    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  121 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  121 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = '3x pi' then
impute(F_COICOP,   11201502    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  121 and F_LINE_NUMBER in    20  and strip(F_DESCRIPTION[1:5]) = '10x c' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'chops' then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'beef'  then
impute(F_COICOP,   11201103    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'panda' then
impute(F_COICOP,   11604233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    19  and strip(F_DESCRIPTION[1:5]) = 'beef'  then
impute(F_COICOP,   11201107    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  124 and F_LINE_NUMBER in    22  and strip(F_DESCRIPTION[1:5]) = '5x co' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  125 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'uto 4' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  125 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = '2x co' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  125 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = '2x co' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  126 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'baked' then
impute(F_COICOP,   11302232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  126 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  126 and F_LINE_NUMBER in    20  and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302227    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'scoop' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'lamp'  then
impute(F_COICOP,   11201303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'reef'  then
impute(F_COICOP,   11302233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'cocon' then
```

```
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  135 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'janda' then
impute(F_COICOP,   32110501    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'pawpa' then
impute(F_COICOP,   11604225    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'salte' then
impute(F_COICOP,   11201103    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'sausa' then
impute(F_COICOP,   11201906    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  138 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'salte' then
impute(F_COICOP,   11201103    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'raw f' then
impute(F_COICOP,   11302232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302227    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11100301    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  140 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'fresh' then
impute(F_COICOP,   11302232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302228    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'flyin' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'soap'  then
impute(F_COICOP,   56116910    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'carro' then
impute(F_COICOP,   11704508    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'eggs'  then
impute(F_COICOP,   11403501    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'orang' then
impute(F_COICOP,   11604203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'apple' then
impute(F_COICOP,   11604201    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'eel'   then
impute(F_COICOP,   11302204    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'panda' then
impute(F_COICOP,   11604233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'cake'  then
impute(F_COICOP,   11100601    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    17  and strip(F_DESCRIPTION[1:5]) = 'orang' then
impute(F_COICOP,   11604203    );  errmsg("%s",F_DESCRIPTION); endif;
```

```
if ID07 in  141 and F_LINE_NUMBER in    18  and strip(F_DESCRIPTION[1:5]) = 'apple' then
impute(F_COICOP,   11604201    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    20  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    21  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    22  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    24  and strip(F_DESCRIPTION[1:5]) = 'cocon' then
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  141 and F_LINE_NUMBER in    25  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'scoop' then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'mango' then
impute(F_COICOP,   11604221    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'viena' then
impute(F_COICOP,   11201906    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'talo'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  143 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'cocoa' then
impute(F_COICOP,   11100102    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'fried' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  147 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'stew'  then
impute(F_COICOP,   11202003    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'uto'   then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'fresh' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'keros' then
impute(F_COICOP,   45312801    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'cocon' then
```

```
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'steam' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'pawpa' then
impute(F_COICOP,   11604225    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'cocon' then
impute(F_COICOP,   11403201    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  148 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  151 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fresh' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  151 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  151 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'lobst' then
impute(F_COICOP,   11302305    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'tuna'  then
impute(F_COICOP,   11302237    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'petro' then
impute(F_COICOP,   72219602    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  154 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'cocon' then
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  155 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'laval' then
impute(F_COICOP,   31109201    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  155 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'petro' then
impute(F_COICOP,   72219602    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  155 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  155 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  155 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'laupe' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'keros' then
impute(F_COICOP,   45312801    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'fried' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'pawpa' then
impute(F_COICOP,   11604225    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'mutto' then
impute(F_COICOP,   11201303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  158 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'soup'  then
impute(F_COICOP,   11604225    );  errmsg("%s",F_DESCRIPTION); endif;
```

```
if ID07 in  158 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'potat' then
impute(F_COICOP,   11704601    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'cocon' then
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'potat' then
impute(F_COICOP,   11704601    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'panca' then
impute(F_COICOP,   11100303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    14  and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'pork'  then
impute(F_COICOP,   11201203    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    17  and strip(F_DESCRIPTION[1:5]) = 'water' then
impute(F_COICOP,   12207101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    18  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  159 and F_LINE_NUMBER in    19  and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'keros' then
impute(F_COICOP,   45312801    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'sweet' then
impute(F_COICOP,   11100518    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'lobst' then
impute(F_COICOP,   11302305    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  160 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  203 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'donut' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  206 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  206 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'dough' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  206 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'dougn' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  206 and F_LINE_NUMBER in    28  and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  206 and F_LINE_NUMBER in    32  and strip(F_DESCRIPTION[1:5]) = 'sandw' then
impute(F_COICOP,   111126401   );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  207 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  207 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  207 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'donut' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  207 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'panda' then
```

```
impute(F_COICOP,   11604233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  208 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  209 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  211 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'swamp' then
impute(F_COICOP,   11704607    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  211 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'bun'   then
impute(F_COICOP,   11100401    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  220 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  232 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'rice'  then
impute(F_COICOP,   11100101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  235 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'panda' then
impute(F_COICOP,   11604233    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  238 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'donut' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  302 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'bowl'  then
impute(F_COICOP,   11100901    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  302 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'flour' then
impute(F_COICOP,   11100202    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  302 and F_LINE_NUMBER in    24  and strip(F_DESCRIPTION[1:5]) = 'chick' then
impute(F_COICOP,   11201511    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'whole' then
impute(F_COICOP,   11201510    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'ice-c' then
impute(F_COICOP,   11805801    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'fresh' then
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'grate' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'bottl' then
impute(F_COICOP,   12207101    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'whole' then
impute(F_COICOP,   11201510    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    20  and strip(F_DESCRIPTION[1:5]) = 'cake'  then
impute(F_COICOP,   11100601    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  305 and F_LINE_NUMBER in    26  and strip(F_DESCRIPTION[1:5]) = 'ice c' then
impute(F_COICOP,   11805801    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  306 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  306 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bag c' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  307 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  307 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  307 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  308 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'bag o' then
impute(F_COICOP,   11604210    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'big b' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'flour' then
impute(F_COICOP,   11100202    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'sugar' then
impute(F_COICOP,   11804902    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'taro'  then
impute(F_COICOP,   11704603    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    8   and strip(F_DESCRIPTION[1:5]) = 'corn'  then
impute(F_COICOP,   11202001    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'lamb'  then
impute(F_COICOP,   11201303    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  310 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'carra' then
impute(F_COICOP,   11704508    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  311 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = '4 x 7' then
impute(F_COICOP,   21308003    );  errmsg("%s",F_DESCRIPTION); endif;
```

```
if ID07 in  318 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  319 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  319 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  319 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  320 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'cabba' then
impute(F_COICOP,   11714517    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  326 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  326 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'cooke' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  327 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'beer'  then
impute(F_COICOP,   21308004    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  327 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = '1plat' then
impute(F_COICOP,   111126401   );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  327 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = '1 bre' then
impute(F_COICOP,   111126401   );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  328 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  329 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11604207    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  332 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'loaf'  then
impute(F_COICOP,   11100301    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  332 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'petro' then
impute(F_COICOP,   72219602    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  332 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  336 and F_LINE_NUMBER in    11  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  337 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'pele'  then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  337 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  337 and F_LINE_NUMBER in    34  and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  338 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fresh' then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  339 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302236    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  339 and F_LINE_NUMBER in    9   and strip(F_DESCRIPTION[1:5]) = 'baked' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  341 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  341 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'chop'  then
impute(F_COICOP,   11100919    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  341 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  341 and F_LINE_NUMBER in    15  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  341 and F_LINE_NUMBER in    16  and strip(F_DESCRIPTION[1:5]) = 'drink' then
impute(F_COICOP,   11604209    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  347 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'avoca' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  348 and F_LINE_NUMBER in    1   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302212    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  348 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  348 and F_LINE_NUMBER in    12  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  348 and F_LINE_NUMBER in    30  and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  350 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'fish'  then
impute(F_COICOP,   11302224    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  351 and F_LINE_NUMBER in    6   and strip(F_DESCRIPTION[1:5]) = 'sprou' then
impute(F_COICOP,   11604232    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  351 and F_LINE_NUMBER in    7   and strip(F_DESCRIPTION[1:5]) = 'donut' then
impute(F_COICOP,   11100302    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  351 and F_LINE_NUMBER in    13  and strip(F_DESCRIPTION[1:5]) = 'fish(' then
```

```
impute(F_COICOP,   11302212    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  352 and F_LINE_NUMBER in    5   and strip(F_DESCRIPTION[1:5]) = 'avoca' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  353 and F_LINE_NUMBER in    2   and strip(F_DESCRIPTION[1:5]) = 'bread' then
impute(F_COICOP,   11704805    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  353 and F_LINE_NUMBER in    3   and strip(F_DESCRIPTION[1:5]) = 'laupe' then
impute(F_COICOP,   11704533    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  353 and F_LINE_NUMBER in    4   and strip(F_DESCRIPTION[1:5]) = 'fanti' then
impute(F_COICOP,   11100518    );  errmsg("%s",F_DESCRIPTION); endif;
if ID07 in  353 and F_LINE_NUMBER in    10  and strip(F_DESCRIPTION[1:5]) = 'ice c' then
impute(F_COICOP,   11805803    );  errmsg("%s",F_DESCRIPTION); endif;

PROC F_HIES_CODE
// second pass
recode  ID07    :   F_LINE_NUMBER   =>  F_COICOP    ;
    102 :   3   =>  11302238    ;
    102 :   4   =>  11100101    ;
    102 :   6   =>  11302238    ;
    102 :   10  =>  11302238    ;
    102 :   11  =>  11100101    ;
    102 :   15  =>  11302238    ;
    102 :   16  =>  11100101    ;
    102 :   17  =>  11100403    ;
    102 :   20  =>  11100101    ;
    102 :   23  =>  11302238    ;
    102 :   26  =>  11302238    ;
    102 :   30  =>  11100403    ;
    102 :   31  =>  11302238    ;
    102 :   33  =>  11100101    ;
    103 :   2   =>  11302238    ;
    103 :   5   =>  11302238    ;
    104 :   2   =>  11302238    ;
    104 :   4   =>  11302238    ;
    105 :   3   =>  111126503   ;
    105 :   8   =>  11302238    ;
    106 :   2   =>  11202001    ;
    108 :   1   =>  11100403    ;
    108 :   6   =>  11302238    ;
    108 :   10  =>  111126598   ;
    108 :   11  =>  11201203    ;
    108 :   14  =>  11100404    ;
    108 :   18  =>  111126598   ;
    108 :   20  =>  111126598   ;
    108 :   23  =>  11302238    ;
    111 :   5   =>  11302238    ;
    111 :   9   =>  11302238    ;
    111 :   12  =>  11302238    ;
    113 :   1   =>  999999998   ;
    114 :   1   =>  11302234    ;
    114 :   2   =>  11302238    ;
    114 :   5   =>  11302238    ;
    114 :   7   =>  11201203    ;
    114 :   9   =>  11302238    ;
    114 :   10  =>  11302238    ;
    114 :   11  =>  11302238    ;
    114 :   12  =>  11302238    ;
    114 :   13  =>  11302238    ;
    114 :   16  =>  111126598   ;
    114 :   18  =>  11302238    ;
    114 :   19  =>  11302238    ;
    116 :   2   =>  11201203    ;
    116 :   9   =>  11302238    ;
    117 :   1   =>  11201203    ;
    117 :   3   =>  11302238    ;
    117 :   6   =>  11302238    ;
    117 :   9   =>  11100402    ;
    117 :   14  =>  11302238    ;
    117 :   18  =>  11302238    ;
    117 :   22  =>  11302238    ;
    117 :   23  =>  11302238    ;
    117 :   27  =>  11201203    ;
```

```
117 :   28  =>  11302238    ;
117 :   32  =>  11302238    ;
118 :   3   =>  11100403    ;
119 :   5   =>  11302238    ;
119 :   8   =>  11302238    ;
119 :   11  =>  11302238    ;
119 :   17  =>  11302238    ;
120 :   2   =>  11302238    ;
120 :   3   =>  11201203    ;
120 :   7   =>  111126598   ;
120 :   9   =>  11302238    ;
120 :   12  =>  11302238    ;
120 :   14  =>  11201201    ;
120 :   23  =>  111126598   ;
120 :   24  =>  11100403    ;
120 :   25  =>  11302238    ;
120 :   28  =>  11302238    ;
120 :   29  =>  11100101    ;
120 :   31  =>  11302238    ;
120 :   33  =>  11302238    ;
121 :   1   =>  11100403    ;
121 :   7   =>  11302238    ;
121 :   11  =>  11302238    ;
121 :   15  =>  11201203    ;
121 :   19  =>  11302238    ;
124 :   1   =>  111126598   ;
124 :   5   =>  11302238    ;
124 :   8   =>  111126598   ;
124 :   13  =>  11302238    ;
124 :   16  =>  11302238    ;
125 :   2   =>  11201303    ;
125 :   6   =>  11604232    ;
125 :   9   =>  11302238    ;
126 :   4   =>  111126598   ;
126 :   15  =>  11201203    ;
126 :   18  =>  11302238    ;
126 :   20  =>  111126598   ;
132 :   2   =>  11302238    ;
132 :   4   =>  11302238    ;
134 :   3   =>  11302238    ;
135 :   3   =>  11302238    ;
135 :   10  =>  11302233    ;
135 :   13  =>  11302232    ;
136 :   1   =>  11302238    ;
136 :   5   =>  11302238    ;
136 :   7   =>  11302238    ;
138 :   3   =>  32110201    ;
138 :   9   =>  11302238    ;
140 :   4   =>  11302238    ;
140 :   5   =>  11302238    ;
140 :   7   =>  11302507    ;
141 :   2   =>  11302234    ;
141 :   3   =>  11302233    ;
141 :   4   =>  11302232    ;
141 :   11  =>  11302232    ;
141 :   13  =>  11302238    ;
141 :   19  =>  11302233    ;
141 :   22  =>  11302238    ;
141 :   25  =>  11302238    ;
142 :   1   =>  11302238    ;
142 :   4   =>  11302238    ;
143 :   2   =>  11302238    ;
143 :   3   =>  11302231    ;
143 :   7   =>  11302238    ;
143 :   9   =>  11302232    ;
147 :   2   =>  11302232    ;
147 :   5   =>  11302232    ;
147 :   6   =>  11100101    ;
147 :   7   =>  11302238    ;
147 :   11  =>  11302238    ;
148 :   2   =>  11302238    ;
```

```
148 :   6   =>  11302238    ;
148 :   9   =>  11504005    ;
151 :   1   =>  11302238    ;
154 :   3   =>  11302238    ;
154 :   7   =>  11302232    ;
155 :   4   =>  11302238    ;
158 :   6   =>  11100403    ;
158 :   7   =>  11302238    ;
159 :   7   =>  11302238    ;
159 :   9   =>  11100403    ;
159 :   11  =>  11302238    ;
159 :   13  =>  11302238    ;
160 :   3   =>  11302238    ;
203 :   5   =>  11100402    ;
203 :   6   =>  11604232    ;
203 :   10  =>  11302233    ;
203 :   16  =>  11100203    ;
204 :   5   =>  11302233    ;
206 :   3   =>  11201203    ;
206 :   5   =>  11100402    ;
206 :   8   =>  11100402    ;
206 :   11  =>  11604232    ;
206 :   17  =>  11302233    ;
206 :   22  =>  11201107    ;
206 :   28  =>  111126598   ;
206 :   31  =>  11302233    ;
207 :   9   =>  11604232    ;
207 :   10  =>  11100402    ;
207 :   15  =>  11302233    ;
207 :   16  =>  11604232    ;
207 :   23  =>  11302233    ;
208 :   7   =>  11604232    ;
208 :   10  =>  11302234    ;
208 :   19  =>  11302233    ;
208 :   21  =>  11604232    ;
209 :   12  =>  11302233    ;
210 :   8   =>  11302233    ;
210 :   11  =>  11100203    ;
210 :   17  =>  11302233    ;
211 :   7   =>  11604232    ;
211 :   9   =>  11302233    ;
219 :   1   =>  11302238    ;
219 :   2   =>  11302238    ;
220 :   1   =>  11302238    ;
220 :   2   =>  11302238    ;
220 :   3   =>  11302238    ;
220 :   4   =>  11302238    ;
220 :   7   =>  11302238    ;
222 :   1   =>  11302225    ;
227 :   2   =>  11302212    ;
232 :   3   =>  11100101    ;
232 :   5   =>  11302238    ;
232 :   6   =>  11302238    ;
236 :   3   =>  11302238    ;
238 :   1   =>  11100402    ;
238 :   7   =>  11302219    ;
241 :   1   =>  11302238    ;
255 :   4   =>  11302225    ;
302 :   4   =>  11302238    ;
302 :   6   =>  11100203    ;
302 :   26  =>  11302238    ;
305 :   1   =>  11302238    ;
305 :   3   =>  999999998   ;
305 :   7   =>  11805802    ;
305 :   10  =>  11302238    ;
305 :   15  =>  11302238    ;
306 :   1   =>  11302238    ;
307 :   1   =>  11302238    ;
307 :   3   =>  11302238    ;
308 :   1   =>  11302238    ;
310 :   2   =>  11302238    ;
```

```
310 :   4   =>  11604232    ;
310 :   5   =>  11100203    ;
310 :   6   =>  11804903    ;
310 :   9   =>  999999998   ;
311 :   5   =>  11302238    ;
320 :   3   =>  11704517    ;
322 :   2   =>  999999998   ;
327 :   1   =>  21308003    ;
327 :   2   =>  11100701    ;
328 :   20  =>  999999998   ;
329 :   3   =>  999999998   ;
332 :   4   =>  11302233    ;
332 :   8   =>  11302238    ;
333 :   1   =>  11302238    ;
333 :   4   =>  11302238    ;
336 :   4   =>  11302238    ;
336 :   6   =>  11302238    ;
336 :   10  =>  11302238    ;
336 :   11  =>  11604232    ;
336 :   12  =>  11302238    ;
336 :   16  =>  11302238    ;
336 :   19  =>  11302238    ;
336 :   23  =>  11100203    ;
336 :   24  =>  11804903    ;
337 :   8   =>  11302238    ;
337 :   10  =>  11604232    ;
337 :   14  =>  11302238    ;
337 :   16  =>  11302238    ;
337 :   29  =>  11302238    ;
337 :   33  =>  11302238    ;
337 :   34  =>  111126598   ;
337 :   37  =>  11804903    ;
337 :   38  =>  11100203    ;
337 :   42  =>  11302238    ;
337 :   43  =>  11504005    ;
338 :   1   =>  11302238    ;
338 :   2   =>  11302238    ;
339 :   1   =>  11302238    ;
339 :   10  =>  11302238    ;
340 :   2   =>  11302238    ;
340 :   7   =>  11302238    ;
340 :   9   =>  11302238    ;
340 :   10  =>  11604232    ;
340 :   11  =>  11302238    ;
340 :   16  =>  11302224    ;
340 :   17  =>  11302238    ;
341 :   1   =>  11302238    ;
341 :   3   =>  11302238    ;
341 :   4   =>  11604232    ;
341 :   7   =>  111126598   ;
341 :   8   =>  11302238    ;
341 :   10  =>  11504005    ;
341 :   13  =>  11604232    ;
341 :   14  =>  11302238    ;
341 :   15  =>  11604232    ;
341 :   24  =>  11100203    ;
341 :   25  =>  11804903    ;
341 :   28  =>  11302238    ;
342 :   2   =>  11100203    ;
342 :   3   =>  11804903    ;
342 :   4   =>  11302238    ;
343 :   1   =>  11302238    ;
343 :   3   =>  11302238    ;
343 :   4   =>  11302238    ;
346 :   1   =>  11302238    ;
346 :   2   =>  11302238    ;
346 :   3   =>  11302238    ;
347 :   3   =>  11604231    ;
347 :   4   =>  11302238    ;
348 :   6   =>  11604232    ;
348 :   7   =>  11302238    ;
```

```
    348 :   9   =>  11302238    ;
    348 :   11  =>  11302232    ;
    348 :   12  =>  11604232    ;
    348 :   14  =>  11302238    ;
    348 :   16  =>  11302238    ;
    348 :   20  =>  11302238    ;
    348 :   23  =>  11804903    ;
    348 :   24  =>  11100203    ;
    348 :   25  =>  11302238    ;
    348 :   29  =>  11302238    ;
    348 :   30  =>  11604232    ;
    349 :   1   =>  11302238    ;
    349 :   2   =>  11302238    ;
    349 :   3   =>  11302238    ;
    349 :   5   =>  11302238    ;
    349 :   6   =>  11302238    ;
    349 :   12  =>  11302238    ;
    350 :   1   =>  11302238    ;
    350 :   3   =>  11100203    ;
    350 :   4   =>  11804903    ;
    351 :   6   =>  11604232    ;
    351 :   7   =>  11100402    ;
    351 :   19  =>  11302238    ;
    351 :   21  =>  11302238    ;
    351 :   25  =>  11302238    ;
    352 :   2   =>  11302238    ;
    352 :   5   =>  11604231    ;
    353 :   9   =>  11302238    ;
    353 :   10  =>  11805801    ;
    353 :   12  =>  11302238    ;
    354 :   2   =>  11302238    ;
    354 :   3   =>  11302238    ;
    356 :   1   =>  11302238    ;
    356 :   3   =>  11100203    ;
    356 :   4   =>  11804903    ;
    356 :   5   =>  11302238    ;
endrecode;

$ = 540 + F_DAY / 1000;
PROC F_AMOUNT

i = 0;

recode  ID07    :   F_LINE_NUMBER   =>  i   ;
    134 :   3   =>  10  ;
    134 :   4   =>  4   ;
    134 :   5   =>  5   ;
    134 :   6   =>  7.5 ;
    138 :   6   =>  3.5 ;
    238 :   7   =>  10  ;
        :       =>  0;
endrecode;

if i > 0 then
    impute($,i);
endif;
PROC F_MULTIPLIER
if ID07 in 308,318,118,326,250,134,105 then
    if ID07 in 308 then if F_DAY in 1:7 then F_MULTIPLIER = 0; else F_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 318 then if F_DAY in 1:7 then F_MULTIPLIER = 0; else F_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 118 then if F_DAY in 1:7 then F_MULTIPLIER = 52; else F_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 326 then if F_DAY in 1:7 then F_MULTIPLIER = 52; else F_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 250 then if F_DAY in 1:7 then F_MULTIPLIER = 52; else F_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 134 then if F_DAY in 1:7 then F_MULTIPLIER = 52; else F_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 105 then if F_DAY in 1:7 then F_MULTIPLIER = 52; else F_MULTIPLIER = 0;
```

```
    endif; endif;
else
    F_MULTIPLIER = 26;
endif;

F_AN_AMT = F_AMOUNT * F_MULTIPLIER;
F_WGHT_AN_AMT = F_AN_AMT * WEIGHT;


PROC F_LOCATION
$ = 1;
PROC F_EXP_TYPE
 $ = 8;

PROC F_LOCAL
$ = notappl;
PROC F_PURPOSE
$ = 1;
PROC HC_LINE_NUMBER
// first pass coicop correction on 21/3/2016
if ID07 in  108 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  108 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  108 and HC_LINE_NUMBER in   13  and strip(HC_DESCRIPTION[1:5]) = 'pulak'    then
impute(HC_COICOP,  11704607    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  108 and HC_LINE_NUMBER in   14  and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  109 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  109 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  109 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  109 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  109 and HC_LINE_NUMBER in   10  and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'green'    then
impute(HC_COICOP,  11604209    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604209    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = 'pawda'    then
impute(HC_COICOP,  11604233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   9   and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   12  and strip(HC_DESCRIPTION[1:5]) = 'lu'   then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   14  and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   15  and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   18  and strip(HC_DESCRIPTION[1:5]) = 'uto'  then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   20  and strip(HC_DESCRIPTION[1:5]) = 'roast'    then
impute(HC_COICOP,  11201510    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   24  and strip(HC_DESCRIPTION[1:5]) = 'panda'    then
impute(HC_COICOP,  11604233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   26  and strip(HC_DESCRIPTION[1:5]) = 'panda'    then
impute(HC_COICOP,  11604233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  111 and HC_LINE_NUMBER in   28  and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  117 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  117 and HC_LINE_NUMBER in   11  and strip(HC_DESCRIPTION[1:5]) = 'varie'    then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  117 and HC_LINE_NUMBER in   18  and strip(HC_DESCRIPTION[1:5]) = 'laupe'    then
```

```
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  118 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = '10x u'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  118 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = '4x br'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  118 and HC_LINE_NUMBER in   9   and strip(HC_DESCRIPTION[1:5]) = '10x f'    then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  125 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = '5x co'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  125 and HC_LINE_NUMBER in   10  and strip(HC_DESCRIPTION[1:5]) = 'laupe'    then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  126 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'chick'    then
impute(HC_COICOP,  11201511    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  126 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = '1x ti'    then
impute(HC_COICOP,  11302509    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  126 and HC_LINE_NUMBER in   11  and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11100401    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  127 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11403201    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  132 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'tuna' then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  132 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  132 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = 'tuna' then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  134 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  134 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  135 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'bonit'    then
impute(HC_COICOP,  11302231    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  138 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  138 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'banan'    then
impute(HC_COICOP,  11604202    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  148 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'tuna' then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  148 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'mahim'    then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  148 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'tuna' then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  148 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'yello'    then
impute(HC_COICOP,  11302229    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  148 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  150 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  150 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  151 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  151 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  152 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  152 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'pelag'    then
impute(HC_COICOP,  11302234    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  152 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302213    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  152 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  154 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  154 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'fresh'    then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  154 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  155 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  155 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'mahim'    then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
```

```
if ID07 in  155 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  155 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  155 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  155 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'tuna' then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  158 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  158 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'fried'    then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'taro' then
impute(HC_COICOP,  11704603    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'casav'    then
impute(HC_COICOP,  11704604    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'panca'    then
impute(HC_COICOP,  11100303    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'reef' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = 'pork' then
impute(HC_COICOP,  11201203    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   9   and strip(HC_DESCRIPTION[1:5]) = 'fried'    then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   10  and strip(HC_DESCRIPTION[1:5]) = 'rice' then
impute(HC_COICOP,  11100102    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  159 and HC_LINE_NUMBER in   11  and strip(HC_DESCRIPTION[1:5]) = 'garde'    then
impute(HC_COICOP,  11302232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  206 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'swamp'    then
impute(HC_COICOP,  11704607    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  224 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'grope'    then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  241 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'barra'    then
impute(HC_COICOP,  11302201    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  241 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'taro' then
impute(HC_COICOP,  11704603    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  301 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  301 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  305 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  308 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   4   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'banan'    then
impute(HC_COICOP,  11604202    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
```

```
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = 'banan'    then
impute(HC_COICOP,  11604202    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   9   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   10  and strip(HC_DESCRIPTION[1:5]) = 'banan'    then
impute(HC_COICOP,  11604202    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  310 and HC_LINE_NUMBER in   11  and strip(HC_DESCRIPTION[1:5]) = 'cocnu'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   7   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  311 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  319 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  320 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'uto,' then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  327 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'flyin'    then
impute(HC_COICOP,  11302236    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  327 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'clams'    then
impute(HC_COICOP,  11302310    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  328 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'pele' then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  328 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'pele' then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  328 and HC_LINE_NUMBER in   14  and strip(HC_DESCRIPTION[1:5]) = 'lemon'    then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  328 and HC_LINE_NUMBER in   21  and strip(HC_DESCRIPTION[1:5]) = 'pele' then
impute(HC_COICOP,  11704533    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  329 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'bread'    then
impute(HC_COICOP,  11604207    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  332 and HC_LINE_NUMBER in   5   and strip(HC_DESCRIPTION[1:5]) = 'fish' then
impute(HC_COICOP,  11302233    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  342 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'sprou'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  347 and HC_LINE_NUMBER in   10  and strip(HC_DESCRIPTION[1:5]) = 'clams'    then
impute(HC_COICOP,  11302310    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  349 and HC_LINE_NUMBER in   1   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  353 and HC_LINE_NUMBER in   2   and strip(HC_DESCRIPTION[1:5]) = 'ripe' then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  353 and HC_LINE_NUMBER in   8   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  353 and HC_LINE_NUMBER in   11  and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604232    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  354 and HC_LINE_NUMBER in   3   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
if ID07 in  354 and HC_LINE_NUMBER in   6   and strip(HC_DESCRIPTION[1:5]) = 'cocon'    then
impute(HC_COICOP,  11604210    );  errmsg("%s",HC_DESCRIPTION);    endif;
PROC HC_HIES_CODE
// second pass
recode  ID07    :   HC_LINE_NUMBER  =>  HC_COICOP   ;
    103 :   3   =>  999999998   ;
    103 :   4   =>  999999998   ;
    108 :   4   =>  999999998   ;
    108 :   6   =>  11302238    ;
    108 :   9   =>  999999998   ;
    108 :   10  =>  11302238    ;
    108 :   12  =>  11302238    ;
    108 :   16  =>  11302238    ;
    109 :   1   =>  11302238    ;
```

```
109 :   4   =>  11302238    ;
109 :   5   =>  11302238    ;
109 :   7   =>  11302238    ;
109 :   9   =>  11302238    ;
109 :   11  =>  11302238    ;
111 :   3   =>  11302238    ;
111 :   4   =>  999999998   ;
111 :   6   =>  11302238    ;
111 :   11  =>  11302238    ;
111 :   13  =>  11302238    ;
111 :   16  =>  999999998   ;
111 :   19  =>  999999998   ;
111 :   20  =>  999999998   ;
111 :   21  =>  999999998   ;
111 :   22  =>  11302238    ;
111 :   30  =>  11504005    ;
117 :   1   =>  999999998   ;
117 :   3   =>  11302238    ;
117 :   4   =>  11302238    ;
117 :   5   =>  999999998   ;
117 :   7   =>  11302238    ;
117 :   9   =>  11302238    ;
117 :   11  =>  11302238    ;
117 :   13  =>  11302238    ;
117 :   15  =>  11302238    ;
117 :   16  =>  999999998   ;
117 :   17  =>  11302238    ;
117 :   19  =>  999999998   ;
117 :   20  =>  11302238    ;
117 :   22  =>  999999998   ;
117 :   23  =>  11302238    ;
117 :   24  =>  999999998   ;
118 :   1   =>  11302238    ;
124 :   2   =>  11302238    ;
124 :   3   =>  11302238    ;
124 :   6   =>  11302238    ;
124 :   7   =>  999999998   ;
125 :   3   =>  11302238    ;
125 :   7   =>  11302238    ;
125 :   8   =>  999999998   ;
125 :   9   =>  11302238    ;
126 :   2   =>  11302238    ;
126 :   3   =>  999999998   ;
126 :   4   =>  999999998   ;
126 :   5   =>  999999998   ;
126 :   6   =>  999999998   ;
126 :   8   =>  999999998   ;
126 :   10  =>  999999998   ;
126 :   11  =>  999999998   ;
126 :   12  =>  999999998   ;
126 :   13  =>  11302238    ;
126 :   14  =>  999999998   ;
126 :   15  =>  999999998   ;
127 :   1   =>  11302238    ;
127 :   3   =>  11302238    ;
127 :   5   =>  11504005    ;
132 :   8   =>  11302238    ;
134 :   1   =>  11302238    ;
150 :   1   =>  11302238    ;
154 :   3   =>  11302238    ;
159 :   1   =>  11302238    ;
159 :   5   =>  999999998   ;
159 :   7   =>  999999998   ;
159 :   9   =>  11302238    ;
159 :   10  =>  999999998   ;
206 :   1   =>  11302212    ;
206 :   9   =>  11604232    ;
206 :   12  =>  11604232    ;
207 :   3   =>  11604232    ;
207 :   4   =>  11604232    ;
208 :   2   =>  11604232    ;
```

```
208 :   4   =>  11604232    ;
208 :   7   =>  11604232    ;
208 :   8   =>  11302225    ;
220 :   1   =>  11302238    ;
220 :   4   =>  999999998   ;
224 :   6   =>  11302225    ;
226 :   4   =>  11504005    ;
232 :   4   =>  11302238    ;
241 :   7   =>  11302238    ;
247 :   3   =>  11302225    ;
249 :   8   =>  11302225    ;
254 :   8   =>  11302225    ;
301 :   1   =>  11302238    ;
301 :   2   =>  11504005    ;
301 :   3   =>  11302238    ;
306 :   1   =>  11302238    ;
308 :   1   =>  11302238    ;
310 :   1   =>  11302238    ;
310 :   2   =>  11302238    ;
310 :   4   =>  11302238    ;
310 :   8   =>  11302238    ;
310 :   9   =>  11302238    ;
311 :   4   =>  11302238    ;
319 :   3   =>  999999998   ;
323 :   1   =>  11302238    ;
323 :   2   =>  999999998   ;
323 :   4   =>  999999998   ;
323 :   5   =>  999999998   ;
323 :   6   =>  999999998   ;
323 :   8   =>  999999998   ;
323 :   10  =>  999999998   ;
323 :   12  =>  999999998   ;
323 :   13  =>  999999998   ;
323 :   16  =>  999999998   ;
323 :   17  =>  999999998   ;
323 :   20  =>  999999998   ;
323 :   22  =>  999999998   ;
323 :   24  =>  999999998   ;
323 :   25  =>  999999998   ;
323 :   26  =>  999999998   ;
323 :   28  =>  999999998   ;
323 :   29  =>  999999998   ;
323 :   30  =>  999999998   ;
326 :   1   =>  11302238    ;
326 :   6   =>  11302238    ;
327 :   1   =>  11302216    ;
327 :   2   =>  11302216    ;
328 :   11  =>  999999998   ;
328 :   12  =>  999999998   ;
328 :   13  =>  999999998   ;
332 :   2   =>  11504004    ;
332 :   4   =>  11504004    ;
332 :   5   =>  11302225    ;
333 :   1   =>  11302238    ;
338 :   4   =>  11302238    ;
339 :   1   =>  11302238    ;
340 :   3   =>  11604233    ;
340 :   5   =>  11704533    ;
342 :   1   =>  11302238    ;
343 :   1   =>  11302238    ;
343 :   2   =>  11302238    ;
343 :   3   =>  11302238    ;
346 :   2   =>  11302238    ;
346 :   3   =>  11302238    ;
346 :   5   =>  11302238    ;
346 :   8   =>  11302238    ;
346 :   9   =>  11302238    ;
346 :   11  =>  11302238    ;
347 :   1   =>  11302238    ;
347 :   2   =>  11302238    ;
347 :   3   =>  11302238    ;
```

```
    347 :   4   =>  11302238    ;
    347 :   6   =>  11302238    ;
    347 :   7   =>  11302238    ;
    347 :   8   =>  11302238    ;
    349 :   2   =>  11302238    ;
    349 :   4   =>  11302238    ;
    349 :   5   =>  11302238    ;
    349 :   6   =>  11302238    ;
    349 :   7   =>  11302238    ;
    349 :   8   =>  11302238    ;
    349 :   9   =>  11504004    ;
    353 :   1   =>  11604210    ;
    354 :   1   =>  11302238    ;
    354 :   2   =>  11302238    ;
    354 :   4   =>  11302238    ;
    354 :   5   =>  11302238    ;
    354 :   7   =>  11302238    ;
    354 :   8   =>  11302238    ;
    354 :   9   =>  11302238    ;
endrecode;

$ = 551 + HC_DAY / 1000;
PROC HC_COICOP
recode  ID07    :   HC_LINE_NUMBER  =>  HC_COICOP   ;
    125 :   4   =>  011302227   ;
    135 :   1   =>  011302227   ;
    203 :   9   =>  011302227   ;
    327 :   4   =>  011302227   ;
    148 :   2   =>  011302239   ;
    155 :   2   =>  011302239   ;
    159 :   11  =>  011704533   ;
    327 :   7   =>  011302213   ;
    208 :   6   =>  011302240   ;
    232 :   2   =>  011302241   ;
    221 :   1   =>  011302201   ;
    327 :   6   =>  011302241   ;
endrecode;
PROC HC_AMOUNT

i = 0;
recode  ID07    :   HC_LINE_NUMBER  =>  i;
    134 :   1   =>  10  ;
    134 :   2   =>  4   ;
    310 :   9   =>  10  ;
        :       =>  0   ;
endrecode;

if i > 0 then
    impute($, i);
endif
PROC HC_MULTIPLIER
if ID07 in 308,318,118,326,250,134,105 then
    if ID07 in 308 then if HC_DAY in 1:7 then HC_MULTIPLIER = 0; else HC_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 318 then if HC_DAY in 1:7 then HC_MULTIPLIER = 0; else HC_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 118 then if HC_DAY in 1:7 then HC_MULTIPLIER = 52; else HC_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 326 then if HC_DAY in 1:7 then HC_MULTIPLIER = 52; else HC_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 250 then if HC_DAY in 1:7 then HC_MULTIPLIER = 52; else HC_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 134 then if HC_DAY in 1:7 then HC_MULTIPLIER = 52; else HC_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 105 then if HC_DAY in 1:7 then HC_MULTIPLIER = 52; else HC_MULTIPLIER = 0;
    endif; endif;
else
    HC_MULTIPLIER = 26;
endif;

HC_AN_AMT = HC_AMOUNT * HC_MULTIPLIER;
```

```
HC_WGHT_AN_AMT = HC_AN_AMT * WEIGHT;


PROC HC_LOCATION
$ = 1;
PROC HC_EXP_TYPE
$ = 2;
PROC HC_LOCAL
$ = notappl;
PROC HC_PURPOSE
$ = 1;
PROC HC_DAY_NAME
numeric tmpDura;
recode  HC_COICOP   =>  tmpDura ;
    11100101-23109001   =>  1   ;   //  Food and drinks
    31109101    =>  4   ;   //  Pareu (printed or not)
    31109102    =>  4   ;   //  Clothing fabric, poplin material
    31109201    =>  4   ;   //  Pareu
    31109202    =>  4   ;   //  Material, Fabrics
    31109203    =>  4   ;   //  Poplin
    31209301    =>  4   ;   //  Men And Boys - Lavalava
    31209302    =>  4   ;   //  Men And Boys  - Shirt
    31209303    =>  4   ;   //  Men And Boys - Trousers
    31209304    =>  4   ;   //  Men And Boys  - T-Shirt
    31209305    =>  4   ;   //  Men And Boys - Underwear
    31209306    =>  4   ;   //  Men And Boys - School Uniform
    31209307    =>  4   ;   //  Men And Boys - Short
    31209308    =>  4   ;   //  Men And Boys - Socks
    31209309    =>  4   ;   //  Men And Boys - Other Clothes For Man.Boy Or Ned
    31209310    =>  4   ;   //  Men And Boys  - Clothes (general - not enough detail)
    31209401    =>  4   ;   //  Women And Girls - Shirt
    31209402    =>  4   ;   //  Women And Girls - Dress, Skirt
    31209403    =>  4   ;   //  Women And Girls - Trousers, Pants
    31209404    =>  4   ;   //  Women And Girls - T-Shirt, Top
    31209405    =>  4   ;   //  Women And Girls  - Suit
    31209406    =>  4   ;   //  Women And Girls - Underwear
    31209407    =>  4   ;   //  Women And Girls - School Uniform
    31209408    =>  4   ;   //  Women And Girls - Short
    31209409    =>  4   ;   //  Women And Girls - Socks
    31209410    =>  4   ;   //  Women And Girls - Other Clother For Women/Girl Or Ned
    31209411    =>  4   ;   //  Women And Girls - lava-lava
    31209412    =>  4   ;   //  Women And Girls - Clothes (general - not enough detail)
    31209501    =>  4   ;   //  Garments For Infant 0 To 2 Years Old
    31209601    =>  4   ;   //  Unspecifed / unisex clothes in general - shirts / tops
    31209602    =>  4   ;   //  Unspecifed / unisex clothes in general - pants, jeans,
    trousers
    31209603    =>  4   ;   //  Unspecifed / unisex clothes in general - dresses, slacks,
    and similar
    31209604    =>  4   ;   //  Unspecifed / unisex clothes in general - underwear, socks
    31209605    =>  4   ;   //  Unspecifed / unisex clothes in general - pajamas, robes
    31209606    =>  4   ;   //  Unspecifed / unisex clothes in general - school uniforms
    31209607    =>  4   ;   //  Unspecifed / unisex clothes in general - business, work attire
    31209608    =>  4   ;   //  Unspecifed / unisex clothes in general - Not enough detail
    31209609    =>  4   ;   //  Unspecifed / unisex clothes in general - Not specified
    elsewhere
    31209610    =>  4   ;   //  Unspecifed / unisex clothes in general - lava lava
    31309601    =>  4   ;   //  Cotton Gloves
    31309602    =>  4   ;   //  Tie
    31309603    =>  4   ;   //  Belt
    31309604    =>  4   ;   //  Cap
    31309605    =>  4   ;   //  Hat
    31309606    =>  4   ;   //  Other Clothing Accessories
    31309701    =>  4   ;   //  Cotton Thread
    31309702    =>  4   ;   //  Rayon Material
    31309703    =>  4   ;   //  Sewing Needles, Hand
    31309704    =>  4   ;   //  Other Sewing Materials
    31409801    =>  4   ;   //  Dyeing Of Garments
    31409901    =>  4   ;   //  Cleaning Of Garments, Laundry Shop
    31410001    =>  4   ;   //  Taylor
    31410101    =>  4   ;   //  Hire Of Garments
    32110201    =>  4   ;   //  Male - Beach Jandals
```

```
32110202    =>  4   ;   //  Male - Sport Shoes
32110203    =>  4   ;   //  Male - Casual Shoes
32110204    =>  4   ;   //  Male - Other Shoes For Men And Boys Or Ned
32110301    =>  4   ;   //  Female - Beach Jandals
32110302    =>  4   ;   //  Female - Sport Shoes
32110303    =>  4   ;   //  Female - Casual Shoes
32110304    =>  4   ;   //  Female - Other Shoes For Woman And Girl Or Ned
32110401    =>  4   ;   //  Shoes For Infant - 0 To 2 Years
32110501    =>  4   ;   //  !!DO NOT USE !!
32110502    =>  4   ;   //  Part Of Shoes (Heels, Soles, Lace…)
32110503    =>  4   ;   //  Other shoes related articles
32210601    =>  2   ;   //  Repair Of Footwear
32210701    =>  2   ;   //  Shoe-Cleaning Service
32210801    =>  2   ;   //  Hire of footwear
41110901    =>  2   ;   //  Rent Paid By Tenants For Main House
41211001    =>  2   ;   //  Rent Paid By Tenants For Land / Land Lease
41211101    =>  2   ;   //  Rent Paid By Tenants For Another House
42111201    =>  2   ;   //  Imputed Rentals Of Owner Occupying Their Main House
42211301    =>  2   ;   //  Imputed Rental For Secondary Residence
42211401    =>  2   ;   //  Imputed Rentals Of Households Paying A Reduced Rental Or
Housed Free
43111501    =>  4   ;   //  Pipe
43111502    =>  4   ;   //  Buteline
43111503    =>  4   ;   //  Elbow
43111504    =>  4   ;   //  Tap
43111505    =>  4   ;   //  Other Plumbing Materials
43111601    =>  4   ;   //  Ceramic Floor Tile
43111602    =>  4   ;   //  Floor Mosaic (Shower)
43111603    =>  4   ;   //  Other Surfacing Materials (Primaflex…)
43111701    =>  4   ;   //  Masking Tape For Painting
43111702    =>  4   ;   //  Paint Brush
43111703    =>  4   ;   //  Paint
43111704    =>  4   ;   //  Other Painting Materials
43111801    =>  4   ;   //  Blocks,Concrete Blocks
43111802    =>  4   ;   //  Cement
43111803    =>  4   ;   //  Door
43111804    =>  4   ;   //  Window, Louvre
43111805    =>  4   ;   //  Electric Cable inc network cable
43111806    =>  4   ;   //  Electric Materials (Plugs…)
43111807    =>  4   ;   //  Sink
43111808    =>  4   ;   //  Toilet Bowl, Sit, Shower
43111809    =>  4   ;   //  Plywood
43111810    =>  4   ;   //  Iron Roofing
43111811    =>  4   ;   //  Timber inc posts, planks, scotia, weatherboard, skirting
43111812    =>  4   ;   //  Hardboard
43111813    =>  4   ;   //  Inuslation foil, DPC
43111814    =>  4   ;   //  Tank iron
43111815    =>  4   ;   //  Rebar, reinforcing steel
43111816    =>  4   ;   //  Angle bar
43111817    =>  4   ;   //  Gib board
43111818    =>  4   ;   //  Cyclone straps
43111819    =>  4   ;   //  Ridge capping
43111820    =>  4   ;   //  Spouting, gutters
43111821    =>  4   ;   //  Other material for house maintenance ned
43211901    =>  2   ;   //  Contractor, Builder, Plumber, Electrician…
44112001    =>  2   ;   //  Water Bill, Water Meter
44112002    =>  2   ;   //  Water connection fees
44112003    =>  2   ;   //  Water truck
44112004    =>  2   ;   //  Other fee related to water supply
44212101    =>  2   ;   //  Garbage disposal
44312201    =>  2   ;   //  Sewer collection and disposal
44412301    =>  2   ;   //  Compound charge, co proprietor charge for caretaking of the
compound / building (cleaning….)
44412401    =>  2   ;   //  Security Services
45112501    =>  2   ;   //  Electricity Charge, Power Meter, Bill
45112502    =>  2   ;   //  Other Electricity Connection Fees
45212601    =>  2   ;   //  Do not use this commodity thank you - town gas and natural gas
45212701    =>  2   ;   //  Gas Cylinder
45212702    =>  1   ;   //  Gas Refill
45312801    =>  1   ;   //  Kerosene for domestic heating and lighting (fuel for vehicle
see 07.2.2_196.03)
```

```
45312802    =>  1   ;   //  Benzene
45312803    =>  1   ;   //  Petrol
45312804    =>  1   ;   //  Other Fuel For Generator Or Lights (Oil…)
45412901    =>  1   ;   //  Coconut Husk
45412902    =>  1   ;   //  Charcoal Imported
45412903    =>  1   ;   //  Other Solid Fuel (Firewood…)
45513001    =>  2   ;   //  Ice used for cooling and refrigeration purposes, ice cube,
ice blocks
51113101    =>  3   ;   //  Couche, Sofa
51113102    =>  3   ;   //  Other Lounge Furniture Or Ned
51113102    =>  3   ;   //  Bookshelf
51113201    =>  3   ;   //  Dining Table
51113202    =>  3   ;   //  Other Table
51113203    =>  3   ;   //  Chairs
51113204    =>  3   ;   //  Dining Hall Cabinet
51113205    =>  3   ;   //  Other Dining Hall Furniture Or Ned
51113301    =>  3   ;   //  Beds With Mattress
51113302    =>  3   ;   //  Beds Without Mattress
51113303    =>  3   ;   //  Mattress
51113304    =>  3   ;   //  Chest Of Drawers
51113305    =>  3   ;   //  Cupboards
51113306    =>  3   ;   //  Dressing Table
51113307    =>  3   ;   //  Wardrobe
51113308    =>  3   ;   //  Other Bedroom Furniture Or Ned
51113401    =>  3   ;   //  Foodsafe
51113402    =>  3   ;   //  Pantry Cupboard
51113403    =>  3   ;   //  Other Kitchen Furniture
51113501    =>  3   ;   //  Outdoor Chairs
51113502    =>  3   ;   //  Outdoor Table
51113503    =>  3   ;   //  Other Outdoor Furniture Or Ned
51113601    =>  3   ;   //  Cradles
51113602    =>  3   ;   //  High Chairs
51113603    =>  3   ;   //  Playpens
51113604    =>  3   ;   //  Other Baby Furniture Or Ned
51113701    =>  3   ;   //  Bathroom Cabinets
51113702    =>  3   ;   //  Other Bathroom Furniture Or Ned
51113801    =>  3   ;   //  Halogen Lamps
51113802    =>  3   ;   //  Other Lamp Or Ned (Exclude Torch)
51113901    =>  3   ;   //  Computer Table
51113902    =>  3   ;   //  Mirror
51113903    =>  3   ;   //  Bookcase
51113904    =>  3   ;   //  Decoration, Carving, Painting…
51113905    =>  3   ;   //  Other Furniture Or Ned
51214001    =>  3   ;   //  Carpet
51214002    =>  3   ;   //  Mat
51214003    =>  3   ;   //  Other Floor Cover
51314101    =>  2   ;   //  Repair Of Furniture
52114201    =>  4   ;   //  Awnings
52114202    =>  4   ;   //  Curtain, Curtain Material
52114203    =>  4   ;   //  Fabric Blinds
52114301    =>  4   ;   //  Bolsters
52114302    =>  4   ;   //  Futons
52114303    =>  4   ;   //  Hammocks
52114304    =>  4   ;   //  Pillows Including Cushions
52114401    =>  4   ;   //  Bed Sheets
52114402    =>  4   ;   //  Bed Spreads
52114403    =>  4   ;   //  Bedding Sets - Combination Eg Fitted, Flat And Pillow Cases
52114404    =>  4   ;   //  Blankets
52114405    =>  4   ;   //  Eiderdowns
52114406    =>  4   ;   //  Mosquito Nets
52114407    =>  4   ;   //  Pillow Cases
52114408    =>  4   ;   //  Other Bedding Or Ned
52114501    =>  4   ;   //  Table Clothes, Napkin
52114601    =>  4   ;   //  Face Towel
52114602    =>  4   ;   //  Hand Towel
52114603    =>  4   ;   //  Bath Towel
52114604    =>  4   ;   //  Beach Towel
52114605    =>  4   ;   //  Other Towel Or Ned
52114701    =>  4   ;   //  Shopping Plastic Bags
52114702    =>  4   ;   //  Shower Curtain
52114703    =>  4   ;   //  Covers For Clothes And Furniture
```

```
52114704    =>  4   ;   //  Doilies
52114705    =>  4   ;   //  Flags
52114706    =>  4   ;   //  Laundry Bags
52114707    =>  4   ;   //  Shoe Bags
52114708    =>  4   ;   //  Tea Towels
52114709    =>  4   ;   //  Other Household Textile
52114801    =>  2   ;   //  Repair Of Such Textiles
53114901    =>  3   ;   //  Stove, Kerosene, 2-Burner
53114902    =>  3   ;   //  Wicks, Kerosene Stove
53114903    =>  3   ;   //  Dishwasher
53114904    =>  3   ;   //  Freezer
53114905    =>  3   ;   //  Fridge-Freezer
53114906    =>  3   ;   //  Microwave
53114907    =>  3   ;   //  Oven Only
53114908    =>  3   ;   //  Refrigerator
53114909    =>  3   ;   //  Stove  – Electric
53114910    =>  3   ;   //  Stove – Gas
53114911    =>  3   ;   //  Stove - Kerosen
53114912    =>  3   ;   //  Wick For Kerosen Stove
53114913    =>  3   ;   //  Stove Ned
53114914    =>  3   ;   //  Other Major Kitchen Appliance Or Ned
53115001    =>  3   ;   //  Clothes Drier
53115002    =>  3   ;   //  Washing Machine
53115003    =>  3   ;   //  Other major laundry appliances
53115101    =>  3   ;   //  Air-Conditioner
53115102    =>  3   ;   //  Electric Water Heater (For Bathing)
53115103    =>  3   ;   //  Solar Power Water Heater
53115104    =>  3   ;   //  Extractor Hoods
53115105    =>  3   ;   //  Safes
53115106    =>  3   ;   //  Sewing Machine
53115107    =>  3   ;   //  Vacuum Cleaners, Steam Cleaning Machines, Carpet Shampooing
Machines And Machines For Scrubbing, Waxing And Polishing Floors
53115108    =>  3   ;   //  Other Major Household Appliances
53115109    =>  3   ;   //  Generator
53115110    =>  3   ;   //  Solar panel
53115111    =>  3   ;   //  Water tank
53215201    =>  4   ;   //  Jug, Electric, Steel
53215202    =>  4   ;   //  Fan For Ceiling
53215203    =>  4   ;   //  Fan For Table, Gie
53215204    =>  4   ;   //  Fan, Pedestal
53215205    =>  4   ;   //  Electric Toaster
53215206    =>  4   ;   //  Rice, Frc Rice Cooker
53215207    =>  4   ;   //  Blenders
53215208    =>  4   ;   //  Can-Openers
53215209    =>  4   ;   //  Coffee Maker
53215210    =>  4   ;   //  Deep Fryers
53215211    =>  4   ;   //  Food Processor
53215212    =>  4   ;   //  Frypans
53215213    =>  4   ;   //  Irons
53215214    =>  4   ;   //  Toaster
53215215    =>  4   ;   //  Other Small Electric Appliance
53315301    =>  2   ;   //  Repair Of Household Appliance
54015401    =>  4   ;   //  Ceremic Ware
54015402    =>  4   ;   //  China (Bone China, Royal Dalton, Lladro)
54015403    =>  4   ;   //  Crystal
54015404    =>  4   ;   //  Enamel
54015405    =>  4   ;   //  Glassware
54015406    =>  4   ;   //  Plastic Ware
54015407    =>  4   ;   //  Steel
54015408    =>  4   ;   //  Other Glassware Or Glassware Ned
54015501    =>  4   ;   //  Cutlery, Flatware And Silverware
54015601    =>  4   ;   //  Frying Pans
54015602    =>  4   ;   //  Hotplates
54015603    =>  4   ;   //  Household Scales
54015604    =>  4   ;   //  Kettle
54015605    =>  4   ;   //  Pots
54015606    =>  4   ;   //  Pressure Cookers
54015607    =>  4   ;   //  Saucepan
54015608    =>  4   ;   //  Tauaga (coconut strainer)
54015609    =>  4   ;   //  Other Non Electric Kitchen Ustensils Or Ned
54015701    =>  4   ;   //  Gas Lighter
```

```
54015702    =>  4   ;   //  Gas Lighter, Bic
54015703    =>  4   ;   //  Bottle Racks
54015704    =>  4   ;   //  Buckets
54015705    =>  4   ;   //  Containers For Bread, Coffee, Spices
54015706    =>  4   ;   //  Feeding Bottles
54015707    =>  4   ;   //  Iceboxes
54015708    =>  4   ;   //  Ironing Boards
54015709    =>  4   ;   //  Laundry Baskets
54015710    =>  4   ;   //  Letter Boxes
54015711    =>  4   ;   //  Portable Money Boxes
54015712    =>  4   ;   //  Thermos Flasks
54015713    =>  4   ;   //  Towel Rails
54015714    =>  4   ;   //  Waste Bins
54015715    =>  4   ;   //  Waste Paper Baskets
54015716    =>  4   ;   //  Scales, Counter 20Kg
54015717    =>  4   ;   //  Hand Basin With Brackets (For Toilets)
54015718    =>  4   ;   //  Jug, Plastic,
54015719    =>  4   ;   //  Plastic Jerry Cans, 10 Litres
54015720    =>  4   ;   //  Tarpaulin,
54015721    =>  4   ;   //  Tarpaulin Clear (Heavy Plastic)
54015722    =>  4   ;   //  Other Non Electric Hh Article Or Ned
54015801    =>  2   ;   //  Repair Of Household Article
55115901    =>  3   ;   //  Cement Mixer Petrol
55115902    =>  3   ;   //  Water Pump Electric Orange Sj400.
55115903    =>  3   ;   //  Compression
55115904    =>  3   ;   //  Engine Cleaning Gun
55115905    =>  3   ;   //  V Belts ( For Concrete Mixers)
55115906    =>  3   ;   //  Brush Cutters
55115907    =>  3   ;   //  Chainsaws
55115908    =>  3   ;   //  Cultivators
55115909    =>  3   ;   //  Electric Drills
55115910    =>  3   ;   //  Garden Tractors
55115911    =>  3   ;   //  Lawnmowers
55115912    =>  3   ;   //  Motor Mowers
55115913    =>  3   ;   //  Sanders And Hedge Cutters
55115914    =>  3   ;   //  Other Motorized Tools And Equipment
55116001    =>  2   ;   //  Repair Of Such Articles
55216101    =>  3   ;   //  Dustmask
55216102    =>  3   ;   //  Hacksaws
55216103    =>  3   ;   //  Hand Saw
55216104    =>  3   ;   //  Petrol Pump
55216105    =>  3   ;   //  Circular Saw
55216106    =>  3   ;   //  Clamp Tool
55216107    =>  3   ;   //  Air Duster Gun
55216108    =>  3   ;   //  Coach Screws
55216109    =>  3   ;   //  Dyna Bolts
55216110    =>  3   ;   //  Crowbar
55216111    =>  3   ;   //  Cut Off Wheel 230X2.5X22Mm
55216112    =>  3   ;   //  Hacksaw
55216113    =>  3   ;   //  Matic Guns, Standard # Skeleton
55216114    =>  3   ;   //  Multimeter Pocket Digital
55216115    =>  3   ;   //  Sandpaper
55216116    =>  3   ;   //  Spray Gun Suction
55216117    =>  3   ;   //  Trowel
55216118    =>  3   ;   //  Wire Brush
55216119    =>  3   ;   //  Files
55216120    =>  3   ;   //  Hammers
55216121    =>  3   ;   //  Pliers
55216122    =>  3   ;   //  Rasps
55216123    =>  3   ;   //  Saws
55216124    =>  3   ;   //  Screwdrivers
55216125    =>  3   ;   //  Spanners
55216126    =>  3   ;   //  Trimming Knives
55216127    =>  3   ;   //  Wrenches
55216128    =>  3   ;   //  Other Hand Tools
55216201    =>  3   ;   //  Hose 1With Fittings
55216202    =>  3   ;   //  Cane Knives, Bush Knives
55216203    =>  3   ;   //  Forks
55216204    =>  3   ;   //  Hoses And Sprinklers
55216205    =>  3   ;   //  Rakes
55216206    =>  3   ;   //  Scythes
```

```
55216207 => 3 ; // Shovels
55216208 => 3 ; // Sickles
55216209 => 3 ; // Spades
55216210 => 3 ; // Watering Cans
55216211 => 3 ; // Wheelbarrows
55216212 => 3 ; // Other Garden Tools
55216301 => 3 ; // Ladders And Steps
55216401 => 3 ; // Entrance Door Lock Set
55216402 => 3 ; // Hasp & Staple
55216403 => 3 ; // Security Door Locks C/W Knob/Handles
55216404 => 3 ; // Security Entrance Door Locks/Knob Handles
55216405 => 3 ; // Toilet Door Lock Set
55216406 => 3 ; // Brass Padlocks
55216407 => 3 ; // Pad Bolt
55216408 => 3 ; // Padlock
55216409 => 3 ; // Handles
55216410 => 3 ; // Hinges
55216411 => 3 ; // Security mesh
55216412 => 3 ; // Other security materials ned
55216501 => 3 ; // Shower Curtain Rods
55216502 => 3 ; // Toilet Paper Holder
55216503 => 3 ; // Toilet Seat Covers
55216504 => 3 ; // Carpet Rods
55216505 => 3 ; // Curtain Rails
55216506 => 3 ; // Hooks
55216507 => 3 ; // Other Metal Article For The House
55216601 => 3 ; // Chains
55216602 => 3 ; // Grids
55216603 => 3 ; // Hoop Segments For Fencing And Bordering
55216604 => 3 ; // Stakes
55216605 => 3 ; // Tie wire (included chicken wire)
55216606 => 3 ; // Mesh, meshwire
55216607 => 3 ; // Other Metal Article For The Garden
55216701 => 3 ; // Battery (Sk-N100)
55216702 => 3 ; // Battery (Sk-N70Zl)
55216703 => 3 ; // Battery, Alkaline
55216704 => 3 ; // Battery, Energizer
55216705 => 3 ; // Battery, Eveready
55216706 => 3 ; // Battery, Magnacell
55216707 => 3 ; // Battery, Panasonic
55216708 => 3 ; // Battery Ned
55216709 => 3 ; // Energy Saver Lights
55216710 => 3 ; // Extension Cords
55216711 => 3 ; // Fluorescent Light
55216712 => 3 ; // Fluorescent Tube
55216713 => 3 ; // Light Bulb
55216714 => 3 ; // Torches Powered Diving Waterproof
55216715 => 3 ; // Spark Plugs
55216716 => 3 ; // Alarms
55216717 => 3 ; // Bells
55216718 => 3 ; // Flashlights
55216719 => 3 ; // Fluorescent Tubes starters
55216720 => 3 ; // Hand Lamps
55216721 => 3 ; // Power Sockets
55216722 => 3 ; // Switches
55216723 => 3 ; // Torches
55216724 => 3 ; // Wiring Flex
55216725 => 3 ; // Other Small Electric Appliance
55216801 => 2 ; // Repair Of Such Articles
56116901 => 1 ; // Air Freshener
56116902 => 1 ; // Dish Washing Liquid
56116903 => 1 ; // Dish Washing Liquid, Joy
56116904 => 1 ; // Disinfectant, Dettol
56116905 => 1 ; // Disinfectant: Pine-O-Cleen
56116906 => 1 ; // Fly Spray Surface, Mortein
56116907 => 1 ; // Fly Spray, Red, Mortein
56116908 => 1 ; // Insect Repellent, Aeroguard
56116909 => 1 ; // Rat Trap
56116910 => 1 ; // Soap, Laundry, Pacific
56116911 => 1 ; // Soap, Laundry, Sudso
56116912 => 1 ; // Toilet Cleaner, Harpic
```

```
56116914    =>  1   ;   //  Washing Powder, Boom
56116915    =>  1   ;   //  Washing Powder, Gain
56116916    =>  1   ;   //  Washing Powder, Lerosa
56116917    =>  1   ;   //  Washing Powder, Surf
56116918    =>  1   ;   //  Washing Powder, Tasi
56116919    =>  1   ;   //  Cleaning Paste Axion
56116920    =>  1   ;   //  Cold Power
56116921    =>  1   ;   //  Fabric Softener, Downy
56116922    =>  1   ;   //  Moth Balls
56116923    =>  1   ;   //  Napisan Powder
56116924    =>  1   ;   //  Bleach, Janola
56116925    =>  1   ;   //  Clorox
56116926    =>  1   ;   //  Clorox, Janola,
56116927    =>  1   ;   //  Vaselin
56116928    =>  1   ;   //  Other Cleaning And Maintenance Products
56117001    =>  1   ;   //  Sponge
56117002    =>  1   ;   //  Sponge Mop With Handle
56117003    =>  1   ;   //  Dustpan And Brush (2 Piece)
56117004    =>  1   ;   //  Mop Complete
56117005    =>  1   ;   //  Mops & Buckets
56117006    =>  1   ;   //  Toilet Cleaning Brushes C/W Stand
56117008    =>  1   ;   //  Brooms
56117009    =>  1   ;   //  Duster
56117010    =>  1   ;   //  Wired steelo
56117010    =>  1   ;   //  Other Articles For Cleaning
56117101    =>  1   ;   //  Aluminium Foil
56117102    =>  1   ;   //  Foam Cups
56117103    =>  1   ;   //  Foam Plates
56117104    =>  1   ;   //  Napkins, Paper
56117105    =>  1   ;   //  Plastic Cups
56117106    =>  1   ;   //  Plastic Dinner Plates
56117107    =>  1   ;   //  Plastic Spoons
56117108    =>  1   ;   //  Table Cloth, Plastic,
56117109    =>  1   ;   //  Filter
56117110    =>  1   ;   //  Vaccum Cleaner Bags
56117111    =>  1   ;   //  Kitchen Paper, Tissue, Paper Towel
56117112    =>  1   ;   //  Plastic Bin Liner
56117113    =>  1   ;   //  Other Paper Product
56117201    =>  1   ;   //  Candle
56117202    =>  1   ;   //  Adhesive Tapes
56117203    =>  1   ;   //  Camphor
56117204    =>  1   ;   //  String rope
56117205    =>  1   ;   //  Clothes□Hangers
56117206    =>  1   ;   //  Fire Extinguishers For Households
56117207    =>  1   ;   //  Glues
56117208    =>  1   ;   //  Incense Sticks
56117209    =>  1   ;   //  Knitting Needles
56117210    =>  1   ;   //  Lamp Wicks
56117211    =>  1   ;   //  Anchor Bolts M12 X 300
56117212    =>  1   ;   //  Rivet
56117213    =>  1   ;   //  Nail plate
56117214    =>  1   ;   //  Screws
56117215    =>  1   ;   //  Nail
56117216    =>  1   ;   //  Rubber gloves
56117222    =>  1   ;   //  PVC glue
56117224    =>  1   ;   //  Hinges
56117225    =>  1   ;   //  Matches
56117226    =>  1   ;   //  Measuring Tape
56117227    =>  1   ;   //  Mosquito Coil
56117228    =>  1   ;   //  Mosquito Coil, Zap
56117237    =>  1   ;   //  Packaging Tape
56117241    =>  1   ;   //  Cloth Pegs Hard Plastic
56117244    =>  1   ;   //  Other Non Durable Articles
56217301    =>  2   ;   //  Butlers
56217302    =>  2   ;   //  Cooks
56217303    =>  2   ;   //  Drivers
56217304    =>  2   ;   //  Gardeners
56217305    =>  2   ;   //  Governesses
56217306    =>  2   ;   //  Maids
56217307    =>  2   ;   //  Secretaries
56217308    =>  2   ;   //  Tutors
```

```
56217309    =>  2   ;   //  Baby sitting / child minding services provided at home
56217310    =>  2   ;   //  Elderly care services provided at home
56217398    =>  2   ;   //  Other Domestic Services - not elsewhere classified (nec)
56217399    =>  2   ;   //  Other Domestic Services - not enough detail (ned)
56217401    =>  2   ;   //  Domestic services, including baby-sitting and housework,
supplied by enterprises or self-employed persons
56217501    =>  2   ;   //  Pest control, fumigation
56217502    =>  2   ;   //  Lawn mowing services
56217503    =>  2   ;   //  Other household services provided by external company to the
household
56217601    =>  2   ;   //  Dry-cleaning, laundering and dyeing of household linen,
household textiles carpets
56217701    =>  2   ;   //  Hire of furniture, equipment (e.g. hire of cement mixer…)
61117801    =>  1   ;   //  Prescription medicine
61117901    =>  1   ;   //  Tablets, Panadol
61117902    =>  1   ;   //  Vicks Vaporub
61117903    =>  1   ;   //  Micreme
61117904    =>  1   ;   //  Panadol
61117905    =>  1   ;   //  Other Patent Medecine
61118001    =>  1   ;   //  Cod Liver Oil
61118002    =>  1   ;   //  Halibut Liver Oil
61118003    =>  1   ;   //  Oral Contraceptives
61118004    =>  1   ;   //  Serums And Vaccines
61118005    =>  1   ;   //  Vitamins And Minerals
61118006    =>  1   ;   //  Other Pharmaceutical Products Or Ned
61218101    =>  1   ;   //  Band Aids, Jj
61218102    =>  1   ;   //  Bandages
61218103    =>  1   ;   //  Clinical Thermometers
61218104    =>  1   ;   //  Condoms
61218105    =>  1   ;   //  First□Aid Kits
61218106    =>  1   ;   //  Hot Water Bottles
61218107    =>  1   ;   //  Hypodermic Syringes
61218108    =>  1   ;   //  Ice Bags
61218109    =>  1   ;   //  Mechanical Contraceptive Devises
61218110    =>  1   ;   //  Medical Hosiery Items
61218111    =>  1   ;   //  Pregnancy Tests
61218112    =>  1   ;   //  Other Medical Product
61318201    =>  3   ;   //  Artificial Limbs And Other Prosthetic Devices
61318202    =>  3   ;   //  Corrective Eye Glasses And Contact Lenses
61318203    =>  3   ;   //  Crutches
61318204    =>  3   ;   //  Dentures Excluding Fitting Costs
61318205    =>  3   ;   //  Devices For Monitoring Blood Pressure
61318206    =>  3   ;   //  Glass Eyes
61318207    =>  3   ;   //  Health Lamps
61318208    =>  3   ;   //  Hearing Aids
61318209    =>  3   ;   //  Medical Massage Equipment
61318210    =>  3   ;   //  Neck Braces
61318211    =>  3   ;   //  Orthopaedic Braces And Supports
61318212    =>  3   ;   //  Orthopaedic Footwear
61318213    =>  3   ;   //  'Special Beds'
61318214    =>  3   ;   //  Surgical Belts, Trusses And Supports
61318215    =>  3   ;   //  Wheelchairs – Powered And Unpowered And Invalid Carriers
61318216    =>  3   ;   //  Other Therapeutic Appliances
61318301    =>  2   ;   //  Repair or hire of therapeutic appliances
62118401    =>  2   ;   //  Consultations Of Doctor, Specialist
62218501    =>  2   ;   //  Consultations Of Dentist
62318601    =>  2   ;   //  Blood Test, Xray, Analysis
62318602    =>  2   ;   //  Nurse And Midwife
62318603    =>  2   ;   //  Other Paramedical Doctor
62318604    =>  2   ;   //  Other paramedical services (chiropractor…)
62318701    =>  2   ;   //  Traditional Healers
63018801    =>  2   ;   //  Hosptial, Clinic Fee
63018901    =>  2   ;   //  Medical service eg services of physicians
63019001    =>  2   ;   //  Paramedical services eg services of chiropractors,
optometrists
71119101    =>  3   ;   //  Car
71119102    =>  3   ;   //  Purchase of pick up truck, utility vehicle
71119103    =>  3   ;   //  Purchase of truck, mini bus, van
71219201    =>  3   ;   //  Motor Cycle
71219202    =>  3   ;   //  Scooter
71319301    =>  3   ;   //  Bicycle
```

```
71419401    =>  3   ;   //  Animal drawn vehicle
72119501    =>  4   ;   //  Aluminium Round Tubes For Boat
72119502    =>  4   ;   //  Fan Belt
72119503    =>  4   ;   //  Tyre
72119504    =>  4   ;   //  Tyre Tubes
72119505    =>  4   ;   //  Batteries For Vehicle
72119506    =>  4   ;   //  Tyre inflator
72119507    =>  4   ;   //  Other Spare Part For Vehicle And Boat
72219601    =>  1   ;   //  Fuel / gas for cars and land vehicle (private use)
72219602    =>  1   ;   //  Fuel / gas for boats and other water crafts (private use)
72219603    =>  1   ;   //  Fuel / gas for other (chainsaw, tillers, tractor…)
72219604    =>  1   ;   //  Fuel / gas without any details
72219605    =>  1   ;   //  Other fuel / gas for vehicle n e s
72219606    =>  1   ;   //
72219607    =>  1   ;   //
72219608    =>  1   ;   //
72219701    =>  1   ;   //  Oil Outboard
72219702    =>  1   ;   //  Oil Outboard Super
72219703    =>  1   ;   //  Oil Outboard Super, Ob
72219704    =>  1   ;   //  Oil Sae15/40
72219705    =>  1   ;   //  Oil Sae30
72219706    =>  1   ;   //  Oil, Air Compresser
72219707    =>  1   ;   //  Oil, Hydraulic
72219708    =>  1   ;   //  Marine Spray 4 Crc/Wd40
72219709    =>  1   ;   //  Gear Oil
72219710    =>  1   ;   //  Grease Cartridges, Multipurpose
72219711    =>  1   ;   //  Oil Lubricant Ned
72319801    =>  2   ;   //  Service For Maintenance Of Car/Boat, Mechanic
72419901    =>  2   ;   //  Driving lesson
72419902    =>  2   ;   //  Driving license fee
72419903    =>  2   ;   //  Registration fee
72419904    =>  2   ;   //  Other fee in respect of vehicle
72420001    =>  2   ;   //  Car Rental
72420002    =>  2   ;   //  Truck rental, hire
72420003    =>  2   ;   //  Mini bus, van rental, hire
72420004    =>  2   ;   //  Motorcycle / scooter rental, hire
72420005    =>  2   ;   //  Other vehicle rental n e s
73220101    =>  2   ;   //  Bus Fare
73220102    =>  2   ;   //  Taxi Fare
73220103    =>  2   ;   //  Other Transport Fare Ned
73320201    =>  2   ;   //  Air Fare
73320202    =>  2   ;   //  Air fare - domestic
73320203    =>  2   ;   //  Air fare without any detail
73420301    =>  2   ;   //  Boat Fare
73420302    =>  2   ;   //  Boat fare (travel overseas)
73420303    =>  2   ;   //  Other fees related to transport by boat n e s
73520401    =>  2   ;   //  Air fare and boat fare together
73620501    =>  2   ;   //  Freight, Delivery Charge
73620502    =>  2   ;   //  Other services related to transport
81120601    =>  2   ;   //  Payments For Delivery Of Letter, Postcards And Parcels
81120602    =>  2   ;   //  Private Mail And Parcel Delivery
81120603    =>  2   ;   //  Purchases Of Postage Stamps
81120604    =>  2   ;   //  Purchase Of Pre□Franked Postcards And Aerogramme
81120605    =>  2   ;   //  Other Postal Fee
82020701    =>  3   ;   //  Purcahse Of Cell (Mobile) Phones, Smartphone
82020702    =>  3   ;   //  Purchase Of Land □Line Telephones
82020703    =>  3   ;   //  Radio Telephones
82020704    =>  3   ;   //  Telefax Machines
82020705    =>  3   ;   //  Other Telephone And Telefax Equipment
82020801    =>  2   ;   //  Repair Of Such Equipment
83020901    =>  2   ;   //  Connection And Subscription Costs Of Cell (Mobile) Phones
83020902    =>  2   ;   //  Purchase Of Recharge Cards For Cell (Mobile) Phones
83020903    =>  2   ;   //  Purchase Of Sim Card
83020904    =>  2   ;   //  Installation And Subscription Costs Of Land□Line Telephone
Equipment
83020905    =>  2   ;   //  Internet Services
83020906    =>  2   ;   //  Other Telephone Services
91121001    =>  3   ;   //  Car Radios
91121002    =>  3   ;   //  Cd-Players
91121003    =>  3   ;   //  Microphones And Earphones
91121004    =>  3   ;   //  Radio Clocks
```

```
91121005    =>  3   ;   //  Radio Sets
91121006    =>  3   ;   //  Stereos And Their Constituent Units Eg Tuners, Amplifiers Etc
91121007    =>  3   ;   //  Television Aerials Of All Types
91121008    =>  3   ;   //  Television Sets
91121009    =>  3   ;   //  Two-Way Radios
91121010    =>  3   ;   //  Video Player/Vcd/Dvd
91121011    =>  3   ;   //  Mp3 Players, Ipods
91121012    =>  3   ;   //  Other Equipment
91221101    =>  3   ;   //  Cameras
91221102    =>  3   ;   //  Video camera
91221201    =>  3   ;   //  Film And Slide Projectors
91221301    =>  3   ;   //  Accessories for camera (flash, objective, lenses…)
91221401    =>  3   ;   //  Binoculars
91221402    =>  3   ;   //  Compass
91221403    =>  3   ;   //  Other Optical Instrument
91221404    =>  3   ;   //  Telescopes
91221405    =>  3   ;   //  Gps devices
91321501    =>  3   ;   //  Applications
91321502    =>  3   ;   //  Calculators
91321503    =>  3   ;   //  Computer Software Packages, E.G. Operating Systems
91321504    =>  3   ;   //  Personal Computers Desktop
91321505    =>  3   ;   //  Personal Computers Laptop
91321506    =>  3   ;   //  Printers And Miscellaneous Accessories Accompanying Them
91321507    =>  3   ;   //  Tablet
91321508    =>  3   ;   //  Other It Equipment
91421601    =>  4   ;   //  Dvd, Blue Ray
91421602    =>  4   ;   //  Cd Compact Disc
91421603    =>  4   ;   //  Other Recording Media
91421604    =>  4   ;   //  portable harddrives, flash drives
91521701    =>  2   ;   //  Repair Of Audio-Visual, Photographic And Information
Processing Equipment
92121801    =>  3   ;   //  Boat without outboard motor
92121802    =>  3   ;   //  Canoe
92121803    =>  3   ;   //  Outboard Motor
92121805    =>  3   ;   //  Dive gear (tanks, bc, regulators, etc.)
92121807    =>  3   ;   //  Water craft trailer
92121808    =>  3   ;   //  Other major durables for outdoor recreation
92121809    =>  3   ;   //  Boat with outboard motor
92221901    =>  3   ;   //  Guitare
92221902    =>  3   ;   //  Guitar Strings, Steel
92221903    =>  3   ;   //  Other Musical Instrument
92221904    =>  3   ;   //  Other musical instrument n e s
92222001    =>  3   ;   //  Billiard tables
92222002    =>  3   ;   //  Gaming machines
92222003    =>  3   ;   //  Pinball machines
92222004    =>  3   ;   //  Pin-pong table (table-tennis table)
92322101    =>  2   ;   //  Maintenance And Repair Of Other Major Durables For
Recreation And Culture
93122201    =>  4   ;   //  Card Games
93122202    =>  4   ;   //  Chess
93122203    =>  4   ;   //  Parlour Games
93122204    =>  4   ;   //  Video Game Console And Accessories
93122205    =>  4   ;   //  Other Games
93122301    =>  4   ;   //  Barbie Dolls
93122302    =>  4   ;   //  Cars (Police) Remote Control
93122303    =>  4   ;   //  Toys of all kinds (dolls, fireworks, toy cars, etc.)
93122304    =>  4   ;   //  Water Guns
93122305    =>  4   ;   //  X-Mas Lights
93122306    =>  4   ;   //  X-Mas Tree Decoration
93122307    =>  4   ;   //  Baby Dolls With Accessories
93122308    =>  4   ;   //  Other Toys
93122401    =>  4   ;   //  Other items for collection e.g. coins, medals, zoological
and botanical specimens etc.)
93122402    =>  4   ;   //  Stamp collecting requisites (used or cancelled postage
stamps, stamp albums etc.)
93222501    =>  4   ;   //  Beach Volley Ball
93222502    =>  4   ;   //  Cricket Ball
93222503    =>  4   ;   //  Other Sport Equipment
93222504    =>  4   ;   //  Golf clubs
93222505    =>  4   ;   //  Nets
93222506    =>  4   ;   //  Rackets
```

```
93222507    =>  4   ;   //  Shuttlecocks
93222508    =>  4   ;   //  Weights
93222509    =>  4   ;   //  Other equipment for sport n e s
93222601    =>  4   ;   //  Parachute and other sky-diving equipment
93222701    =>  4   ;   //  Firearms (gun, bullet…)
93222801    =>  4   ;   //  Fishing Lines
93222802    =>  4   ;   //  Speargun
93222803    =>  4   ;   //  Mask, Flippers
93222804    =>  4   ;   //  Other Fishing Equipments
93222901    =>  4   ;   //  Frisbee
93222902    =>  4   ;   //  Inflatable Boats
93222903    =>  4   ;   //  Inflatable Swimming Pools
93222904    =>  4   ;   //  Volleyball
93222905    =>  4   ;   //  Other Beach And Open Air Games
93223001    =>  4   ;   //  Air Mattresses And Inflating Pumps
93223002    =>  4   ;   //  Camping Stoves And Barbecues
93223003    =>  4   ;   //  Sleeping Bags
93223004    =>  4   ;   //  Tents And Accessories
93223101    =>  4   ;   //  Football Boots
93223102    =>  4   ;   //  Golfing Shoes
93223103    =>  4   ;   //  Other Such Footwear Fitted With Ice□Skates, Rollers, Spikes,
Studs, Etc.
93223104    =>  4   ;   //  Rugby Boots
93223201    =>  4   ;   //  Belts
93223202    =>  4   ;   //  Body Padding
93223203    =>  4   ;   //  Boxing Gloves
93223204    =>  4   ;   //  Goggles
93223205    =>  4   ;   //  Headgear
93223206    =>  4   ;   //  Life Jackets
93223207    =>  4   ;   //  Shin Guards
93223208    =>  4   ;   //  Supports
93223301    =>  2   ;   //  Repair Of Such Articles
93323401    =>  1   ;   //  Fertilisers, Composts And Garden Peat
93323402    =>  1   ;   //  Horticultural Preparations
93323403    =>  1   ;   //  Other Garden Plants And Flower Stuff
93323404    =>  1   ;   //  Pots and plant holders
93323405    =>  1   ;   //  Other plants related items n e s
93423501    =>  1   ;   //  Purchase of pets (cat, dog,….)
93423601    =>  1   ;   //  Animal Feed
93423602    =>  1   ;   //  Pig feed, pig pellet, grower
93423603    =>  1   ;   //  Chicken feed
93423604    =>  1   ;   //  Other animal feed (including expired food purchased for
animal feed)
93423605    =>  1   ;   //  Other pet-related products (feed, leashes, vet fee, etc.)
93523701    =>  2   ;   //  Veterinary
93523801    =>  2   ;   //  Other Services Eg Training
94123901    =>  2   ;   //  Bowling Alleys
94123902    =>  2   ;   //  Fairgrounds And Amusement Parks
94123903    =>  2   ;   //  Fitness Centres
94123904    =>  2   ;   //  Golf Courses
94123905    =>  2   ;   //  Gymnasia
94123906    =>  2   ;   //  Horse Racing Courses
94123907    =>  2   ;   //  Motor Racing Circuits
94123908    =>  2   ;   //  Pin□Ball Machines (Excluding Game Of Chance)
94123909    =>  2   ;   //  Roundabouts, See□Saws And Other Playground
94123910    =>  2   ;   //  Facilities For Children
94123911    =>  2   ;   //  See□Saws And Playground Facilities For Children
94123912    =>  2   ;   //  Skating Rinks
94123913    =>  2   ;   //  Sports Stadium
94123914    =>  2   ;   //  Squash Courts
94123915    =>  2   ;   //  Swimming Pools
94123916    =>  2   ;   //  Tennis Courts
94123917    =>  2   ;   //  Velodromes
94124001    =>  2   ;   //  Hire of boat
94124002    =>  2   ;   //  Hire of other accessories for sport and recreation
94124101    =>  2   ;   //  Aerobics
94124102    =>  2   ;   //  Chess
94124103    =>  2   ;   //  Dancing
94124104    =>  2   ;   //  Music
94124105    =>  2   ;   //  Swimming
94124106    =>  2   ;   //  Other Group Lessons
```

```
94124201    =>  2   ;   //  Services Of Mountain Guides, Tour Guides Etc
94124301    =>  2   ;   //  Navigational Aid Services For Boating
94124401    =>  2   ;   //  Hire of game-specific footwear (excludes purchase (932))
94224501    =>  2   ;   //  Art Galleries And Exhibitions
94224502    =>  2   ;   //  Cinemas
94224503    =>  2   ;   //  Concert And Music Halls
94224504    =>  2   ;   //  Historical Monuments, National Parks, Zoological And
Botanical Gardens
94224505    =>  2   ;   //  And Aquaria
94224506    =>  2   ;   //  Libraries
94224507    =>  2   ;   //  Theatre
94224508    =>  2   ;   //  Cinema
94224509    =>  2   ;   //  Game At The Stadium
94224510    =>  2   ;   //  Other Services
94224601    =>  2   ;   //  Dvd rental, hire
94224701    =>  2   ;   //  Cable Tv Fee
94224702    =>  2   ;   //  Other Fee Related To Cable Tv
94224801    =>  2   ;   //  Services Of Photographers Eg Film Developing , Print
Processing, Enlarging, Portrait Photography, Wedding Photography
94324901    =>  2   ;   //  Bingo Card
94324902    =>  2   ;   //  Gambling Machine
94324903    =>  2   ;   //  Other Gamble Fee
95125001    =>  4   ;   //  Atlases
95125002    =>  4   ;   //  Dictionaries
95125003    =>  4   ;   //  Encyclopaedias
95125004    =>  4   ;   //  Guidebooks
95125005    =>  4   ;   //  Musical Scores
95125006    =>  4   ;   //  Novels, Story Books
95125007    =>  4   ;   //  Scrapbooks And Albums For Children
95125008    =>  4   ;   //  Textbooks, School Books
95125009    =>  4   ;   //  Other Book Related Expense
95125101    =>  2   ;   //  Bookbinding
95225201    =>  1   ;   //  Magazines
95225202    =>  1   ;   //  Newspapers
95225203    =>  1   ;   //  Periodicals
95325301    =>  1   ;   //  Greeting Cards, Visiting Card
95325302    =>  1   ;   //  Maps And Globes
95325303    =>  1   ;   //  Poster, Post Card…
95425401    =>  1   ;   //  Envelopes
95425402    =>  1   ;   //  Marking Pen
95425403    =>  1   ;   //  Pencil Case
95425404    =>  1   ;   //  Pencil Sharpener
95425405    =>  1   ;   //  Pens, Blue/Black, Ball Point
95425406    =>  1   ;   //  Pens, Blue/Blk/Red, Metal Point, Bic
95425407    =>  1   ;   //  Rubber (Eraser)
95425408    =>  1   ;   //  Carpenter’S Pencil
95425409    =>  1   ;   //  Carpenter'S Folding Ruler
95425410    =>  1   ;   //  Chalk Line Refill Blue Johnson
95425411    =>  1   ;   //  Crepe Paper
95425412    =>  1   ;   //  Dual Metric/Imperial 8/10M
95425413    =>  1   ;   //  Gift Wrapper
95425414    =>  1   ;   //  Sellotape, 2" Clear
95425415    =>  1   ;   //  Tialeniu School Badges-Jan. 2011
95425416    =>  1   ;   //  Other Stationery Or Ned
95425501    =>  1   ;   //  Ink Cartridge Brother
95425601    =>  1   ;   //  Exercise book
96025701    =>  2   ;   //  All Inclusive Holidays Or Tours Which Provide For Travel,
Food, Accommodation, Guides Etc
101025801   =>  2   ;   //  Kindergarden Fees
101025802   =>  2   ;   //  Primary School Fees
102025901   =>  2   ;   //  Secondary School Fees
103026001   =>  2   ;   //  Technical Education School Fees
103026002   =>  2   ;   //  Vocational Education School Fees
104026101   =>  2   ;   //  University Fees
105026201   =>  2   ;   //  Other Educational Program Fees
105026202   =>  2   ;   //  Tutoring for academic subjects (science….), private lesson
for school purposes
105026203   =>  2   ;   //  Other expenditure related to education (teacher day, sport
day, school events…)
111126301   =>  2   ;   //  Restaurant, Café
111126302   =>  2   ;   //  Bar Night Clubs
```

```
111126401   =>  2   ;   //  Ham, Sandwich, Local
111126402   =>  2   ;   //  Ham, Sandwich, Us
111126403   =>  2   ;   //  Other Take Away Food
111126501   =>  2   ;   //  Take away ice cream
111126502   =>  2   ;   //  Take away pizza
111126503   =>  2   ;   //  Cooked poultry, chicken take away
111126504   =>  2   ;   //  Fish and chips take away
111126505   =>  2   ;   //  Burger, cheese burger take away
111126506   =>  2   ;   //  Fried chicken take away
111126507   =>  2   ;   //  Meat pies take away
111126508   =>  2   ;   //  Sandwich take away
111126598   =>  2   ;   //  Other take away plate, take away food (chinese food,
chicken, barbecue, cassava, fish, sashimi ….)
111126599   =>  2   ;   //  Take away food without any detail
111126601   =>  2   ;   //  The sale of cooked dishes by catering contractors whether
collected by the customer or delivered to the customer’s home
111226701   =>  2   ;   //  School restaurant
112026801   =>  2   ;   //  Hotel, Motel, Guest House
112026802   =>  2   ;   //  Packaged accomidation with food (hotel)
112026901   =>  2   ;   //  Boarding Schools, Univ Compound
112027001   =>  2   ;   //  Accommodation services provided by others
121127101   =>  2   ;   //  Hairdresser
121127201   =>  2   ;   //  Body Care, Depilation (Waxing, Threading, Etc.) And The Like
121127202   =>  2   ;   //  Facials
121127203   =>  2   ;   //  Manicure
121127204   =>  2   ;   //  Non□Medical Massage
121127205   =>  2   ;   //  Pedicure
121127206   =>  2   ;   //  Turkish Baths, Saunas
121127207   =>  2   ;   //  Other Personal Grooming
121127208   =>  2   ;   //  Tattoo, piercings and other body art
121227301   =>  4   ;   //  Curling Tongs And Styling Combs
121227302   =>  4   ;   //  Electric Razors And Hair Trimmers
121227303   =>  4   ;   //  Electric Toothbrushes
121227304   =>  4   ;   //  Hand□Held And Hood Hair Dryers
121227305   =>  4   ;   //  Sunlamps
121227306   =>  4   ;   //  Other Electric Appliance For Personal Care
121227401   =>  2   ;   //  Repair Of Such Appliances
121327501   =>  1   ;   //  Razor Blades
121327502   =>  1   ;   //  Razor Refill, Gillette
121327503   =>  1   ;   //  Razor, Contour
121327504   =>  1   ;   //  Tooth Brush
121327505   =>  1   ;   //  Hair Brush
121327506   =>  1   ;   //  Nail Brush
121327507   =>  1   ;   //  Nail Files
121327508   =>  1   ;   //  Personal Scale, Baby Scale
121327509   =>  1   ;   //  Combs
121327510   =>  1   ;   //  Other Non Electric Appliance For Personal Care
121327601   =>  1   ;   //  Shampoo, Baby Shampoo
121327602   =>  1   ;   //  Hair Conditioner, Palmolive
121327603   =>  1   ;   //  Hair Gel, (Mousse)
121327604   =>  1   ;   //  Hair Shampoo, Palmolive
121327605   =>  1   ;   //  Shaving Cream, Nivea
121327606   =>  1   ;   //  Soap, Lux
121327607   =>  1   ;   //  Soap, Palmolive
121327608   =>  1   ;   //  Soap, Protex
121327609   =>  1   ;   //  Body Wash
121327610   =>  1   ;   //  Body Wash, Lux
121327611   =>  1   ;   //  Body Wash, Protex
121327612   =>  1   ;   //  Body Wash, Suave
121327613   =>  1   ;   //  Toothpaste, Colgate
121327614   =>  1   ;   //  Cleaning Milk
121327615   =>  1   ;   //  Soap Ned
121327616   =>  1   ;   //  Other Articles Of Personal Hygiene
121327701   =>  1   ;   //  Baby Oil, 200Mls, Johnson
121327702   =>  1   ;   //  Baby Powder
121327703   =>  1   ;   //  Baby Powder, Johnsons
121327704   =>  1   ;   //  Body Lotion, Nivea
121327705   =>  1   ;   //  Deodorant Lynx Spray, Men
121327706   =>  1   ;   //  Deodorant Lynx Spray, Women
121327707   =>  1   ;   //  Deodorant Mum Roll-On, Dove (For Ladies)
121327708   =>  1   ;   //  Deodorant Mum Roll-On, Rexona, (For Ladies)
```

```
121327709   =>  1   ;   //  Deodorant Mum Roll-On, Rexona,(For Men)
121327710   =>  1   ;   //  Deodorant Spray, (Men), Rexona
121327711   =>  1   ;   //  Deodorant Spray, (Women), Rexona
121327712   =>  1   ;   //  Deodorant Spray, For Men, Nivea
121327713   =>  1   ;   //  Deodorant Spray, For Women, Nivea
121327714   =>  1   ;   //  Deodorant, Impulse
121327715   =>  1   ;   //  Deodorant, Lynx Roll-On, Men
121327716   =>  1   ;   //  Perfume
121327717   =>  1   ;   //  Make Up, Lipstick
121327718   =>  1   ;   //  Other Beauty Product
121327801   =>  1   ;   //  Baby Wipes, Huggies
121327802   =>  1   ;   //  Cotton Buds, Ear Buds
121327803   =>  1   ;   //  Ladies Pads, Johnsons
121327804   =>  1   ;   //  Ladies Pads, Libra
121327805   =>  1   ;   //  Ladies Pads, Modess
121327806   =>  1   ;   //  Ladies Pads, Stay Free
121327807   =>  1   ;   //  Nappies, Huggies
121327808   =>  1   ;   //  Nappies, Soft Love
121327809   =>  1   ;   //  Toilet Paper, Sofan
121327810   =>  1   ;   //  Toilet Paper, Swan
121327811   =>  1   ;   //  Toilet Paper, Wondersoft
121327812   =>  1   ;   //  Cotton
121327813   =>  1   ;   //  Nappies for adults, Depends
121327814   =>  1   ;   //  Other Product
122027901   =>  2   ;   //  Services provided by prostitutes and the like
123128001   =>  3   ;   //  Diamon, gold, platnum, and similar precious metals and stones
123128101   =>  3   ;   //  Jewellery (local or not): earing, necklace, bracelet
123128102   =>  3   ;   //  Artificial flowers, artificial tiare
123128103   =>  3   ;   //  Jewellery without any detail
123128104   =>  3   ;   //  Fresh flower arrangments in bouquet, lei, ei, head ea, ea
katu, poe tiare
123128201   =>  3   ;   //  Clocks
123128202   =>  3   ;   //  Watch
123128203   =>  3   ;   //  Alarm Clock
123128204   =>  3   ;   //  Other Clock And Watch Related Article
123128301   =>  2   ;   //  Repair Of Such Articles
123228401   =>  4   ;   //  Handbag
123228402   =>  4   ;   //  Suit Case
123228403   =>  4   ;   //  School Backpacks, (Boys)
123228404   =>  4   ;   //  School Backpacks, (Girls)
123228405   =>  4   ;   //  Other Travel Goods
123228501   =>  4   ;   //  Baby Carriages, Carrycots
123228502   =>  4   ;   //  Car Beds And Seats
123228503   =>  4   ;   //  Pushchairs
123228504   =>  4   ;   //  Reins And Harnesses
123228505   =>  4   ;   //  Other Articles For Babies
123228601   =>  4   ;   //  Lighter
123228602   =>  4   ;   //  Pipe
123228603   =>  4   ;   //  Ashtray
123228604   =>  4   ;   //  Other Article For Smokers
123228701   =>  4   ;   //  Umbrella
123228702   =>  4   ;   //  Umbrella (Beach)
123228703   =>  4   ;   //  Fan
123228704   =>  4   ;   //  Sunglasses
123228705   =>  4   ;   //  Fey Ring
123228706   =>  4   ;   //  Other Miscellaneous Personal Article
123228707   =>  4   ;   //  traditional crafts - fan
123228708   =>  4   ;   //  traditional crafts - handbags, bags, pouches
123228709   =>  4   ;   //  traditional crafts - hats
123228710   =>  4   ;   //  traditional crafts - mats
123228801   =>  4   ;   //  Funerary Articles Eg Coffins, Gravestones, Urns
123228901   =>  2   ;   //  Repair Of Such Articles
124029001   =>  2   ;   //  Retirement Homes For Elderly Persons And Residences For
Disabled Persons
124029101   =>  2   ;   //  Home help, day care and rehabilitation
125129201   =>  2   ;   //  Life Assurance
125129202   =>  2   ;   //  Education Assurance
125229301   =>  2   ;   //  Home Insurance
125329401   =>  2   ;   //  Health Insurance
125429501   =>  2   ;   //  Car Insurance
125429502   =>  2   ;   //  Boat Insurance
```

```
    125429503   =>  2   ;   //  Other Insurance Related To Transport
    125529601   =>  2   ;   //  Personal Insurance
    126129701   =>  2   ;   //  Financial intermediation services indirectly measured
    126229801   =>  2   ;   //  Banking Fees
    126229802   =>  2   ;   //  Money Transfer Fees
    126229803   =>  2   ;   //  Credit Card Fee
    126229804   =>  2   ;   //  Other Financial Fees
    127029901   =>  2   ;   //  Passport Fees
    127029902   =>  2   ;   //  Administrative Document, Certificate (Birth, Death…)
    127029903   =>  2   ;   //  Photocopies
    127029904   =>  2   ;   //  Fees related to the purchase of a house (appraisal, realty,
    etc.)
    127029905   =>  2   ;   //  Legal expenses
    127029907   =>  2   ;   //  Other fees related to personal services
    901100101-999999999 =>  5   ;   //  Current cash transfers: birthdays
endrecode;


if HC_COICOP in 1:970000000 and HC_MULTIPLIER > 1 then
    numeric tmp2;
    numeric tmp = noccurs(EXPENDITURE_EDT)+1;
    insert(EXPENDITURE_EDT(tmp));

    E_LINE_NUMBER(tmp) = tmp;
    E_HIES_CODE(tmp) = HC_HIES_CODE;
    E_COICOP(tmp) = HC_COICOP;
    E_AMOUNT(tmp) = HC_AMOUNT;
    E_AN_AMT(tmp) = HC_AN_AMT;
    E_WGHT_AN_AMT(tmp) = HC_WGHT_AN_AMT;
    E_MULTIPLIER(tmp) = HC_MULTIPLIER;
    E_LOCATION(tmp) = HC_LOCATION;
    E_EXP_TYPE(tmp) = HC_EXP_TYPE;
    E_PROD_TYPE(tmp) = HC_PROD_TYPE;
    E_NUMBER(tmp) = HC_QUANTITY;
    E_UNIT(tmp) = HC_UNIT;
    E_CATEGORY(tmp) = notappl;
    E_BENEFICIARY(tmp) = 51;
    E_DURABILITY(tmp) = tmpDura;
    E_DESCRIPTION(tmp) = "[diary home consume]"+tolower(HC_DESCRIPTION);

    // INCOME SIDE
    numeric tmpCOICOP = HC_COICOP;
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = HC_HIES_CODE; // using code below
    numeric tmpMulti = HC_MULTIPLIER;
    numeric tmpISCO = notappl; // not necessary for this section
    numeric tmpISIC = notappl; // not necessary for this seciton
    numeric tmpValue = HC_AMOUNT;
    numeric tmpIncomeType = HC_EXP_TYPE; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind
    home  [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = HC_LOCATION;
    numeric tmpBeneficiary = 51;
    string tmpDescription = "[diary home consume]"+tolower(HC_DESCRIPTION);
    numeric tmpProductType = HC_PROD_TYPE;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = HC_QUANTITY;
    numeric tmpUnit = HC_UNIT;
    numeric tmpCategory = notappl;
    numeric tmpDurability = tmpDura;

    tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
```

```
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
    I_CATEGORY(tmp) = tmpCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;



endif;
PROC HS_LINE_NUMBER

// first pass coicop correction on 21/3/2016
if ID07 in  111 and HS_LINE_NUMBER in   1   and strip(HS_DESCRIPTION[1:5]) = 'dress'    then
impute(HS_COICOP,  31209402    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  111 and HS_LINE_NUMBER in   2   and strip(HS_DESCRIPTION[1:5]) = 'dress'    then
impute(HS_COICOP,  31209402    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  135 and HS_LINE_NUMBER in   1   and strip(HS_DESCRIPTION[1:5]) = 'tunaf'    then
impute(HS_COICOP,  11302237    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  135 and HS_LINE_NUMBER in   2   and strip(HS_DESCRIPTION[1:5]) = 'tuna' then
impute(HS_COICOP,  11302237    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  138 and HS_LINE_NUMBER in   1   and strip(HS_DESCRIPTION[1:5]) = 'tuna' then
impute(HS_COICOP,  11302237    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  154 and HS_LINE_NUMBER in   1   and strip(HS_DESCRIPTION[1:5]) = 'reef' then
impute(HS_COICOP,  11302233    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   1   and strip(HS_DESCRIPTION[1:5]) = 'ice c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   2   and strip(HS_DESCRIPTION[1:5]) = '1ce c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   3   and strip(HS_DESCRIPTION[1:5]) = 'ice-c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   4   and strip(HS_DESCRIPTION[1:5]) = 'ice-c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   5   and strip(HS_DESCRIPTION[1:5]) = 'ice-c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  318 and HS_LINE_NUMBER in   6   and strip(HS_DESCRIPTION[1:5]) = 'ice-c'    then
impute(HS_COICOP,  11805803    );  errmsg("%s",HS_DESCRIPTION);    endif;
if ID07 in  347 and HS_LINE_NUMBER in   2   and strip(HS_DESCRIPTION[1:5]) = 'uto b'    then
impute(HS_COICOP,  11604232    );  errmsg("%s",HS_DESCRIPTION);    endif;

PROC HS_HIES_CODE
//second pass
recode  ID07    :   HS_LINE_NUMBER  =>  HS_COICOP   ;
    103 :   1   =>  999999998   ;
    111 :   1   =>  999999998   ;
    111 :   2   =>  999999998   ;
    135 :   1   =>  11302237    ;
    135 :   2   =>  11302237    ;
    138 :   1   =>  11302237    ;
    143 :   1   =>  999999998   ;
    154 :   1   =>  11302233    ;
    217 :   1   =>  11302238    ;
    301 :   1   =>  999999998   ;
    318 :   1   =>  999999998   ;
    318 :   2   =>  999999998   ;
    318 :   3   =>  999999998   ;
    318 :   4   =>  999999998   ;
    318 :   5   =>  999999998   ;
    318 :   6   =>  999999998   ;
    328 :   1   =>  11201202    ;
    328 :   2   =>  11604207    ;
    328 :   3   =>  11604210    ;
    328 :   4   =>  11302213    ;
    347 :   1   =>  11302238    ;
    347 :   2   =>  11604232    ;
```

```
    347 :   3   =>  999999998   ;
    347 :   4   =>  999999998   ;
    353 :   1   =>  999999998   ;
    353 :   2   =>  999999998   ;
endrecode;

$ = 552 + HS_DAY / 1000;
PROC HS_MULTIPLIER
if ID07 in 308,318,118,326,250,134,105 then
    if ID07 in 308 then if HS_DAY in 1:7 then HS_MULTIPLIER = 0; else HS_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 318 then if HS_DAY in 1:7 then HS_MULTIPLIER = 0; else HS_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 118 then if HS_DAY in 1:7 then HS_MULTIPLIER = 52; else HS_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 326 then if HS_DAY in 1:7 then HS_MULTIPLIER = 52; else HS_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 250 then if HS_DAY in 1:7 then HS_MULTIPLIER = 52; else HS_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 134 then if HS_DAY in 1:7 then HS_MULTIPLIER = 52; else HS_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 105 then if HS_DAY in 1:7 then HS_MULTIPLIER = 52; else HS_MULTIPLIER = 0;
    endif; endif;
else
    HS_MULTIPLIER = 26;
endif;

HS_AN_AMT = HS_AMOUNT * HS_MULTIPLIER;
HS_WGHT_AN_AMT = HS_AN_AMT * WEIGHT;


PROC HS_LOCATION
$ = 1;
PROC HS_EXP_TYPE
$ = 1;
PROC HS_LOCAL
$ = notappl;
PROC HS_PURPOSE
$ = 1;
PROC HG_LINE_NUMBER

// first pass coicop correction on 21/3/2016
if ID07 in  103 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'handi'    then
impute(HG_COICOP,  999999994   );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  103 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'handi'    then
impute(HG_COICOP,  999999994   );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  104 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  104 and HG_LINE_NUMBER in   6   and strip(HG_DESCRIPTION[1:5]) = 'green'    then
impute(HG_COICOP,  11704518    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  104 and HG_LINE_NUMBER in   8   and strip(HG_DESCRIPTION[1:5]) = 'pupki'    then
impute(HG_COICOP,  11704524    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  109 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'woven'    then
impute(HG_COICOP,  999999992   );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  111 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'dress'    then
impute(HG_COICOP,  31209402    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  111 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604210    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  111 and HG_LINE_NUMBER in   9   and strip(HG_DESCRIPTION[1:5]) = 'roast'    then
impute(HG_COICOP,  11201508    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  111 and HG_LINE_NUMBER in   11  and strip(HG_DESCRIPTION[1:5]) = 'sewin'    then
impute(HG_COICOP,  31209402    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  111 and HG_LINE_NUMBER in   12  and strip(HG_DESCRIPTION[1:5]) = 'dress'    then
impute(HG_COICOP,  31209402    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  121 and HG_LINE_NUMBER in   4   and strip(HG_DESCRIPTION[1:5]) = 'masim'    then
impute(HG_COICOP,  11302236    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  126 and HG_LINE_NUMBER in   7   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604210    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  127 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'puta' then
impute(HG_COICOP,  11100304    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  135 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'tokel'    then
```

```
impute(HG_COICOP,  999999991   );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  135 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'tuna' then
impute(HG_COICOP,  11302231    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  135 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'reef' then
impute(HG_COICOP,  11302233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  147 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'bread'    then
impute(HG_COICOP,  11604207    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  147 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'broom'    then
impute(HG_COICOP,  56117008    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  147 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'broom'    then
impute(HG_COICOP,  56117008    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  150 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'tuna' then
impute(HG_COICOP,  11302237    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  152 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'reef' then
impute(HG_COICOP,  11302233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  152 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'reef' then
impute(HG_COICOP,  11302233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  154 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'reef' then
impute(HG_COICOP,  11302233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  154 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'tuna' then
impute(HG_COICOP,  11302237    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  155 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'tuna' then
impute(HG_COICOP,  11302237    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  159 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'raw f'    then
impute(HG_COICOP,  11302233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  159 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'uto'  then
impute(HG_COICOP,  11604232    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  160 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'bread'    then
impute(HG_COICOP,  11604207    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  160 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'bread'    then
impute(HG_COICOP,  11604207    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  160 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'petro'    then
impute(HG_COICOP,  72219602    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  160 and HG_LINE_NUMBER in   4   and strip(HG_DESCRIPTION[1:5]) = 'rice' then
impute(HG_COICOP,  11100101    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   10  and strip(HG_DESCRIPTION[1:5]) = 'pawpa'    then
impute(HG_COICOP,  11604225    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   11  and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   13  and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   14  and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   15  and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   16  and strip(HG_DESCRIPTION[1:5]) = 'papay'    then
impute(HG_COICOP,  11604225    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  203 and HG_LINE_NUMBER in   17  and strip(HG_DESCRIPTION[1:5]) = 'panda'    then
impute(HG_COICOP,  11604233    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  220 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'bread'    then
impute(HG_COICOP,  11704805    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  241 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'taro' then
impute(HG_COICOP,  11704603    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  301 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'fish' then
impute(HG_COICOP,  11302236    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  302 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'banan'    then
impute(HG_COICOP,  11604202    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  304 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'donut'    then
impute(HG_COICOP,  11100302    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  306 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604209    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  310 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'banan'    then
impute(HG_COICOP,  11604205    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  310 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'banan'    then
impute(HG_COICOP,  11604205    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  311 and HG_LINE_NUMBER in   2   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604232    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  311 and HG_LINE_NUMBER in   4   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604232    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  311 and HG_LINE_NUMBER in   5   and strip(HG_DESCRIPTION[1:5]) = 'cocon'    then
impute(HG_COICOP,  11604209    );  errmsg("%s",HG_DESCRIPTION);    endif;
```

```
if ID07 in  318 and HG_LINE_NUMBER in   1   and strip(HG_DESCRIPTION[1:5]) = 'dough'    then
impute(HG_COICOP,  11100302    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  327 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = '2 tok'    then
impute(HG_COICOP,  999999991   );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  328 and HG_LINE_NUMBER in   3   and strip(HG_DESCRIPTION[1:5]) = 'pele' then
impute(HG_COICOP,  11704533    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  328 and HG_LINE_NUMBER in   5   and strip(HG_DESCRIPTION[1:5]) = 'pele' then
impute(HG_COICOP,  11704533    );  errmsg("%s",HG_DESCRIPTION);    endif;
if ID07 in  347 and HG_LINE_NUMBER in   7   and strip(HG_DESCRIPTION[1:5]) = 'clams'    then
impute(HG_COICOP,  11302310    );  errmsg("%s",HG_DESCRIPTION);    endif;

PROC HG_HIES_CODE
//second pass
recode  ID07    :   HG_LINE_NUMBER  =>  HG_COICOP   ;
    102 :   1   =>  999999998   ;
    102 :   2   =>  999999998   ;
    102 :   3   =>  999999998   ;
    103 :   1   =>  999999998   ;
    103 :   2   =>  999999998   ;
    103 :   4   =>  11302238    ;
    103 :   5   =>  999999998   ;
    103 :   6   =>  999999998   ;
    104 :   1   =>  999999998   ;
    104 :   4   =>  11302238    ;
    104 :   5   =>  999999998   ;
    108 :   1   =>  11302238    ;
    108 :   3   =>  11302238    ;
    109 :   1   =>  999999998   ;
    109 :   2   =>  11302238    ;
    111 :   1   =>  999999998   ;
    111 :   2   =>  11302238    ;
    111 :   4   =>  999999998   ;
    111 :   5   =>  999999998   ;
    111 :   6   =>  999999998   ;
    111 :   7   =>  999999998   ;
    111 :   8   =>  999999998   ;
    111 :   9   =>  999999998   ;
    111 :   10  =>  999999998   ;
    111 :   11  =>  999999998   ;
    111 :   12  =>  999999998   ;
    118 :   1   =>  11302238    ;
    118 :   2   =>  11302238    ;
    119 :   1   =>  11302237    ;
    121 :   4   =>  11302234    ;
    124 :   3   =>  11302237    ;
    124 :   4   =>  11302238    ;
    125 :   1   =>  11302237    ;
    125 :   2   =>  11302238    ;
    125 :   3   =>  999999998   ;
    125 :   4   =>  11302238    ;
    126 :   1   =>  11302237    ;
    126 :   2   =>  11302237    ;
    126 :   3   =>  11302237    ;
    126 :   4   =>  11302237    ;
    126 :   5   =>  11302237    ;
    126 :   6   =>  11302237    ;
    127 :   2   =>  999999998   ;
    132 :   1   =>  11302237    ;
    132 :   2   =>  11302237    ;
    135 :   1   =>  999999998   ;
    139 :   1   =>  11302237    ;
    139 :   2   =>  11302237    ;
    139 :   3   =>  11302237    ;
    147 :   2   =>  999999998   ;
    147 :   3   =>  999999998   ;
    160 :   3   =>  999999998   ;
    160 :   4   =>  999999998   ;
    203 :   5   =>  11302237    ;
    208 :   2   =>  11604232    ;
    208 :   4   =>  11504005    ;
    208 :   6   =>  11604232    ;
```

```
    217 :   1   =>  11302238    ;
    220 :   1   =>  11604207    ;
    301 :   1   =>  11302238    ;
    304 :   1   =>  999999998   ;
    304 :   2   =>  999999998   ;
    318 :   1   =>  999999998   ;
    318 :   2   =>  999999998   ;
    318 :   3   =>  999999998   ;
    323 :   1   =>  999999998   ;
    326 :   1   =>  999999998   ;
    326 :   2   =>  11302216    ;
    327 :   3   =>  999999998   ;
    332 :   1   =>  11302237    ;
    333 :   1   =>  999999998   ;
    339 :   1   =>  999999998   ;
    343 :   1   =>  999999998   ;
    346 :   2   =>  11302238    ;
    347 :   1   =>  11302238    ;
    347 :   2   =>  11302238    ;
    347 :   3   =>  11302238    ;
    347 :   5   =>  11302238    ;
    349 :   1   =>  11302238    ;
    349 :   2   =>  11302238    ;
    349 :   5   =>  11302238    ;
    353 :   1   =>  11302238    ;
    353 :   2   =>  11302238    ;
    354 :   1   =>  11302238    ;
    354 :   2   =>  11302238    ;
    354 :   3   =>  11302238    ;
    354 :   4   =>  11604232    ;
    354 :   5   =>  11302238    ;
    354 :   7   =>  999999998   ;
endrecode;

$ = 553 + HG_DAY / 1000;
PROC HG_AMOUNT
if ID07 = 150 then
    if HG_LINE_NUMBER = 1 then
        impute ($, 10);
    endif;
endif;

PROC HG_MULTIPLIER
if ID07 in 308,318,118,326,250,134,105 then
    if ID07 in 308 then if HG_DAY in 1:7 then HG_MULTIPLIER = 0; else HG_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 318 then if HG_DAY in 1:7 then HG_MULTIPLIER = 0; else HG_MULTIPLIER = 52;
    endif; endif;
    if ID07 in 118 then if HG_DAY in 1:7 then HG_MULTIPLIER = 52; else HG_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 326 then if HG_DAY in 1:7 then HG_MULTIPLIER = 52; else HG_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 250 then if HG_DAY in 1:7 then HG_MULTIPLIER = 52; else HG_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 134 then if HG_DAY in 1:7 then HG_MULTIPLIER = 52; else HG_MULTIPLIER = 0;
    endif; endif;
    if ID07 in 105 then if HG_DAY in 1:7 then HG_MULTIPLIER = 52; else HG_MULTIPLIER = 0;
    endif; endif;
else
    HG_MULTIPLIER = 26;
endif;

HG_AN_AMT = HG_AMOUNT * HG_MULTIPLIER;
HG_WGHT_AN_AMT = HG_AN_AMT * WEIGHT;


PROC HG_LOCATION
$ = 1;
PROC HG_EXP_TYPE
$ = 4;
PROC HG_LOCAL
```

```
$ = notappl;
PROC HG_PURPOSE
$ = 2;
PROC HG_DAY_NAME
numeric tmpDura;
recode  HG_COICOP   =>  tmpDura ;
    11100101-23109001   =>  1   ;   //  Food and drinks
    31109101    =>  4   ;   //  Pareu (printed or not)
    31109102    =>  4   ;   //  Clothing fabric, poplin material
    31109201    =>  4   ;   //  Pareu
    31109202    =>  4   ;   //  Material, Fabrics
    31109203    =>  4   ;   //  Poplin
    31209301    =>  4   ;   //  Men And Boys - Lavalava
    31209302    =>  4   ;   //  Men And Boys  - Shirt
    31209303    =>  4   ;   //  Men And Boys - Trousers
    31209304    =>  4   ;   //  Men And Boys  - T-Shirt
    31209305    =>  4   ;   //  Men And Boys - Underwear
    31209306    =>  4   ;   //  Men And Boys - School Uniform
    31209307    =>  4   ;   //  Men And Boys - Short
    31209308    =>  4   ;   //  Men And Boys - Socks
    31209309    =>  4   ;   //  Men And Boys - Other Clothes For Man.Boy Or Ned
    31209310    =>  4   ;   //  Men And Boys  - Clothes (general - not enough detail)
    31209401    =>  4   ;   //  Women And Girls - Shirt
    31209402    =>  4   ;   //  Women And Girls - Dress, Skirt
    31209403    =>  4   ;   //  Women And Girls - Trousers, Pants
    31209404    =>  4   ;   //  Women And Girls - T-Shirt, Top
    31209405    =>  4   ;   //  Women And Girls  - Suit
    31209406    =>  4   ;   //  Women And Girls - Underwear
    31209407    =>  4   ;   //  Women And Girls - School Uniform
    31209408    =>  4   ;   //  Women And Girls - Short
    31209409    =>  4   ;   //  Women And Girls - Socks
    31209410    =>  4   ;   //  Women And Girls - Other Clother For Women/Girl Or Ned
    31209411    =>  4   ;   //  Women And Girls - lava-lava
    31209412    =>  4   ;   //  Women And Girls - Clothes (general - not enough detail)
    31209501    =>  4   ;   //  Garments For Infant 0 To 2 Years Old
    31209601    =>  4   ;   //  Unspecifed / unisex clothes in general - shirts / tops
    31209602    =>  4   ;   //  Unspecifed / unisex clothes in general - pants, jeans,
    trousers
    31209603    =>  4   ;   //  Unspecifed / unisex clothes in general - dresses, slacks,
    and similar
    31209604    =>  4   ;   //  Unspecifed / unisex clothes in general - underwear, socks
    31209605    =>  4   ;   //  Unspecifed / unisex clothes in general - pajamas, robes
    31209606    =>  4   ;   //  Unspecifed / unisex clothes in general - school uniforms
    31209607    =>  4   ;   //  Unspecifed / unisex clothes in general - business, work attire
    31209608    =>  4   ;   //  Unspecifed / unisex clothes in general - Not enough detail
    31209609    =>  4   ;   //  Unspecifed / unisex clothes in general - Not specified
    elsewhere
    31209610    =>  4   ;   //  Unspecifed / unisex clothes in general - lava lava
    31309601    =>  4   ;   //  Cotton Gloves
    31309602    =>  4   ;   //  Tie
    31309603    =>  4   ;   //  Belt
    31309604    =>  4   ;   //  Cap
    31309605    =>  4   ;   //  Hat
    31309606    =>  4   ;   //  Other Clothing Accessories
    31309701    =>  4   ;   //  Cotton Thread
    31309702    =>  4   ;   //  Rayon Material
    31309703    =>  4   ;   //  Sewing Needles, Hand
    31309704    =>  4   ;   //  Other Sewing Materials
    31409801    =>  4   ;   //  Dyeing Of Garments
    31409901    =>  4   ;   //  Cleaning Of Garments, Laundry Shop
    31410001    =>  4   ;   //  Taylor
    31410101    =>  4   ;   //  Hire Of Garments
    32110201    =>  4   ;   //  Male - Beach Jandals
    32110202    =>  4   ;   //  Male - Sport Shoes
    32110203    =>  4   ;   //  Male - Casual Shoes
    32110204    =>  4   ;   //  Male - Other Shoes For Men And Boys Or Ned
    32110301    =>  4   ;   //  Female - Beach Jandals
    32110302    =>  4   ;   //  Female - Sport Shoes
    32110303    =>  4   ;   //  Female - Casual Shoes
    32110304    =>  4   ;   //  Female - Other Shoes For Woman And Girl Or Ned
    32110401    =>  4   ;   //  Shoes For Infant - 0 To 2 Years
```

```
32110501    =>  4   ;   //  !!DO NOT USE !!
32110502    =>  4   ;   //  Part Of Shoes (Heels, Soles, Lace…)
32110503    =>  4   ;   //  Other shoes related articles
32210601    =>  2   ;   //  Repair Of Footwear
32210701    =>  2   ;   //  Shoe-Cleaning Service
32210801    =>  2   ;   //  Hire of footwear
41110901    =>  2   ;   //  Rent Paid By Tenants For Main House
41211001    =>  2   ;   //  Rent Paid By Tenants For Land / Land Lease
41211101    =>  2   ;   //  Rent Paid By Tenants For Another House
42111201    =>  2   ;   //  Imputed Rentals Of Owner Occupying Their Main House
42211301    =>  2   ;   //  Imputed Rental For Secondary Residence
42211401    =>  2   ;   //  Imputed Rentals Of Households Paying A Reduced Rental Or
Housed Free
43111501    =>  4   ;   //  Pipe
43111502    =>  4   ;   //  Buteline
43111503    =>  4   ;   //  Elbow
43111504    =>  4   ;   //  Tap
43111505    =>  4   ;   //  Other Plumbing Materials
43111601    =>  4   ;   //  Ceramic Floor Tile
43111602    =>  4   ;   //  Floor Mosaic (Shower)
43111603    =>  4   ;   //  Other Surfacing Materials (Primaflex…)
43111701    =>  4   ;   //  Masking Tape For Painting
43111702    =>  4   ;   //  Paint Brush
43111703    =>  4   ;   //  Paint
43111704    =>  4   ;   //  Other Painting Materials
43111801    =>  4   ;   //  Blocks,Concrete Blocks
43111802    =>  4   ;   //  Cement
43111803    =>  4   ;   //  Door
43111804    =>  4   ;   //  Window, Louvre
43111805    =>  4   ;   //  Electric Cable inc network cable
43111806    =>  4   ;   //  Electric Materials (Plugs…)
43111807    =>  4   ;   //  Sink
43111808    =>  4   ;   //  Toilet Bowl, Sit, Shower
43111809    =>  4   ;   //  Plywood
43111810    =>  4   ;   //  Iron Roofing
43111811    =>  4   ;   //  Timber inc posts, planks, scotia, weatherboard, skirting
43111812    =>  4   ;   //  Hardboard
43111813    =>  4   ;   //  Inuslation foil, DPC
43111814    =>  4   ;   //  Tank iron
43111815    =>  4   ;   //  Rebar, reinforcing steel
43111816    =>  4   ;   //  Angle bar
43111817    =>  4   ;   //  Gib board
43111818    =>  4   ;   //  Cyclone straps
43111819    =>  4   ;   //  Ridge capping
43111820    =>  4   ;   //  Spouting, gutters
43111821    =>  4   ;   //  Other material for house maintenance ned
43211901    =>  2   ;   //  Contractor, Builder, Plumber, Electrician…
44112001    =>  2   ;   //  Water Bill, Water Meter
44112002    =>  2   ;   //  Water connection fees
44112003    =>  2   ;   //  Water truck
44112004    =>  2   ;   //  Other fee related to water supply
44212101    =>  2   ;   //  Garbage disposal
44312201    =>  2   ;   //  Sewer collection and disposal
44412301    =>  2   ;   //  Compound charge, co proprietor charge for caretaking of the
compound / building (cleaning….)
44412401    =>  2   ;   //  Security Services
45112501    =>  2   ;   //  Electricity Charge, Power Meter, Bill
45112502    =>  2   ;   //  Other Electricity Connection Fees
45212601    =>  2   ;   //  Do not use this commodity thank you - town gas and natural gas
45212701    =>  2   ;   //  Gas Cylinder
45212702    =>  1   ;   //  Gas Refill
45312801    =>  1   ;   //  Kerosene for domestic heating and lighting (fuel for vehicle
see 07.2.2_196.03)
45312802    =>  1   ;   //  Benzene
45312803    =>  1   ;   //  Petrol
45312804    =>  1   ;   //  Other Fuel For Generator Or Lights (Oil…)
45412901    =>  1   ;   //  Coconut Husk
45412902    =>  1   ;   //  Charcoal Imported
45412903    =>  1   ;   //  Other Solid Fuel (Firewood…)
45513001    =>  2   ;   //  Ice used for cooling and refrigeration purposes, ice cube,
ice blocks
```

```
51113101    =>  3   ;   //  Couche, Sofa
51113102    =>  3   ;   //  Other Lounge Furniture Or Ned
51113102    =>  3   ;   //  Bookshelf
51113201    =>  3   ;   //  Dining Table
51113202    =>  3   ;   //  Other Table
51113203    =>  3   ;   //  Chairs
51113204    =>  3   ;   //  Dining Hall Cabinet
51113205    =>  3   ;   //  Other Dining Hall Furniture Or Ned
51113301    =>  3   ;   //  Beds With Mattress
51113302    =>  3   ;   //  Beds Without Mattress
51113303    =>  3   ;   //  Mattress
51113304    =>  3   ;   //  Chest Of Drawers
51113305    =>  3   ;   //  Cupboards
51113306    =>  3   ;   //  Dressing Table
51113307    =>  3   ;   //  Wardrobe
51113308    =>  3   ;   //  Other Bedroom Furniture Or Ned
51113401    =>  3   ;   //  Foodsafe
51113402    =>  3   ;   //  Pantry Cupboard
51113403    =>  3   ;   //  Other Kitchen Furniture
51113501    =>  3   ;   //  Outdoor Chairs
51113502    =>  3   ;   //  Outdoor Table
51113503    =>  3   ;   //  Other Outdoor Furniture Or Ned
51113601    =>  3   ;   //  Cradles
51113602    =>  3   ;   //  High Chairs
51113603    =>  3   ;   //  Playpens
51113604    =>  3   ;   //  Other Baby Furniture Or Ned
51113701    =>  3   ;   //  Bathroom Cabinets
51113702    =>  3   ;   //  Other Bathroom Furniture Or Ned
51113801    =>  3   ;   //  Halogen Lamps
51113802    =>  3   ;   //  Other Lamp Or Ned (Exclude Torch)
51113901    =>  3   ;   //  Computer Table
51113902    =>  3   ;   //  Mirror
51113903    =>  3   ;   //  Bookcase
51113904    =>  3   ;   //  Decoration, Carving, Painting…
51113905    =>  3   ;   //  Other Furniture Or Ned
51214001    =>  3   ;   //  Carpet
51214002    =>  3   ;   //  Mat
51214003    =>  3   ;   //  Other Floor Cover
51314101    =>  2   ;   //  Repair Of Furniture
52114201    =>  4   ;   //  Awnings
52114202    =>  4   ;   //  Curtain, Curtain Material
52114203    =>  4   ;   //  Fabric Blinds
52114301    =>  4   ;   //  Bolsters
52114302    =>  4   ;   //  Futons
52114303    =>  4   ;   //  Hammocks
52114304    =>  4   ;   //  Pillows Including Cushions
52114401    =>  4   ;   //  Bed Sheets
52114402    =>  4   ;   //  Bed Spreads
52114403    =>  4   ;   //  Bedding Sets - Combination Eg Fitted, Flat And Pillow Cases
52114404    =>  4   ;   //  Blankets
52114405    =>  4   ;   //  Eiderdowns
52114406    =>  4   ;   //  Mosquito Nets
52114407    =>  4   ;   //  Pillow Cases
52114408    =>  4   ;   //  Other Bedding Or Ned
52114501    =>  4   ;   //  Table Clothes, Napkin
52114601    =>  4   ;   //  Face Towel
52114602    =>  4   ;   //  Hand Towel
52114603    =>  4   ;   //  Bath Towel
52114604    =>  4   ;   //  Beach Towel
52114605    =>  4   ;   //  Other Towel Or Ned
52114701    =>  4   ;   //  Shopping Plastic Bags
52114702    =>  4   ;   //  Shower Curtain
52114703    =>  4   ;   //  Covers For Clothes And Furniture
52114704    =>  4   ;   //  Doilies
52114705    =>  4   ;   //  Flags
52114706    =>  4   ;   //  Laundry Bags
52114707    =>  4   ;   //  Shoe Bags
52114708    =>  4   ;   //  Tea Towels
52114709    =>  4   ;   //  Other Household Textile
52114801    =>  2   ;   //  Repair Of Such Textiles
53114901    =>  3   ;   //  Stove, Kerosene, 2-Burner
```

```
53114902    =>  3   ;   //  Wicks, Kerosene Stove
53114903    =>  3   ;   //  Dishwasher
53114904    =>  3   ;   //  Freezer
53114905    =>  3   ;   //  Fridge-Freezer
53114906    =>  3   ;   //  Microwave
53114907    =>  3   ;   //  Oven Only
53114908    =>  3   ;   //  Refrigerator
53114909    =>  3   ;   //  Stove  – Electric
53114910    =>  3   ;   //  Stove – Gas
53114911    =>  3   ;   //  Stove - Kerosen
53114912    =>  3   ;   //  Wick For Kerosen Stove
53114913    =>  3   ;   //  Stove Ned
53114914    =>  3   ;   //  Other Major Kitchen Appliance Or Ned
53115001    =>  3   ;   //  Clothes Drier
53115002    =>  3   ;   //  Washing Machine
53115003    =>  3   ;   //  Other major laundry appliances
53115101    =>  3   ;   //  Air-Conditioner
53115102    =>  3   ;   //  Electric Water Heater (For Bathing)
53115103    =>  3   ;   //  Solar Power Water Heater
53115104    =>  3   ;   //  Extractor Hoods
53115105    =>  3   ;   //  Safes
53115106    =>  3   ;   //  Sewing Machine
53115107    =>  3   ;   //  Vacuum Cleaners, Steam Cleaning Machines, Carpet Shampooing
Machines And Machines For Scrubbing, Waxing And Polishing Floors
53115108    =>  3   ;   //  Other Major Household Appliances
53115109    =>  3   ;   //  Generator
53115110    =>  3   ;   //  Solar panel
53115111    =>  3   ;   //  Water tank
53215201    =>  4   ;   //  Jug, Electric, Steel
53215202    =>  4   ;   //  Fan For Ceiling
53215203    =>  4   ;   //  Fan For Table, Gie
53215204    =>  4   ;   //  Fan, Pedestal
53215205    =>  4   ;   //  Electric Toaster
53215206    =>  4   ;   //  Rice, Frc Rice Cooker
53215207    =>  4   ;   //  Blenders
53215208    =>  4   ;   //  Can-Openers
53215209    =>  4   ;   //  Coffee Maker
53215210    =>  4   ;   //  Deep Fryers
53215211    =>  4   ;   //  Food Processor
53215212    =>  4   ;   //  Frypans
53215213    =>  4   ;   //  Irons
53215214    =>  4   ;   //  Toaster
53215215    =>  4   ;   //  Other Small Electric Appliance
53315301    =>  2   ;   //  Repair Of Household Appliance
54015401    =>  4   ;   //  Ceremic Ware
54015402    =>  4   ;   //  China (Bone China, Royal Dalton, Lladro)
54015403    =>  4   ;   //  Crystal
54015404    =>  4   ;   //  Enamel
54015405    =>  4   ;   //  Glassware
54015406    =>  4   ;   //  Plastic Ware
54015407    =>  4   ;   //  Steel
54015408    =>  4   ;   //  Other Glassware Or Glassware Ned
54015501    =>  4   ;   //  Cutlery, Flatware And Silverware
54015601    =>  4   ;   //  Frying Pans
54015602    =>  4   ;   //  Hotplates
54015603    =>  4   ;   //  Household Scales
54015604    =>  4   ;   //  Kettle
54015605    =>  4   ;   //  Pots
54015606    =>  4   ;   //  Pressure Cookers
54015607    =>  4   ;   //  Saucepan
54015608    =>  4   ;   //  Tauaga (coconut strainer)
54015609    =>  4   ;   //  Other Non Electric Kitchen Ustensils Or Ned
54015701    =>  4   ;   //  Gas Lighter
54015702    =>  4   ;   //  Gas Lighter, Bic
54015703    =>  4   ;   //  Bottle Racks
54015704    =>  4   ;   //  Buckets
54015705    =>  4   ;   //  Containers For Bread, Coffee, Spices
54015706    =>  4   ;   //  Feeding Bottles
54015707    =>  4   ;   //  Iceboxes
54015708    =>  4   ;   //  Ironing Boards
54015709    =>  4   ;   //  Laundry Baskets
```

```
54015710    =>  4   ;   //  Letter Boxes
54015711    =>  4   ;   //  Portable Money Boxes
54015712    =>  4   ;   //  Thermos Flasks
54015713    =>  4   ;   //  Towel Rails
54015714    =>  4   ;   //  Waste Bins
54015715    =>  4   ;   //  Waste Paper Baskets
54015716    =>  4   ;   //  Scales, Counter 20Kg
54015717    =>  4   ;   //  Hand Basin With Brackets (For Toilets)
54015718    =>  4   ;   //  Jug, Plastic,
54015719    =>  4   ;   //  Plastic Jerry Cans, 10 Litres
54015720    =>  4   ;   //  Tarpaulin,
54015721    =>  4   ;   //  Tarpaulin Clear (Heavy Plastic)
54015722    =>  4   ;   //  Other Non Electric Hh Article Or Ned
54015801    =>  2   ;   //  Repair Of Household Article
55115901    =>  3   ;   //  Cement Mixer Petrol
55115902    =>  3   ;   //  Water Pump Electric Orange Sj400.
55115903    =>  3   ;   //  Compression
55115904    =>  3   ;   //  Engine Cleaning Gun
55115905    =>  3   ;   //  V Belts ( For Concrete Mixers)
55115906    =>  3   ;   //  Brush Cutters
55115907    =>  3   ;   //  Chainsaws
55115908    =>  3   ;   //  Cultivators
55115909    =>  3   ;   //  Electric Drills
55115910    =>  3   ;   //  Garden Tractors
55115911    =>  3   ;   //  Lawnmowers
55115912    =>  3   ;   //  Motor Mowers
55115913    =>  3   ;   //  Sanders And Hedge Cutters
55115914    =>  3   ;   //  Other Motorized Tools And Equipment
55116001    =>  2   ;   //  Repair Of Such Articles
55216101    =>  3   ;   //  Dustmask
55216102    =>  3   ;   //  Hacksaws
55216103    =>  3   ;   //  Hand Saw
55216104    =>  3   ;   //  Petrol Pump
55216105    =>  3   ;   //  Circular Saw
55216106    =>  3   ;   //  Clamp Tool
55216107    =>  3   ;   //  Air Duster Gun
55216108    =>  3   ;   //  Coach Screws
55216109    =>  3   ;   //  Dyna Bolts
55216110    =>  3   ;   //  Crowbar
55216111    =>  3   ;   //  Cut Off Wheel 230X2.5X22Mm
55216112    =>  3   ;   //  Hacksaw
55216113    =>  3   ;   //  Matic Guns, Standard # Skeleton
55216114    =>  3   ;   //  Multimeter Pocket Digital
55216115    =>  3   ;   //  Sandpaper
55216116    =>  3   ;   //  Spray Gun Suction
55216117    =>  3   ;   //  Trowel
55216118    =>  3   ;   //  Wire Brush
55216119    =>  3   ;   //  Files
55216120    =>  3   ;   //  Hammers
55216121    =>  3   ;   //  Pliers
55216122    =>  3   ;   //  Rasps
55216123    =>  3   ;   //  Saws
55216124    =>  3   ;   //  Screwdrivers
55216125    =>  3   ;   //  Spanners
55216126    =>  3   ;   //  Trimming Knives
55216127    =>  3   ;   //  Wrenches
55216128    =>  3   ;   //  Other Hand Tools
55216201    =>  3   ;   //  Hose 1With Fittings
55216202    =>  3   ;   //  Cane Knives, Bush Knives
55216203    =>  3   ;   //  Forks
55216204    =>  3   ;   //  Hoses And Sprinklers
55216205    =>  3   ;   //  Rakes
55216206    =>  3   ;   //  Scythes
55216207    =>  3   ;   //  Shovels
55216208    =>  3   ;   //  Sickles
55216209    =>  3   ;   //  Spades
55216210    =>  3   ;   //  Watering Cans
55216211    =>  3   ;   //  Wheelbarrows
55216212    =>  3   ;   //  Other Garden Tools
55216301    =>  3   ;   //  Ladders And Steps
55216401    =>  3   ;   //  Entrance Door Lock Set
```

```
55216402 => 3 ; // Hasp & Staple
55216403 => 3 ; // Security Door Locks C/W Knob/Handles
55216404 => 3 ; // Security Entrance Door Locks/Knob Handles
55216405 => 3 ; // Toilet Door Lock Set
55216406 => 3 ; // Brass Padlocks
55216407 => 3 ; // Pad Bolt
55216408 => 3 ; // Padlock
55216409 => 3 ; // Handles
55216410 => 3 ; // Hinges
55216411 => 3 ; // Security mesh
55216412 => 3 ; // Other security materials ned
55216501 => 3 ; // Shower Curtain Rods
55216502 => 3 ; // Toilet Paper Holder
55216503 => 3 ; // Toilet Seat Covers
55216504 => 3 ; // Carpet Rods
55216505 => 3 ; // Curtain Rails
55216506 => 3 ; // Hooks
55216507 => 3 ; // Other Metal Article For The House
55216601 => 3 ; // Chains
55216602 => 3 ; // Grids
55216603 => 3 ; // Hoop Segments For Fencing And Bordering
55216604 => 3 ; // Stakes
55216605 => 3 ; // Tie wire (included chicken wire)
55216606 => 3 ; // Mesh, meshwire
55216607 => 3 ; // Other Metal Article For The Garden
55216701 => 3 ; // Battery (Sk-N100)
55216702 => 3 ; // Battery (Sk-N70Zl)
55216703 => 3 ; // Battery, Alkaline
55216704 => 3 ; // Battery, Energizer
55216705 => 3 ; // Battery, Eveready
55216706 => 3 ; // Battery, Magnacell
55216707 => 3 ; // Battery, Panasonic
55216708 => 3 ; // Battery Ned
55216709 => 3 ; // Energy Saver Lights
55216710 => 3 ; // Extension Cords
55216711 => 3 ; // Fluorescent Light
55216712 => 3 ; // Fluorescent Tube
55216713 => 3 ; // Light Bulb
55216714 => 3 ; // Torches Powered Diving Waterproof
55216715 => 3 ; // Spark Plugs
55216716 => 3 ; // Alarms
55216717 => 3 ; // Bells
55216718 => 3 ; // Flashlights
55216719 => 3 ; // Fluorescent Tubes starters
55216720 => 3 ; // Hand Lamps
55216721 => 3 ; // Power Sockets
55216722 => 3 ; // Switches
55216723 => 3 ; // Torches
55216724 => 3 ; // Wiring Flex
55216725 => 3 ; // Other Small Electric Appliance
55216801 => 2 ; // Repair Of Such Articles
56116901 => 1 ; // Air Freshener
56116902 => 1 ; // Dish Washing Liquid
56116903 => 1 ; // Dish Washing Liquid, Joy
56116904 => 1 ; // Disinfectant, Dettol
56116905 => 1 ; // Disinfectant: Pine-O-Cleen
56116906 => 1 ; // Fly Spray Surface, Mortein
56116907 => 1 ; // Fly Spray, Red, Mortein
56116908 => 1 ; // Insect Repellent, Aeroguard
56116909 => 1 ; // Rat Trap
56116910 => 1 ; // Soap, Laundry, Pacific
56116911 => 1 ; // Soap, Laundry, Sudso
56116912 => 1 ; // Toilet Cleaner, Harpic
56116914 => 1 ; // Washing Powder, Boom
56116915 => 1 ; // Washing Powder, Gain
56116916 => 1 ; // Washing Powder, Lerosa
56116917 => 1 ; // Washing Powder, Surf
56116918 => 1 ; // Washing Powder, Tasi
56116919 => 1 ; // Cleaning Paste Axion
56116920 => 1 ; // Cold Power
56116921 => 1 ; // Fabric Softener, Downy
```

```
56116922    =>  1   ;   //  Moth Balls
56116923    =>  1   ;   //  Napisan Powder
56116924    =>  1   ;   //  Bleach, Janola
56116925    =>  1   ;   //  Clorox
56116926    =>  1   ;   //  Clorox, Janola,
56116927    =>  1   ;   //  Vaselin
56116928    =>  1   ;   //  Other Cleaning And Maintenance Products
56117001    =>  1   ;   //  Sponge
56117002    =>  1   ;   //  Sponge Mop With Handle
56117003    =>  1   ;   //  Dustpan And Brush (2 Piece)
56117004    =>  1   ;   //  Mop Complete
56117005    =>  1   ;   //  Mops & Buckets
56117006    =>  1   ;   //  Toilet Cleaning Brushes C/W Stand
56117008    =>  1   ;   //  Brooms
56117009    =>  1   ;   //  Duster
56117010    =>  1   ;   //  Wired steelo
56117010    =>  1   ;   //  Other Articles For Cleaning
56117101    =>  1   ;   //  Aluminium Foil
56117102    =>  1   ;   //  Foam Cups
56117103    =>  1   ;   //  Foam Plates
56117104    =>  1   ;   //  Napkins, Paper
56117105    =>  1   ;   //  Plastic Cups
56117106    =>  1   ;   //  Plastic Dinner Plates
56117107    =>  1   ;   //  Plastic Spoons
56117108    =>  1   ;   //  Table Cloth, Plastic,
56117109    =>  1   ;   //  Filter
56117110    =>  1   ;   //  Vaccum Cleaner Bags
56117111    =>  1   ;   //  Kitchen Paper, Tissue, Paper Towel
56117112    =>  1   ;   //  Plastic Bin Liner
56117113    =>  1   ;   //  Other Paper Product
56117201    =>  1   ;   //  Candle
56117202    =>  1   ;   //  Adhesive Tapes
56117203    =>  1   ;   //  Camphor
56117204    =>  1   ;   //  String rope
56117205    =>  1   ;   //  Clothes□Hangers
56117206    =>  1   ;   //  Fire Extinguishers For Households
56117207    =>  1   ;   //  Glues
56117208    =>  1   ;   //  Incense Sticks
56117209    =>  1   ;   //  Knitting Needles
56117210    =>  1   ;   //  Lamp Wicks
56117211    =>  1   ;   //  Anchor Bolts M12 X 300
56117212    =>  1   ;   //  Rivet
56117213    =>  1   ;   //  Nail plate
56117214    =>  1   ;   //  Screws
56117215    =>  1   ;   //  Nail
56117216    =>  1   ;   //  Rubber gloves
56117222    =>  1   ;   //  PVC glue
56117224    =>  1   ;   //  Hinges
56117225    =>  1   ;   //  Matches
56117226    =>  1   ;   //  Measuring Tape
56117227    =>  1   ;   //  Mosquito Coil
56117228    =>  1   ;   //  Mosquito Coil, Zap
56117237    =>  1   ;   //  Packaging Tape
56117241    =>  1   ;   //  Cloth Pegs Hard Plastic
56117244    =>  1   ;   //  Other Non Durable Articles
56217301    =>  2   ;   //  Butlers
56217302    =>  2   ;   //  Cooks
56217303    =>  2   ;   //  Drivers
56217304    =>  2   ;   //  Gardeners
56217305    =>  2   ;   //  Governesses
56217306    =>  2   ;   //  Maids
56217307    =>  2   ;   //  Secretaries
56217308    =>  2   ;   //  Tutors
56217309    =>  2   ;   //  Baby sitting / child minding services provided at home
56217310    =>  2   ;   //  Elderly care services provided at home
56217398    =>  2   ;   //  Other Domestic Services - not elsewhere classified (nec)
56217399    =>  2   ;   //  Other Domestic Services - not enough detail (ned)
56217401    =>  2   ;   //  Domestic services, including baby-sitting and housework,
supplied by enterprises or self-employed persons
56217501    =>  2   ;   //  Pest control, fumigation
56217502    =>  2   ;   //  Lawn mowing services
```

```
56217503    =>  2   ;   //  Other household services provided by external company to the
household
56217601    =>  2   ;   //  Dry-cleaning, laundering and dyeing of household linen,
household textiles carpets
56217701    =>  2   ;   //  Hire of furniture, equipment (e.g. hire of cement mixer…)
61117801    =>  1   ;   //  Prescription medicine
61117901    =>  1   ;   //  Tablets, Panadol
61117902    =>  1   ;   //  Vicks Vaporub
61117903    =>  1   ;   //  Micreme
61117904    =>  1   ;   //  Panadol
61117905    =>  1   ;   //  Other Patent Medecine
61118001    =>  1   ;   //  Cod Liver Oil
61118002    =>  1   ;   //  Halibut Liver Oil
61118003    =>  1   ;   //  Oral Contraceptives
61118004    =>  1   ;   //  Serums And Vaccines
61118005    =>  1   ;   //  Vitamins And Minerals
61118006    =>  1   ;   //  Other Pharmaceutical Products Or Ned
61218101    =>  1   ;   //  Band Aids, Jj
61218102    =>  1   ;   //  Bandages
61218103    =>  1   ;   //  Clinical Thermometers
61218104    =>  1   ;   //  Condoms
61218105    =>  1   ;   //  First□Aid Kits
61218106    =>  1   ;   //  Hot Water Bottles
61218107    =>  1   ;   //  Hypodermic Syringes
61218108    =>  1   ;   //  Ice Bags
61218109    =>  1   ;   //  Mechanical Contraceptive Devises
61218110    =>  1   ;   //  Medical Hosiery Items
61218111    =>  1   ;   //  Pregnancy Tests
61218112    =>  1   ;   //  Other Medical Product
61318201    =>  3   ;   //  Artificial Limbs And Other Prosthetic Devices
61318202    =>  3   ;   //  Corrective Eye Glasses And Contact Lenses
61318203    =>  3   ;   //  Crutches
61318204    =>  3   ;   //  Dentures Excluding Fitting Costs
61318205    =>  3   ;   //  Devices For Monitoring Blood Pressure
61318206    =>  3   ;   //  Glass Eyes
61318207    =>  3   ;   //  Health Lamps
61318208    =>  3   ;   //  Hearing Aids
61318209    =>  3   ;   //  Medical Massage Equipment
61318210    =>  3   ;   //  Neck Braces
61318211    =>  3   ;   //  Orthopaedic Braces And Supports
61318212    =>  3   ;   //  Orthopaedic Footwear
61318213    =>  3   ;   //  'Special Beds'
61318214    =>  3   ;   //  Surgical Belts, Trusses And Supports
61318215    =>  3   ;   //  Wheelchairs – Powered And Unpowered And Invalid Carriers
61318216    =>  3   ;   //  Other Therapeutic Appliances
61318301    =>  2   ;   //  Repair or hire of therapeutic appliances
62118401    =>  2   ;   //  Consultations Of Doctor, Specialist
62218501    =>  2   ;   //  Consultations Of Dentist
62318601    =>  2   ;   //  Blood Test, Xray, Analysis
62318602    =>  2   ;   //  Nurse And Midwife
62318603    =>  2   ;   //  Other Paramedical Doctor
62318604    =>  2   ;   //  Other paramedical services (chiropractor…)
62318701    =>  2   ;   //  Traditional Healers
63018801    =>  2   ;   //  Hosptial, Clinic Fee
63018901    =>  2   ;   //  Medical service eg services of physicians
63019001    =>  2   ;   //  Paramedical services eg services of chiropractors,
optometrists
71119101    =>  3   ;   //  Car
71119102    =>  3   ;   //  Purchase of pick up truck, utility vehicle
71119103    =>  3   ;   //  Purchase of truck, mini bus, van
71219201    =>  3   ;   //  Motor Cycle
71219202    =>  3   ;   //  Scooter
71319301    =>  3   ;   //  Bicycle
71419401    =>  3   ;   //  Animal drawn vehicle
72119501    =>  4   ;   //  Aluminium Round Tubes For Boat
72119502    =>  4   ;   //  Fan Belt
72119503    =>  4   ;   //  Tyre
72119504    =>  4   ;   //  Tyre Tubes
72119505    =>  4   ;   //  Batteries For Vehicle
72119506    =>  4   ;   //  Tyre inflator
72119507    =>  4   ;   //  Other Spare Part For Vehicle And Boat
```

```
72219601    =>  1   ;   //  Fuel / gas for cars and land vehicle (private use)
72219602    =>  1   ;   //  Fuel / gas for boats and other water crafts (private use)
72219603    =>  1   ;   //  Fuel / gas for other (chainsaw, tillers, tractor…)
72219604    =>  1   ;   //  Fuel / gas without any details
72219605    =>  1   ;   //  Other fuel / gas for vehicle n e s
72219606    =>  1   ;   //
72219607    =>  1   ;   //
72219608    =>  1   ;   //
72219701    =>  1   ;   //  Oil Outboard
72219702    =>  1   ;   //  Oil Outboard Super
72219703    =>  1   ;   //  Oil Outboard Super, Ob
72219704    =>  1   ;   //  Oil Sae15/40
72219705    =>  1   ;   //  Oil Sae30
72219706    =>  1   ;   //  Oil, Air Compresser
72219707    =>  1   ;   //  Oil, Hydraulic
72219708    =>  1   ;   //  Marine Spray 4 Crc/Wd40
72219709    =>  1   ;   //  Gear Oil
72219710    =>  1   ;   //  Grease Cartridges, Multipurpose
72219711    =>  1   ;   //  Oil Lubricant Ned
72319801    =>  2   ;   //  Service For Maintenance Of Car/Boat, Mechanic
72419901    =>  2   ;   //  Driving lesson
72419902    =>  2   ;   //  Driving license fee
72419903    =>  2   ;   //  Registration fee
72419904    =>  2   ;   //  Other fee in respect of vehicle
72420001    =>  2   ;   //  Car Rental
72420002    =>  2   ;   //  Truck rental, hire
72420003    =>  2   ;   //  Mini bus, van rental, hire
72420004    =>  2   ;   //  Motorcycle / scooter rental, hire
72420005    =>  2   ;   //  Other vehicle rental n e s
73220101    =>  2   ;   //  Bus Fare
73220102    =>  2   ;   //  Taxi Fare
73220103    =>  2   ;   //  Other Transport Fare Ned
73320201    =>  2   ;   //  Air Fare
73320202    =>  2   ;   //  Air fare - domestic
73320203    =>  2   ;   //  Air fare without any detail
73420301    =>  2   ;   //  Boat Fare
73420302    =>  2   ;   //  Boat fare (travel overseas)
73420303    =>  2   ;   //  Other fees related to transport by boat n e s
73520401    =>  2   ;   //  Air fare and boat fare together
73620501    =>  2   ;   //  Freight, Delivery Charge
73620502    =>  2   ;   //  Other services related to transport
81120601    =>  2   ;   //  Payments For Delivery Of Letter, Postcards And Parcels
81120602    =>  2   ;   //  Private Mail And Parcel Delivery
81120603    =>  2   ;   //  Purchases Of Postage Stamps
81120604    =>  2   ;   //  Purchase Of Pre□Franked Postcards And Aerogramme
81120605    =>  2   ;   //  Other Postal Fee
82020701    =>  3   ;   //  Purcahse Of Cell (Mobile) Phones, Smartphone
82020702    =>  3   ;   //  Purchase Of Land □Line Telephones
82020703    =>  3   ;   //  Radio Telephones
82020704    =>  3   ;   //  Telefax Machines
82020705    =>  3   ;   //  Other Telephone And Telefax Equipment
82020801    =>  2   ;   //  Repair Of Such Equipment
83020901    =>  2   ;   //  Connection And Subscription Costs Of Cell (Mobile) Phones
83020902    =>  2   ;   //  Purchase Of Recharge Cards For Cell (Mobile) Phones
83020903    =>  2   ;   //  Purchase Of Sim Card
83020904    =>  2   ;   //  Installation And Subscription Costs Of Land□Line Telephone
Equipment
83020905    =>  2   ;   //  Internet Services
83020906    =>  2   ;   //  Other Telephone Services
91121001    =>  3   ;   //  Car Radios
91121002    =>  3   ;   //  Cd-Players
91121003    =>  3   ;   //  Microphones And Earphones
91121004    =>  3   ;   //  Radio Clocks
91121005    =>  3   ;   //  Radio Sets
91121006    =>  3   ;   //  Stereos And Their Constituent Units Eg Tuners, Amplifiers Etc
91121007    =>  3   ;   //  Television Aerials Of All Types
91121008    =>  3   ;   //  Television Sets
91121009    =>  3   ;   //  Two-Way Radios
91121010    =>  3   ;   //  Video Player/Vcd/Dvd
91121011    =>  3   ;   //  Mp3 Players, Ipods
91121012    =>  3   ;   //  Other Equipment
```

```
91221101    =>  3   ;   //  Cameras
91221102    =>  3   ;   //  Video camera
91221201    =>  3   ;   //  Film And Slide Projectors
91221301    =>  3   ;   //  Accessories for camera (flash, objective, lenses…)
91221401    =>  3   ;   //  Binoculars
91221402    =>  3   ;   //  Compass
91221403    =>  3   ;   //  Other Optical Instrument
91221404    =>  3   ;   //  Telescopes
91221405    =>  3   ;   //  Gps devices
91321501    =>  3   ;   //  Applications
91321502    =>  3   ;   //  Calculators
91321503    =>  3   ;   //  Computer Software Packages, E.G. Operating Systems
91321504    =>  3   ;   //  Personal Computers Desktop
91321505    =>  3   ;   //  Personal Computers Laptop
91321506    =>  3   ;   //  Printers And Miscellaneous Accessories Accompanying Them
91321507    =>  3   ;   //  Tablet
91321508    =>  3   ;   //  Other It Equipment
91421601    =>  4   ;   //  Dvd, Blue Ray
91421602    =>  4   ;   //  Cd Compact Disc
91421603    =>  4   ;   //  Other Recording Media
91421604    =>  4   ;   //  portable harddrives, flash drives
91521701    =>  2   ;   //  Repair Of Audio-Visual, Photographic And Information
Processing Equipment
92121801    =>  3   ;   //  Boat without outboard motor
92121802    =>  3   ;   //  Canoe
92121803    =>  3   ;   //  Outboard Motor
92121805    =>  3   ;   //  Dive gear (tanks, bc, regulators, etc.)
92121807    =>  3   ;   //  Water craft trailer
92121808    =>  3   ;   //  Other major durables for outdoor recreation
92121809    =>  3   ;   //  Boat with outboard motor
92221901    =>  3   ;   //  Guitare
92221902    =>  3   ;   //  Guitar Strings, Steel
92221903    =>  3   ;   //  Other Musical Instrument
92221904    =>  3   ;   //  Other musical instrument n e s
92222001    =>  3   ;   //  Billiard tables
92222002    =>  3   ;   //  Gaming machines
92222003    =>  3   ;   //  Pinball machines
92222004    =>  3   ;   //  Pin-pong table (table-tennis table)
92322101    =>  2   ;   //  Maintenance And Repair Of Other Major Durables For
Recreation And Culture
93122201    =>  4   ;   //  Card Games
93122202    =>  4   ;   //  Chess
93122203    =>  4   ;   //  Parlour Games
93122204    =>  4   ;   //  Video Game Console And Accessories
93122205    =>  4   ;   //  Other Games
93122301    =>  4   ;   //  Barbie Dolls
93122302    =>  4   ;   //  Cars (Police) Remote Control
93122303    =>  4   ;   //  Toys of all kinds (dolls, fireworks, toy cars, etc.)
93122304    =>  4   ;   //  Water Guns
93122305    =>  4   ;   //  X-Mas Lights
93122306    =>  4   ;   //  X-Mas Tree Decoration
93122307    =>  4   ;   //  Baby Dolls With Accessories
93122308    =>  4   ;   //  Other Toys
93122401    =>  4   ;   //  Other items for collection e.g. coins, medals, zoological
and botanical specimens etc.)
93122402    =>  4   ;   //  Stamp collecting requisites (used or cancelled postage
stamps, stamp albums etc.)
93222501    =>  4   ;   //  Beach Volley Ball
93222502    =>  4   ;   //  Cricket Ball
93222503    =>  4   ;   //  Other Sport Equipment
93222504    =>  4   ;   //  Golf clubs
93222505    =>  4   ;   //  Nets
93222506    =>  4   ;   //  Rackets
93222507    =>  4   ;   //  Shuttlecocks
93222508    =>  4   ;   //  Weights
93222509    =>  4   ;   //  Other equipment for sport n e s
93222601    =>  4   ;   //  Parachute and other sky-diving equipment
93222701    =>  4   ;   //  Firearms (gun, bullet…)
93222801    =>  4   ;   //  Fishing Lines
93222802    =>  4   ;   //  Speargun
93222803    =>  4   ;   //  Mask, Flippers
```

```
93222804    =>  4   ;   //  Other Fishing Equipments
93222901    =>  4   ;   //  Frisbee
93222902    =>  4   ;   //  Inflatable Boats
93222903    =>  4   ;   //  Inflatable Swimming Pools
93222904    =>  4   ;   //  Volleyball
93222905    =>  4   ;   //  Other Beach And Open Air Games
93223001    =>  4   ;   //  Air Mattresses And Inflating Pumps
93223002    =>  4   ;   //  Camping Stoves And Barbecues
93223003    =>  4   ;   //  Sleeping Bags
93223004    =>  4   ;   //  Tents And Accessories
93223101    =>  4   ;   //  Football Boots
93223102    =>  4   ;   //  Golfing Shoes
93223103    =>  4   ;   //  Other Such Footwear Fitted With Ice□Skates, Rollers, Spikes,
Studs, Etc.
93223104    =>  4   ;   //  Rugby Boots
93223201    =>  4   ;   //  Belts
93223202    =>  4   ;   //  Body Padding
93223203    =>  4   ;   //  Boxing Gloves
93223204    =>  4   ;   //  Goggles
93223205    =>  4   ;   //  Headgear
93223206    =>  4   ;   //  Life Jackets
93223207    =>  4   ;   //  Shin Guards
93223208    =>  4   ;   //  Supports
93223301    =>  2   ;   //  Repair Of Such Articles
93323401    =>  1   ;   //  Fertilisers, Composts And Garden Peat
93323402    =>  1   ;   //  Horticultural Preparations
93323403    =>  1   ;   //  Other Garden Plants And Flower Stuff
93323404    =>  1   ;   //  Pots and plant holders
93323405    =>  1   ;   //  Other plants related items n e s
93423501    =>  1   ;   //  Purchase of pets (cat, dog,….)
93423601    =>  1   ;   //  Animal Feed
93423602    =>  1   ;   //  Pig feed, pig pellet, grower
93423603    =>  1   ;   //  Chicken feed
93423604    =>  1   ;   //  Other animal feed (including expired food purchased for
animal feed)
93423605    =>  1   ;   //  Other pet-related products (feed, leashes, vet fee, etc.)
93523701    =>  2   ;   //  Veterinary
93523801    =>  2   ;   //  Other Services Eg Training
94123901    =>  2   ;   //  Bowling Alleys
94123902    =>  2   ;   //  Fairgrounds And Amusement Parks
94123903    =>  2   ;   //  Fitness Centres
94123904    =>  2   ;   //  Golf Courses
94123905    =>  2   ;   //  Gymnasia
94123906    =>  2   ;   //  Horse Racing Courses
94123907    =>  2   ;   //  Motor Racing Circuits
94123908    =>  2   ;   //  Pin□Ball Machines (Excluding Game Of Chance)
94123909    =>  2   ;   //  Roundabouts, See□Saws And Other Playground
94123910    =>  2   ;   //  Facilities For Children
94123911    =>  2   ;   //  See□Saws And Playground Facilities For Children
94123912    =>  2   ;   //  Skating Rinks
94123913    =>  2   ;   //  Sports Stadium
94123914    =>  2   ;   //  Squash Courts
94123915    =>  2   ;   //  Swimming Pools
94123916    =>  2   ;   //  Tennis Courts
94123917    =>  2   ;   //  Velodromes
94124001    =>  2   ;   //  Hire of boat
94124002    =>  2   ;   //  Hire of other accessories for sport and recreation
94124101    =>  2   ;   //  Aerobics
94124102    =>  2   ;   //  Chess
94124103    =>  2   ;   //  Dancing
94124104    =>  2   ;   //  Music
94124105    =>  2   ;   //  Swimming
94124106    =>  2   ;   //  Other Group Lessons
94124201    =>  2   ;   //  Services Of Mountain Guides, Tour Guides Etc
94124301    =>  2   ;   //  Navigational Aid Services For Boating
94124401    =>  2   ;   //  Hire of game-specific footwear (excludes purchase (932))
94224501    =>  2   ;   //  Art Galleries And Exhibitions
94224502    =>  2   ;   //  Cinemas
94224503    =>  2   ;   //  Concert And Music Halls
94224504    =>  2   ;   //  Historical Monuments, National Parks, Zoological And
Botanical Gardens
```

```
94224505    =>  2   ;   //  And Aquaria
94224506    =>  2   ;   //  Libraries
94224507    =>  2   ;   //  Theatre
94224508    =>  2   ;   //  Cinema
94224509    =>  2   ;   //  Game At The Stadium
94224510    =>  2   ;   //  Other Services
94224601    =>  2   ;   //  Dvd rental, hire
94224701    =>  2   ;   //  Cable Tv Fee
94224702    =>  2   ;   //  Other Fee Related To Cable Tv
94224801    =>  2   ;   //  Services Of Photographers Eg Film Developing , Print
Processing, Enlarging, Portrait Photography, Wedding Photography
94324901    =>  2   ;   //  Bingo Card
94324902    =>  2   ;   //  Gambling Machine
94324903    =>  2   ;   //  Other Gamble Fee
95125001    =>  4   ;   //  Atlases
95125002    =>  4   ;   //  Dictionaries
95125003    =>  4   ;   //  Encyclopaedias
95125004    =>  4   ;   //  Guidebooks
95125005    =>  4   ;   //  Musical Scores
95125006    =>  4   ;   //  Novels, Story Books
95125007    =>  4   ;   //  Scrapbooks And Albums For Children
95125008    =>  4   ;   //  Textbooks, School Books
95125009    =>  4   ;   //  Other Book Related Expense
95125101    =>  2   ;   //  Bookbinding
95225201    =>  1   ;   //  Magazines
95225202    =>  1   ;   //  Newspapers
95225203    =>  1   ;   //  Periodicals
95325301    =>  1   ;   //  Greeting Cards, Visiting Card
95325302    =>  1   ;   //  Maps And Globes
95325303    =>  1   ;   //  Poster, Post Card…
95425401    =>  1   ;   //  Envelopes
95425402    =>  1   ;   //  Marking Pen
95425403    =>  1   ;   //  Pencil Case
95425404    =>  1   ;   //  Pencil Sharpener
95425405    =>  1   ;   //  Pens, Blue/Black, Ball Point
95425406    =>  1   ;   //  Pens, Blue/Blk/Red, Metal Point, Bic
95425407    =>  1   ;   //  Rubber (Eraser)
95425408    =>  1   ;   //  Carpenter’S Pencil
95425409    =>  1   ;   //  Carpenter'S Folding Ruler
95425410    =>  1   ;   //  Chalk Line Refill Blue Johnson
95425411    =>  1   ;   //  Crepe Paper
95425412    =>  1   ;   //  Dual Metric/Imperial 8/10M
95425413    =>  1   ;   //  Gift Wrapper
95425414    =>  1   ;   //  Sellotape, 2" Clear
95425415    =>  1   ;   //  Tialeniu School Badges-Jan. 2011
95425416    =>  1   ;   //  Other Stationery Or Ned
95425501    =>  1   ;   //  Ink Cartridge Brother
95425601    =>  1   ;   //  Exercise book
96025701    =>  2   ;   //  All Inclusive Holidays Or Tours Which Provide For Travel,
Food, Accommodation, Guides Etc
101025801   =>  2   ;   //  Kindergarden Fees
101025802   =>  2   ;   //  Primary School Fees
102025901   =>  2   ;   //  Secondary School Fees
103026001   =>  2   ;   //  Technical Education School Fees
103026002   =>  2   ;   //  Vocational Education School Fees
104026101   =>  2   ;   //  University Fees
105026201   =>  2   ;   //  Other Educational Program Fees
105026202   =>  2   ;   //  Tutoring for academic subjects (science….), private lesson
for school purposes
105026203   =>  2   ;   //  Other expenditure related to education (teacher day, sport
day, school events…)
111126301   =>  2   ;   //  Restaurant, Café
111126302   =>  2   ;   //  Bar Night Clubs
111126401   =>  2   ;   //  Ham, Sandwich, Local
111126402   =>  2   ;   //  Ham, Sandwich, Us
111126403   =>  2   ;   //  Other Take Away Food
111126501   =>  2   ;   //  Take away ice cream
111126502   =>  2   ;   //  Take away pizza
111126503   =>  2   ;   //  Cooked poultry, chicken take away
111126504   =>  2   ;   //  Fish and chips take away
111126505   =>  2   ;   //  Burger, cheese burger take away
```

```
111126506   =>  2   ;   //  Fried chicken take away
111126507   =>  2   ;   //  Meat pies take away
111126508   =>  2   ;   //  Sandwich take away
111126598   =>  2   ;   //  Other take away plate, take away food (chinese food,
chicken, barbecue, cassava, fish, sashimi ….)
111126599   =>  2   ;   //  Take away food without any detail
111126601   =>  2   ;   //  The sale of cooked dishes by catering contractors whether
collected by the customer or delivered to the customer’s home
111226701   =>  2   ;   //  School restaurant
112026801   =>  2   ;   //  Hotel, Motel, Guest House
112026802   =>  2   ;   //  Packaged accomidation with food (hotel)
112026901   =>  2   ;   //  Boarding Schools, Univ Compound
112027001   =>  2   ;   //  Accommodation services provided by others
121127101   =>  2   ;   //  Hairdresser
121127201   =>  2   ;   //  Body Care, Depilation (Waxing, Threading, Etc.) And The Like
121127202   =>  2   ;   //  Facials
121127203   =>  2   ;   //  Manicure
121127204   =>  2   ;   //  Non□Medical Massage
121127205   =>  2   ;   //  Pedicure
121127206   =>  2   ;   //  Turkish Baths, Saunas
121127207   =>  2   ;   //  Other Personal Grooming
121127208   =>  2   ;   //  Tattoo, piercings and other body art
121227301   =>  4   ;   //  Curling Tongs And Styling Combs
121227302   =>  4   ;   //  Electric Razors And Hair Trimmers
121227303   =>  4   ;   //  Electric Toothbrushes
121227304   =>  4   ;   //  Hand□Held And Hood Hair Dryers
121227305   =>  4   ;   //  Sunlamps
121227306   =>  4   ;   //  Other Electric Appliance For Personal Care
121227401   =>  2   ;   //  Repair Of Such Appliances
121327501   =>  1   ;   //  Razor Blades
121327502   =>  1   ;   //  Razor Refill, Gillette
121327503   =>  1   ;   //  Razor, Contour
121327504   =>  1   ;   //  Tooth Brush
121327505   =>  1   ;   //  Hair Brush
121327506   =>  1   ;   //  Nail Brush
121327507   =>  1   ;   //  Nail Files
121327508   =>  1   ;   //  Personal Scale, Baby Scale
121327509   =>  1   ;   //  Combs
121327510   =>  1   ;   //  Other Non Electric Appliance For Personal Care
121327601   =>  1   ;   //  Shampoo, Baby Shampoo
121327602   =>  1   ;   //  Hair Conditioner, Palmolive
121327603   =>  1   ;   //  Hair Gel, (Mousse)
121327604   =>  1   ;   //  Hair Shampoo, Palmolive
121327605   =>  1   ;   //  Shaving Cream, Nivea
121327606   =>  1   ;   //  Soap, Lux
121327607   =>  1   ;   //  Soap, Palmolive
121327608   =>  1   ;   //  Soap, Protex
121327609   =>  1   ;   //  Body Wash
121327610   =>  1   ;   //  Body Wash, Lux
121327611   =>  1   ;   //  Body Wash, Protex
121327612   =>  1   ;   //  Body Wash, Suave
121327613   =>  1   ;   //  Toothpaste, Colgate
121327614   =>  1   ;   //  Cleaning Milk
121327615   =>  1   ;   //  Soap Ned
121327616   =>  1   ;   //  Other Articles Of Personal Hygiene
121327701   =>  1   ;   //  Baby Oil, 200Mls, Johnson
121327702   =>  1   ;   //  Baby Powder
121327703   =>  1   ;   //  Baby Powder, Johnsons
121327704   =>  1   ;   //  Body Lotion, Nivea
121327705   =>  1   ;   //  Deodorant Lynx Spray, Men
121327706   =>  1   ;   //  Deodorant Lynx Spray, Women
121327707   =>  1   ;   //  Deodorant Mum Roll-On, Dove (For Ladies)
121327708   =>  1   ;   //  Deodorant Mum Roll-On, Rexona, (For Ladies)
121327709   =>  1   ;   //  Deodorant Mum Roll-On, Rexona,(For Men)
121327710   =>  1   ;   //  Deodorant Spray, (Men), Rexona
121327711   =>  1   ;   //  Deodorant Spray, (Women), Rexona
121327712   =>  1   ;   //  Deodorant Spray, For Men, Nivea
121327713   =>  1   ;   //  Deodorant Spray, For Women, Nivea
121327714   =>  1   ;   //  Deodorant, Impulse
121327715   =>  1   ;   //  Deodorant, Lynx Roll-On, Men
121327716   =>  1   ;   //  Perfume
```

```
121327717   =>  1   ;   //  Make Up, Lipstick
121327718   =>  1   ;   //  Other Beauty Product
121327801   =>  1   ;   //  Baby Wipes, Huggies
121327802   =>  1   ;   //  Cotton Buds, Ear Buds
121327803   =>  1   ;   //  Ladies Pads, Johnsons
121327804   =>  1   ;   //  Ladies Pads, Libra
121327805   =>  1   ;   //  Ladies Pads, Modess
121327806   =>  1   ;   //  Ladies Pads, Stay Free
121327807   =>  1   ;   //  Nappies, Huggies
121327808   =>  1   ;   //  Nappies, Soft Love
121327809   =>  1   ;   //  Toilet Paper, Sofan
121327810   =>  1   ;   //  Toilet Paper, Swan
121327811   =>  1   ;   //  Toilet Paper, Wondersoft
121327812   =>  1   ;   //  Cotton
121327813   =>  1   ;   //  Nappies for adults, Depends
121327814   =>  1   ;   //  Other Product
122027901   =>  2   ;   //  Services provided by prostitutes and the like
123128001   =>  3   ;   //  Diamon, gold, platnum, and similar precious metals and stones
123128101   =>  3   ;   //  Jewellery (local or not): earing, necklace, bracelet
123128102   =>  3   ;   //  Artificial flowers, artificial tiare
123128103   =>  3   ;   //  Jewellery without any detail
123128104   =>  3   ;   //  Fresh flower arrangments in bouquet, lei, ei, head ea, ea
katu, poe tiare
123128201   =>  3   ;   //  Clocks
123128202   =>  3   ;   //  Watch
123128203   =>  3   ;   //  Alarm Clock
123128204   =>  3   ;   //  Other Clock And Watch Related Article
123128301   =>  2   ;   //  Repair Of Such Articles
123228401   =>  4   ;   //  Handbag
123228402   =>  4   ;   //  Suit Case
123228403   =>  4   ;   //  School Backpacks, (Boys)
123228404   =>  4   ;   //  School Backpacks, (Girls)
123228405   =>  4   ;   //  Other Travel Goods
123228501   =>  4   ;   //  Baby Carriages, Carrycots
123228502   =>  4   ;   //  Car Beds And Seats
123228503   =>  4   ;   //  Pushchairs
123228504   =>  4   ;   //  Reins And Harnesses
123228505   =>  4   ;   //  Other Articles For Babies
123228601   =>  4   ;   //  Lighter
123228602   =>  4   ;   //  Pipe
123228603   =>  4   ;   //  Ashtray
123228604   =>  4   ;   //  Other Article For Smokers
123228701   =>  4   ;   //  Umbrella
123228702   =>  4   ;   //  Umbrella (Beach)
123228703   =>  4   ;   //  Fan
123228704   =>  4   ;   //  Sunglasses
123228705   =>  4   ;   //  Fey Ring
123228706   =>  4   ;   //  Other Miscellaneous Personal Article
123228707   =>  4   ;   //  traditional crafts - fan
123228708   =>  4   ;   //  traditional crafts - handbags, bags, pouches
123228709   =>  4   ;   //  traditional crafts - hats
123228710   =>  4   ;   //  traditional crafts - mats
123228801   =>  4   ;   //  Funerary Articles Eg Coffins, Gravestones, Urns
123228901   =>  2   ;   //  Repair Of Such Articles
124029001   =>  2   ;   //  Retirement Homes For Elderly Persons And Residences For
Disabled Persons
124029101   =>  2   ;   //  Home help, day care and rehabilitation
125129201   =>  2   ;   //  Life Assurance
125129202   =>  2   ;   //  Education Assurance
125229301   =>  2   ;   //  Home Insurance
125329401   =>  2   ;   //  Health Insurance
125429501   =>  2   ;   //  Car Insurance
125429502   =>  2   ;   //  Boat Insurance
125429503   =>  2   ;   //  Other Insurance Related To Transport
125529601   =>  2   ;   //  Personal Insurance
126129701   =>  2   ;   //  Financial intermediation services indirectly measured
126229801   =>  2   ;   //  Banking Fees
126229802   =>  2   ;   //  Money Transfer Fees
126229803   =>  2   ;   //  Credit Card Fee
126229804   =>  2   ;   //  Other Financial Fees
127029901   =>  2   ;   //  Passport Fees
```

```
    127029902   =>  2   ;   //  Administrative Document, Certificate (Birth, Death…)
    127029903   =>  2   ;   //  Photocopies
    127029904   =>  2   ;   //  Fees related to the purchase of a house (appraisal, realty,
    etc.)
    127029905   =>  2   ;   //  Legal expenses
    127029907   =>  2   ;   //  Other fees related to personal services
    901100101-999999999 =>  5   ;   //  Current cash transfers: birthdays
endrecode;


if HG_COICOP in 1:970000000 and HG_MULTIPLIER > 1 then
    numeric tmp2;
    numeric tmp = noccurs(EXPENDITURE_EDT)+1;
    insert(EXPENDITURE_EDT(tmp));

    E_LINE_NUMBER(tmp) = tmp;
    E_HIES_CODE(tmp) = HG_HIES_CODE;
    E_COICOP(tmp) = HG_COICOP;
    E_AMOUNT(tmp) = HG_AMOUNT;
    E_AN_AMT(tmp) = HG_AN_AMT;
    E_WGHT_AN_AMT(tmp) = HG_WGHT_AN_AMT;
    E_MULTIPLIER(tmp) = HG_MULTIPLIER;
    E_LOCATION(tmp) = HG_LOCATION;
    E_EXP_TYPE(tmp) = HG_EXP_TYPE;
    E_PROD_TYPE(tmp) = HG_PROD_TYPE;
    E_NUMBER(tmp) = HG_QUANTITY;
    E_UNIT(tmp) = HG_UNIT;
    E_CATEGORY(tmp) = notappl;
    E_BENEFICIARY(tmp) = 52;
    E_DURABILITY(tmp) = tmpDura;
    E_DESCRIPTION(tmp) = "[diary home given]"+tolower(HG_DESCRIPTION);



    // INCOME SIDE
    numeric tmpCOICOP = HG_COICOP;
    numeric tmpPACCOI = notappl; // using code below
    numeric tmpHIEScode = HG_HIES_CODE; // using code below
    numeric tmpMulti = HG_MULTIPLIER;
    numeric tmpISCO = notappl; // not necessary for this section
    numeric tmpISIC = notappl; // not necessary for this seciton
    numeric tmpValue = HG_AMOUNT;
    numeric tmpIncomeType = HG_EXP_TYPE; //[1]cash  [2]home  [3]in-kind bought  [4]in-kind
    home  [5]rent  [6]intermediate  [7]in-kind income
    numeric tmpLocation = HG_LOCATION;
    numeric tmpBeneficiary = 52;
    string tmpDescription = "[diary home given]"+tolower(HG_DESCRIPTION);
    numeric tmpProductType = HG_PROD_TYPE;
    numeric tmpQuantity = notappl;
    numeric tmpNumber = HG_QUANTITY;
    numeric tmpUnit = HG_UNIT;
    numeric tmpCategory = notappl;
    numeric tmpDurability = tmpDura;

    tmp = noccurs(INCOME_EDT)+1;
    insert(INCOME_EDT(tmp));

    I_LINE_NUMBER(tmp) = tmp;
    I_HIES_CODE(tmp) = tmpHIEScode;
    I_PACCOI(tmp) = tmpPACCOI;
    I_COICOP(tmp) = tmpCOICOP;
    I_GROSS(tmp) = tmpValue;
    I_AN_GROSS(tmp) = I_GROSS(tmp) * tmpMulti;
    I_WGHT_AN_GROSS(tmp) = I_AN_GROSS(tmp) * WEIGHT;
    I_MULTIPLIER(tmp) = tmpMulti;
    I_LOCATION(tmp) = tmpLocation;
    I_INCOME_TYPE(tmp) = tmpIncomeType;
    I_PROD_TYPE(tmp) = tmpProductType;
    I_QUANTITY(tmp) = tmpQuantity;
    I_NUMBER(tmp) = tmpNumber;
    I_UNIT(tmp) = tmpUnit;
```

```
    I_CATEGORY(tmp) = tmpCategory;
    I_BENEFICIARY(tmp) = tmpBeneficiary;
    I_DURABILITY(tmp) = tmpDurability;
    I_INDUSTRY(tmp) = tmpISIC;
    I_OCCUPATION(tmp) = tmpISCO;
    I_DESCRIPTION(tmp) = tmpDescription;

endif;



PROC I_PACCOI
// COICOP to PACCOI map
if I_COICOP in 11704517 then if I_INCOME_TYPE in 1 then I_PACCOI = 13101;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13201;elseif I_INCOME_TYPE in 4 then I_PACCOI =53101;
endif; endif;
if I_COICOP in 11704523 then if I_INCOME_TYPE in 1 then I_PACCOI = 13101;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13201;elseif I_INCOME_TYPE in 4 then I_PACCOI =53101;
endif; endif;
if I_COICOP in 11704514 then if I_INCOME_TYPE in 1 then I_PACCOI = 13102;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13202;elseif I_INCOME_TYPE in 4 then I_PACCOI =53102;
endif; endif;
if I_COICOP in 11704511 then if I_INCOME_TYPE in 1 then I_PACCOI = 13105;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13205;elseif I_INCOME_TYPE in 4 then I_PACCOI =53105;
endif; endif;
if I_COICOP in 11704518 then if I_INCOME_TYPE in 1 then I_PACCOI = 13109;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13209;elseif I_INCOME_TYPE in 4 then I_PACCOI =53109;
endif; endif;
if I_COICOP in 11704524 then if I_INCOME_TYPE in 1 then I_PACCOI = 13109;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13209;elseif I_INCOME_TYPE in 4 then I_PACCOI =53109;
endif; endif;
if I_COICOP in 11704525 then if I_INCOME_TYPE in 1 then I_PACCOI = 13109;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13209;elseif I_INCOME_TYPE in 4 then I_PACCOI =53109;
endif; endif;
if I_COICOP in 11704533 then if I_INCOME_TYPE in 1 then I_PACCOI = 13109;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13209;elseif I_INCOME_TYPE in 4 then I_PACCOI =53109;
endif; endif;
if I_COICOP in 11704603 then if I_INCOME_TYPE in 1 then I_PACCOI = 13110;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13210;elseif I_INCOME_TYPE in 4 then I_PACCOI =53110;
endif; endif;
if I_COICOP in 11704607 then if I_INCOME_TYPE in 1 then I_PACCOI = 13110;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13210;elseif I_INCOME_TYPE in 4 then I_PACCOI =53110;
endif; endif;
if I_COICOP in 11704601 then if I_INCOME_TYPE in 1 then I_PACCOI = 13113;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13213;elseif I_INCOME_TYPE in 4 then I_PACCOI =53113;
endif; endif;
if I_COICOP in 11704604 then if I_INCOME_TYPE in 1 then I_PACCOI = 13114;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13214;elseif I_INCOME_TYPE in 4 then I_PACCOI =53114;
endif; endif;
if I_COICOP in 11604209 then if I_INCOME_TYPE in 1 then I_PACCOI = 13116;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13216;elseif I_INCOME_TYPE in 4 then I_PACCOI =53116;
endif; endif;
if I_COICOP in 11604210 then if I_INCOME_TYPE in 1 then I_PACCOI = 13117;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13217;elseif I_INCOME_TYPE in 4 then I_PACCOI =53117;
endif; endif;
if I_COICOP in 11604207 then if I_INCOME_TYPE in 1 then I_PACCOI = 13118;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13218;elseif I_INCOME_TYPE in 4 then I_PACCOI =53118;
endif; endif;
if I_COICOP in 11604202 then if I_INCOME_TYPE in 1 then I_PACCOI = 13119;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13219;elseif I_INCOME_TYPE in 4 then I_PACCOI =53119;
endif; endif;
if I_COICOP in 11604205 then if I_INCOME_TYPE in 1 then I_PACCOI = 13119;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13219;elseif I_INCOME_TYPE in 4 then I_PACCOI =53119;
endif; endif;
if I_COICOP in 11604206 then if I_INCOME_TYPE in 1 then I_PACCOI = 13119;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13219;elseif I_INCOME_TYPE in 4 then I_PACCOI =53119;
endif; endif;
if I_COICOP in 11604219 then if I_INCOME_TYPE in 1 then I_PACCOI = 13120;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13220;elseif I_INCOME_TYPE in 4 then I_PACCOI =53120;
endif; endif;
if I_COICOP in 11604225 then if I_INCOME_TYPE in 1 then I_PACCOI = 13121;elseif
```

```
I_INCOME_TYPE in 2 then I_PACCOI = 13221;elseif I_INCOME_TYPE in 4 then I_PACCOI =53121;
endif; endif;
if I_COICOP in 11604233 then if I_INCOME_TYPE in 1 then I_PACCOI = 13124;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13224;elseif I_INCOME_TYPE in 4 then I_PACCOI =53124;
endif; endif;
if I_COICOP in 11504004 then if I_INCOME_TYPE in 1 then I_PACCOI = 13130;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13230;elseif I_INCOME_TYPE in 4 then I_PACCOI =53130;
endif; endif;
if I_COICOP in 11504005 then if I_INCOME_TYPE in 1 then I_PACCOI = 13130;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13230;elseif I_INCOME_TYPE in 4 then I_PACCOI =53130;
endif; endif;
if I_COICOP in 11604232 then if I_INCOME_TYPE in 1 then I_PACCOI = 13131;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13231;elseif I_INCOME_TYPE in 4 then I_PACCOI =53131;
endif; endif;
if I_COICOP in 11302227 then if I_INCOME_TYPE in 1 then I_PACCOI = 13301;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13401;elseif I_INCOME_TYPE in 4 then I_PACCOI =53201;
endif; endif;
if I_COICOP in 11302231 then if I_INCOME_TYPE in 1 then I_PACCOI = 13301;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13401;elseif I_INCOME_TYPE in 4 then I_PACCOI =53201;
endif; endif;
if I_COICOP in 11302237 then if I_INCOME_TYPE in 1 then I_PACCOI = 13301;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13401;elseif I_INCOME_TYPE in 4 then I_PACCOI =53201;
endif; endif;
if I_COICOP in 11302228 then if I_INCOME_TYPE in 1 then I_PACCOI = 13302;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13402;elseif I_INCOME_TYPE in 4 then I_PACCOI =53202;
endif; endif;
if I_COICOP in 11302229 then if I_INCOME_TYPE in 1 then I_PACCOI = 13302;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13402;elseif I_INCOME_TYPE in 4 then I_PACCOI =53202;
endif; endif;
if I_COICOP in 11302232 then if I_INCOME_TYPE in 1 then I_PACCOI = 13303;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13403;elseif I_INCOME_TYPE in 4 then I_PACCOI =53203;
endif; endif;
if I_COICOP in 11302240 then if I_INCOME_TYPE in 1 then I_PACCOI = 13304;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13404;elseif I_INCOME_TYPE in 4 then I_PACCOI =53204;
endif; endif;
if I_COICOP in 11302239 then if I_INCOME_TYPE in 1 then I_PACCOI = 13305;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13405;elseif I_INCOME_TYPE in 4 then I_PACCOI =53205;
endif; endif;
if I_COICOP in 11302234 then if I_INCOME_TYPE in 1 then I_PACCOI = 13306;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13406;elseif I_INCOME_TYPE in 4 then I_PACCOI =53206;
endif; endif;
if I_COICOP in 11302201 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302208 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302212 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302217 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302224 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302241 then if I_INCOME_TYPE in 1 then I_PACCOI = 13309;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13409;elseif I_INCOME_TYPE in 4 then I_PACCOI =53209;
endif; endif;
if I_COICOP in 11302205 then if I_INCOME_TYPE in 1 then I_PACCOI = 13310;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13410;elseif I_INCOME_TYPE in 4 then I_PACCOI =53210;
endif; endif;
if I_COICOP in 11302213 then if I_INCOME_TYPE in 1 then I_PACCOI = 13310;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13410;elseif I_INCOME_TYPE in 4 then I_PACCOI =53210;
endif; endif;
if I_COICOP in 11302216 then if I_INCOME_TYPE in 1 then I_PACCOI = 13310;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13410;elseif I_INCOME_TYPE in 4 then I_PACCOI =53210;
endif; endif;
if I_COICOP in 11302233 then if I_INCOME_TYPE in 1 then I_PACCOI = 13310;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13410;elseif I_INCOME_TYPE in 4 then I_PACCOI =53210;
endif; endif;
```

```
if I_COICOP in 11302225 then if I_INCOME_TYPE in 1 then I_PACCOI = 13311;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13411;elseif I_INCOME_TYPE in 4 then I_PACCOI =53211;
endif; endif;
if I_COICOP in 11302238 then if I_INCOME_TYPE in 1 then I_PACCOI = 13312;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13412;elseif I_INCOME_TYPE in 4 then I_PACCOI =53212;
endif; endif;
if I_COICOP in 11302302 then if I_INCOME_TYPE in 1 then I_PACCOI = 13315;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13415;elseif I_INCOME_TYPE in 4 then I_PACCOI =53215;
endif; endif;
if I_COICOP in 11302310 then if I_INCOME_TYPE in 1 then I_PACCOI = 13319;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13419;elseif I_INCOME_TYPE in 4 then I_PACCOI =53219;
endif; endif;
if I_COICOP in 11201203 then if I_INCOME_TYPE in 1 then I_PACCOI = 13503;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13603;elseif I_INCOME_TYPE in 4 then I_PACCOI =53303;
endif; endif;
if I_COICOP in 11201511 then if I_INCOME_TYPE in 1 then I_PACCOI = 13508;elseif
I_INCOME_TYPE in 2 then I_PACCOI = 13608;elseif I_INCOME_TYPE in 4 then I_PACCOI =53308;
endif; endif;


if I_INCOME_TYPE in 3 then
    I_PACCOI = 52100+int(I_COICOP / 10000000);
endif;

recode  I_HIES_CODE =>  I_PACCOI    ;
    202.018 =>  62000   ;
    202.019 =>  61000   ;
    204.311 =>  52104   ;
    204.331 =>  52104   ;
    204.332 =>  52104   ;
    204.351 =>  52104   ;
    205.101 =>  52105   ;
    205.201 =>  52105   ;
    206.502 =>  71305   ;
    206.504 =>  52107   ;
    207.201 =>  52107   ;
    207.202 =>  52107   ;
    207.203 =>  52111   ;
    207.204 =>  52111   ;
    207.206 =>  52107   ;
    208.102 =>  52105   ;
    208.201 =>  52112   ;
    208.203 =>  52107   ;
    208.204 =>  52107   ;
    208.206 =>  52112   ;
    208.207 =>  52112   ;
    212.101 =>  52112   ;
    213.004 =>  52112   ;
    301.202 =>  52110   ;
    302.006 =>  52106   ;
    303.001 =>  52103   ;
    303.004 =>  52103   ;
    303.006 =>  52103   ;
    306.003 =>  52102   ;
    306.005 =>  52102   ;
endrecode;
recode  I_HIES_CODE =>  I_INCOME_TYPE   ;
    202.018 =>  5   ;
    202.019 =>  5   ;
    204.311 =>  3   ;
    204.331 =>  3   ;
    204.332 =>  3   ;
    204.351 =>  3   ;
    205.101 =>  3   ;
    205.201 =>  3   ;
    206.502 =>  6   ;
    206.504 =>  3   ;
    207.201 =>  3   ;
    207.202 =>  3   ;
    207.203 =>  3   ;
    207.204 =>  3   ;
```

```
    207.206 =>  3   ;
    208.102 =>  3   ;
    208.201 =>  3   ;
    208.203 =>  3   ;
    208.204 =>  3   ;
    208.206 =>  3   ;
    208.207 =>  3   ;
    212.101 =>  3   ;
    213.004 =>  3   ;
    301.202 =>  3   ;
    302.006 =>  3   ;
    303.001 =>  3   ;
    303.004 =>  3   ;
    303.006 =>  3   ;
    306.003 =>  3   ;
    306.005 =>  3   ;
endrecode;
recode  I_HIES_CODE =>  I_PACCOI    ;
    401.113 - 401.197   =>  11101   ;
    401.213 - 401.294   =>  11102   ;
    401.484 - 401.494   =>  11298   ;
    401.531 - 401.585   =>  11104   ;
    401.601 - 401.684   =>  11105   ;
endrecode;
recode  I_HIES_CODE =>  I_INCOME_TYPE   ;
    401.113 - 401.197   =>  1   ;
    401.213 - 401.294   =>  1   ;
    401.484 - 401.494   =>  7   ;
    401.531 - 401.585   =>  1   ;
    401.601 - 401.684   =>  1   ;
endrecode;
recode  I_HIES_CODE =>  I_PACCOI    ;
    402.003 =>  13103   ;
    402.004 =>  13104   ;
    402.016 =>  13116   ;
    402.019 =>  13119   ;
    402.021 =>  13121   ;
    402.901 =>  71201   ;
    402.906 =>  71299   ;
    403.001 =>  13301   ;
    403.002 =>  13302   ;
    403.003 =>  13303   ;
    403.004 =>  13304   ;
    403.005 =>  13305   ;
    403.901 =>  71301   ;
    403.902 =>  71302   ;
    403.904 =>  71304   ;
    403.905 =>  71399   ;
    404.001 =>  13503   ;
    404.002 =>  13508   ;
    404.901 =>  71401   ;
    404.902 =>  71402   ;
    404.903 =>  71403   ;
    404.907 =>  71499   ;
    405.001 =>  13701   ;
    405.002 =>  13702   ;
    405.003 =>  13703   ;
    405.004 =>  13704   ;
    405.006 =>  13706   ;
    405.008 =>  13708   ;
    405.009 =>  13709   ;
    405.010 =>  13710   ;
    405.901 =>  71501   ;
    405.902 =>  71502   ;
    405.903 =>  71503   ;
    405.904 =>  71599   ;
    406.111 =>  12100   ;
    406.147 =>  12100   ;
    406.156 =>  12100   ;
    406.902 =>  71102   ;
    406.903 =>  71103   ;
```

```
    406.905 =>  71105   ;
    406.907 =>  71107   ;
    406.911 =>  71112   ;
    406.913 =>  71113   ;
    407.011 =>  21000   ;
    407.012 =>  22000   ;
    407.013 =>  23000   ;
    407.021 =>  31000   ;
    407.022 =>  32000   ;
    407.023 =>  33000   ;
    407.024 =>  35000   ;
    407.025 =>  39000   ;
    407.031 =>  42000   ;
    408.003 =>  51103   ;
    408.005 =>  51105   ;
    408.006 =>  51199   ;
endrecode;
recode  I_HIES_CODE =>  I_INCOME_TYPE   ;
    402.003 =>  1   ;
    402.004 =>  1   ;
    402.016 =>  1   ;
    402.019 =>  1   ;
    402.021 =>  1   ;
    402.901 =>  6   ;
    402.906 =>  6   ;
    403.001 =>  1   ;
    403.002 =>  1   ;
    403.003 =>  1   ;
    403.004 =>  1   ;
    403.005 =>  1   ;
    403.901 =>  6   ;
    403.902 =>  6   ;
    403.904 =>  6   ;
    403.905 =>  6   ;
    404.001 =>  1   ;
    404.002 =>  1   ;
    404.901 =>  6   ;
    404.902 =>  6   ;
    404.903 =>  6   ;
    404.907 =>  6   ;
    405.001 =>  1   ;
    405.002 =>  1   ;
    405.003 =>  1   ;
    405.004 =>  1   ;
    405.006 =>  1   ;
    405.008 =>  1   ;
    405.009 =>  1   ;
    405.010 =>  1   ;
    405.901 =>  6   ;
    405.902 =>  6   ;
    405.903 =>  6   ;
    405.904 =>  6   ;
    406.111 =>  1   ;
    406.147 =>  1   ;
    406.156 =>  1   ;
    406.902 =>  6   ;
    406.903 =>  6   ;
    406.905 =>  6   ;
    406.907 =>  6   ;
    406.911 =>  6   ;
    406.913 =>  6   ;
    407.011 =>  1   ;
    407.012 =>  1   ;
    407.013 =>  1   ;
    407.021 =>  1   ;
    407.022 =>  1   ;
    407.023 =>  1   ;
    407.024 =>  1   ;
    407.025 =>  1   ;
    407.031 =>  1   ;
    408.003 =>  1   ;
```

```
    408.005 =>  1   ;
    408.006 =>  1   ;
endrecode;
PROC I_MULTIPLIER
if I_COICOP in 11100101:127029905,901100101:901100706,908108001 and !(I_HIES_CODE in
400:490) then
    if I_COICOP in 11100101:12207506 then
        if $ in 26 then
            impute($, 26.07142857);
        else
            impute($, 52.14285714);
        endif;
    else
        do varying i = 1 until i >= 316
            if I_COICOP in Selection(i,2) then
                break;
            endif
        enddo;

        {
        HIES ID Items week 1    Items week 2    Total both weeks    W1/W2 ratio Source
        Multiplier
        308     1               18              19                  0.1         Week 2  52
        318     2               22              24                  0.1         Week 2  52
        118     19              6               25                  3.2         Week 1  52
        326     42              13              55                  3.2         Week 1  52
        250     45              11              56                  4.1         Week 1  52
        134     55              3               58                  18.3        Week 1  52
        105     31                              31                  0.0         Week 1  52

        week1 = P_DAY in 1:7
        week2 = P_DAY in 8:14
        }

        numeric Source = Selection(i,3);
        numeric TwoD = Selection(i,4);
        numeric OneD = Selection(i,5);
        numeric Mod = Selection(i,6);
        numeric HHOneDiary = 0;

        if ID07 in 308,318,118,326,250,134,105 then
            HHoneDiary = 1;
        endif;

        if Source in 1:3 then
            if Source = 1 then
                if I_HIES_CODE >= 500 then
                    if HHOneDiary then
                        I_MULTIPLIER = OneD;
                        errmsg("source1:One Diary") summary;
                    else
                        I_MULTIPLIER = TwoD;
                        errmsg("source1:Two Diary") summary;
                    endif;
                else
                    I_MULTIPLIER = 0;
                    errmsg("source1:Module(0)") summary;
                endif;
            endif;
            if Source = 2 then
                if I_HIES_CODE < 500 then
                    I_MULTIPLIER = Mod;
                    errmsg("source2:Module") summary;
                else
                    I_MULTIPLIER = 0;
                    errmsg("source2:Diary(0)") summary;
                endif;
            endif;
            if Source = 3 then
                if HHOneDiary then
                    I_MULTIPLIER = OneD;
```

```
                    errmsg("source3:One Diary") summary;
                else
                    I_MULTIPLIER = TwoD;
                    errmsg("source3:Two Diary") summary;
                endif;
            endif;
        else
            errmsg("SOURCE/COICOP not found: %d",I_COICOP);
        endif;
    endif;
endif;

recode  ID07    :   I_LINE_NUMBER   :   I_HIES_CODE =>  I_MULTIPLIER    ;
        135 :   5   :   401.184 =>  26  ;
        148 :   4   :   401.135 =>  26  ;
        147 :   3   :   401.184 =>  26  ;
        148 :   3   :   401.135 =>  26  ;
        141 :   3   :   401.135 =>  26  ;
        141 :   4   :   401.185 =>  26  ;
        158 :   2   :   401.184 =>  26  ;
        135 :   3   :   401.185 =>  26  ;
        155 :   4   :   401.135 =>  26  ;
        135 :   4   :   401.184 =>  26  ;
        154 :   3   :   401.186 =>  26  ;
        159 :   3   :   401.141 =>  26  ;
        121 :   5   :   401.184 =>  26  ;
        154 :   4   :   401.184 =>  26  ;
endrecode;

I_AN_GROSS = I_GROSS * I_MULTIPLIER;
I_WGHT_AN_GROSS = I_AN_GROSS * WEIGHT;
PROC I_CATEGORY
recode  I_PACCOI    =>  I_CATEGORY  ;
    11101   =>  1011    ;
    11102   =>  1011    ;
    11103   =>  1011    ;
    11104   =>  1011    ;
    11105   =>  1011    ;
    11201   =>  7011    ;
    11202   =>  7011    ;
    11203   =>  7011    ;
    11204   =>  7011    ;
    11205   =>  7011    ;
    11206   =>  7011    ;
    11207   =>  7011    ;
    11208   =>  7011    ;
    11209   =>  7011    ;
    11210   =>  7011    ;
    11211   =>  7011    ;
    11212   =>  7011    ;
    11298   =>  7011    ;
    11299   =>  7011    ;
    12100   =>  1012    ;
    13101   =>  1013    ;
    13102   =>  1013    ;
    13103   =>  1013    ;
    13104   =>  1013    ;
    13105   =>  1013    ;
    13106   =>  1013    ;
    13107   =>  1013    ;
    13108   =>  1013    ;
    13109   =>  1013    ;
    13110   =>  1013    ;
    13111   =>  1013    ;
    13112   =>  1013    ;
    13113   =>  1013    ;
    13114   =>  1013    ;
    13115   =>  1013    ;
    13116   =>  1013    ;
    13117   =>  1013    ;
    13118   =>  1013    ;
```

```
13119   =>  1013    ;
13120   =>  1013    ;
13121   =>  1013    ;
13122   =>  1013    ;
13123   =>  1013    ;
13124   =>  1013    ;
13125   =>  1013    ;
13126   =>  1013    ;
13127   =>  1013    ;
13128   =>  1013    ;
13129   =>  1013    ;
13130   =>  1013    ;
13131   =>  1013    ;
13132   =>  1013    ;
13133   =>  1013    ;
13134   =>  1013    ;
13199   =>  1013    ;
13201   =>  2017    ;
13202   =>  2017    ;
13203   =>  2017    ;
13204   =>  2017    ;
13205   =>  2017    ;
13206   =>  2017    ;
13207   =>  2017    ;
13208   =>  2017    ;
13209   =>  2017    ;
13210   =>  2017    ;
13211   =>  2017    ;
13212   =>  2017    ;
13213   =>  2017    ;
13214   =>  2017    ;
13215   =>  2017    ;
13216   =>  2017    ;
13217   =>  2017    ;
13218   =>  2017    ;
13219   =>  2017    ;
13220   =>  2017    ;
13221   =>  2017    ;
13222   =>  2017    ;
13223   =>  2017    ;
13224   =>  2017    ;
13225   =>  2017    ;
13226   =>  2017    ;
13227   =>  2017    ;
13228   =>  2017    ;
13229   =>  2017    ;
13230   =>  2017    ;
13231   =>  2017    ;
13232   =>  2017    ;
13233   =>  2017    ;
13298   =>  2017    ;
13299   =>  2017    ;
13301   =>  1014    ;
13302   =>  1014    ;
13303   =>  1014    ;
13304   =>  1014    ;
13305   =>  1014    ;
13306   =>  1014    ;
13307   =>  1014    ;
13308   =>  1014    ;
13309   =>  1014    ;
13310   =>  1014    ;
13311   =>  1014    ;
13312   =>  1014    ;
13313   =>  1014    ;
13314   =>  1014    ;
13315   =>  1014    ;
13316   =>  1014    ;
13317   =>  1014    ;
13318   =>  1014    ;
13319   =>  1014    ;
```

```
13320   =>  1014    ;
13321   =>  1014    ;
13322   =>  1014    ;
13323   =>  1014    ;
13324   =>  1014    ;
13398   =>  1014    ;
13399   =>  1014    ;
13401   =>  2017    ;
13402   =>  2017    ;
13403   =>  2017    ;
13404   =>  2017    ;
13405   =>  2017    ;
13406   =>  2017    ;
13407   =>  2017    ;
13408   =>  2017    ;
13409   =>  2017    ;
13410   =>  2017    ;
13411   =>  2017    ;
13412   =>  2017    ;
13413   =>  2017    ;
13414   =>  2017    ;
13415   =>  2017    ;
13416   =>  2017    ;
13417   =>  2017    ;
13418   =>  2017    ;
13419   =>  2017    ;
13420   =>  2017    ;
13421   =>  2017    ;
13422   =>  2017    ;
13423   =>  2017    ;
13424   =>  2017    ;
13425   =>  2017    ;
13498   =>  2017    ;
13499   =>  2017    ;
13501   =>  1015    ;
13502   =>  1015    ;
13503   =>  1015    ;
13504   =>  1015    ;
13505   =>  1015    ;
13506   =>  1015    ;
13507   =>  1015    ;
13508   =>  1015    ;
13509   =>  1015    ;
13521   =>  1015    ;
13522   =>  1015    ;
13523   =>  1015    ;
13524   =>  1015    ;
13525   =>  1015    ;
13526   =>  1015    ;
13527   =>  1015    ;
13581   =>  1015    ;
13582   =>  1015    ;
13583   =>  1015    ;
13584   =>  1015    ;
13585   =>  1015    ;
13586   =>  1015    ;
13598   =>  1015    ;
13599   =>  1015    ;
13601   =>  2017    ;
13602   =>  2017    ;
13603   =>  2017    ;
13604   =>  2017    ;
13605   =>  2017    ;
13606   =>  2017    ;
13607   =>  2017    ;
13608   =>  2017    ;
13609   =>  2017    ;
13621   =>  2017    ;
13622   =>  2017    ;
13623   =>  2017    ;
13624   =>  2017    ;
```

```
13625   =>  2017    ;
13626   =>  2017    ;
13627   =>  2017    ;
13681   =>  2017    ;
13682   =>  2017    ;
13683   =>  2017    ;
13684   =>  2017    ;
13685   =>  2017    ;
13686   =>  2017    ;
13698   =>  2017    ;
13699   =>  2017    ;
13701   =>  1016    ;
13702   =>  1016    ;
13703   =>  1016    ;
13704   =>  1016    ;
13705   =>  1016    ;
13706   =>  1016    ;
13707   =>  1016    ;
13708   =>  1016    ;
13709   =>  1016    ;
13710   =>  1016    ;
13711   =>  1016    ;
13712   =>  1016    ;
13798   =>  1016    ;
13799   =>  1016    ;
13801   =>  2017    ;
13802   =>  2017    ;
13803   =>  2017    ;
13804   =>  2017    ;
13805   =>  2017    ;
13806   =>  2017    ;
13807   =>  2017    ;
13808   =>  2017    ;
13809   =>  2017    ;
13810   =>  2017    ;
13811   =>  2017    ;
13812   =>  2017    ;
13898   =>  2017    ;
13899   =>  2017    ;
21000   =>  1021    ;
22000   =>  1022    ;
23000   =>  1023    ;
24000   =>  1024    ;
25000   =>  1025    ;
29000   =>  1029    ;
31000   =>  1031    ;
32000   =>  1032    ;
33000   =>  1033    ;
34000   =>  1034    ;
35000   =>  1035    ;
39000   =>  1039    ;
41000   =>  1041    ;
42000   =>  1042    ;
43000   =>  1043    ;
44000   =>  1044    ;
45100   =>  1045    ;
45200   =>  1045    ;
49000   =>  1049    ;
51101   =>  1051    ;
51102   =>  1051    ;
51103   =>  1051    ;
51104   =>  1051    ;
51105   =>  1051    ;
51198   =>  1051    ;
51199   =>  1051    ;
52101   =>  3052    ;
52102   =>  3052    ;
52103   =>  3052    ;
52104   =>  3052    ;
52105   =>  3052    ;
52106   =>  3052    ;
```

```
52107   =>  3052    ;
52108   =>  3052    ;
52109   =>  3052    ;
52110   =>  3052    ;
52111   =>  3052    ;
52112   =>  3052    ;
52198   =>  3052    ;
52199   =>  3052    ;
53101   =>  4053    ;
53102   =>  4053    ;
53103   =>  4053    ;
53104   =>  4053    ;
53105   =>  4053    ;
53106   =>  4053    ;
53107   =>  4053    ;
53108   =>  4053    ;
53109   =>  4053    ;
53110   =>  4053    ;
53111   =>  4053    ;
53112   =>  4053    ;
53113   =>  4053    ;
53114   =>  4053    ;
53115   =>  4053    ;
53116   =>  4053    ;
53117   =>  4053    ;
53118   =>  4053    ;
53119   =>  4053    ;
53120   =>  4053    ;
53121   =>  4053    ;
53122   =>  4053    ;
53123   =>  4053    ;
53124   =>  4053    ;
53125   =>  4053    ;
53126   =>  4053    ;
53127   =>  4053    ;
53128   =>  4053    ;
53129   =>  4053    ;
53130   =>  4053    ;
53131   =>  4053    ;
53132   =>  4053    ;
53133   =>  4053    ;
53198   =>  4053    ;
53199   =>  4053    ;
53201   =>  4053    ;
53202   =>  4053    ;
53203   =>  4053    ;
53204   =>  4053    ;
53205   =>  4053    ;
53206   =>  4053    ;
53207   =>  4053    ;
53208   =>  4053    ;
53209   =>  4053    ;
53210   =>  4053    ;
53211   =>  4053    ;
53212   =>  4053    ;
53213   =>  4053    ;
53214   =>  4053    ;
53215   =>  4053    ;
53216   =>  4053    ;
53217   =>  4053    ;
53218   =>  4053    ;
53219   =>  4053    ;
53220   =>  4053    ;
53221   =>  4053    ;
53222   =>  4053    ;
53223   =>  4053    ;
53224   =>  4053    ;
53298   =>  4053    ;
53299   =>  4053    ;
53301   =>  4053    ;
53302   =>  4053    ;
```

```
53303   =>  4053    ;
53304   =>  4053    ;
53305   =>  4053    ;
53306   =>  4053    ;
53307   =>  4053    ;
53308   =>  4053    ;
53309   =>  4053    ;
53310   =>  4053    ;
53311   =>  4053    ;
53312   =>  4053    ;
53313   =>  4053    ;
53314   =>  4053    ;
53315   =>  4053    ;
53316   =>  4053    ;
53317   =>  4053    ;
53318   =>  4053    ;
53319   =>  4053    ;
53320   =>  4053    ;
53321   =>  4053    ;
53322   =>  4053    ;
53398   =>  4053    ;
53399   =>  4053    ;
53401   =>  4053    ;
53402   =>  4053    ;
53403   =>  4053    ;
53404   =>  4053    ;
53405   =>  4053    ;
53406   =>  4053    ;
53407   =>  4053    ;
53408   =>  4053    ;
53409   =>  4053    ;
53410   =>  4053    ;
53411   =>  4053    ;
53412   =>  4053    ;
53498   =>  4053    ;
53499   =>  4053    ;
61000   =>  5061    ;
62000   =>  5061    ;
71101   =>  6012    ;
71102   =>  6012    ;
71103   =>  6012    ;
71104   =>  6012    ;
71105   =>  6012    ;
71106   =>  6012    ;
71107   =>  6012    ;
71108   =>  6012    ;
71109   =>  6012    ;
71110   =>  6012    ;
71111   =>  6012    ;
71112   =>  6012    ;
71113   =>  6012    ;
71198   =>  6012    ;
71199   =>  6012    ;
71201   =>  6013    ;
71202   =>  6013    ;
71203   =>  6013    ;
71204   =>  6013    ;
71205   =>  6013    ;
71298   =>  6013    ;
71299   =>  6013    ;
71301   =>  6014    ;
71302   =>  6014    ;
71303   =>  6014    ;
71304   =>  6014    ;
71305   =>  6014    ;
71398   =>  6014    ;
71399   =>  6014    ;
71401   =>  6015    ;
71402   =>  6015    ;
71403   =>  6015    ;
71404   =>  6015    ;
```

```
    71405   =>  6015    ;
    71406   =>  6015    ;
    71498   =>  6015    ;
    71499   =>  6015    ;
    71501   =>  6016    ;
    71502   =>  6016    ;
    71503   =>  6016    ;
    71598   =>  6016    ;
    71599   =>  6016    ;
endrecode;
if I_INCOME_TYPE in 4 then
    I_CATEGORY = 4053;
endif;

recode  I_CATEGORY  =>  I_CATEGORY  ;
    1011    =>  11  ;
    1012    =>  12  ;
    1013    =>  13  ;
    1014    =>  14  ;
    1015    =>  15  ;
    1016    =>  16  ;
    1021    =>  21  ;
    1022    =>  22  ;
    1023    =>  23  ;
    1031    =>  31  ;
    1032    =>  32  ;
    1033    =>  33  ;
    1035    =>  35  ;
    1039    =>  39  ;
    1042    =>  42  ;
    1051    =>  51  ;
    2017    =>  17  ;
    3052    =>  52  ;
    4053    =>  53  ;
    5061    =>  61  ;
    6012    =>  12  ;
    6013    =>  13  ;
    6014    =>  14  ;
    6015    =>  15  ;
    6016    =>  16  ;
    7011    =>  11  ;
endrecode;

PROC I_DESCRIPTION
    // INCOME SIDE
    if I_WGHT_AN_GROSS > 0 then
        numeric tmp = noccurs(F_INCOME_EDT)+1;
        insert(F_INCOME_EDT(tmp));
        FI_LINE_NUMBER(tmp) = tmp;
        FI_HIES_CODE(tmp) = I_HIES_CODE;
        FI_PACCOI(tmp) = I_PACCOI;
        FI_COICOP(tmp) = I_COICOP;
        FI_AMOUNT(tmp) = I_GROSS;
        FI_AN_AMOUNT(tmp) = I_AN_GROSS;
        FI_WGHT_AN_AMOUNT(tmp) = I_WGHT_AN_GROSS;
        FI_MULTIPLIER(tmp) = I_MULTIPLIER;
        FI_LOCATION(tmp) = I_LOCATION;
        FI_INCOME_TYPE(tmp) = I_INCOME_TYPE;
        FI_PROD_TYPE(tmp) = I_PROD_TYPE;
        //FI_QUANTITY(tmp) = I_QUANTITY;
        FI_NUMBER(tmp) = I_NUMBER;
        FI_UNIT(tmp) = I_UNIT;
        FI_CATEGORY(tmp) = I_CATEGORY;
        FI_BENEFICIARY(tmp) = I_BENEFICIARY;
        FI_INDUSTRY(tmp) = I_INDUSTRY;
        FI_OCCUPATION(tmp) = I_OCCUPATION;
        FI_DESCRIPTION(tmp) = I_DESCRIPTION;
    endif;
PROC EXPENDITURE_EDT
preproc
    if ID07 in 108, 210 then
```

```
        i = noccurs($)+1;
        insert($(i));
        E_LINE_NUMBER(i) = i;
        E_HIES_CODE(i) = 203.031;
        E_COICOP(i) = 45212702;
        E_MULTIPLIER(i) = 1;
        E_AMOUNT(i) = 460;
        E_AN_AMT(i) = E_AMOUNT(i) * E_MULTIPLIER(i);
        E_WGHT_AN_AMT(i) = E_AN_AMT(i)*WEIGHT;
        E_LOCATION(i) = 1;
        E_EXP_TYPE(i) = 1;
        E_PROD_TYPE(i) = notappl;
        E_NUMBER(i) = notappl;
        E_UNIT(i) = notappl;
        E_CATEGORY(i) = notappl;
        E_BENEFICIARY(i) = 51;
        E_DURABILITY(i) = 1;
        E_DESCRIPTION(i) = "[utility exp]lpg";
    endif;
PROC E_COICOP
// data fix from bertrand on June 29

if '[diary purchase]fruit loops' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11101006; endif;
if '[diary home consume]barraduca' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11302201; endif;
if '[diary home consume]parrot fish' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11302213; endif;
if '[diary purchase]fish' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11302238; endif;
if '[diary home consume]mahimahi' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11302239; endif;
if '[diary home given]masimasi fish' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11302239; endif;
if '[diary home consume]wahoo' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11302240; endif;
if '[diary home given]wahoo' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11302240; endif;
if '[diary home consume]5 flying fish' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11302241; endif;
if '[diary home consume]flying fish' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11302241; endif;
if '[diary purchase]milk' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11402607; endif;
if '[diary purchase]box milk 12 x 1ltr' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11402607; endif;
if '[diary purchase]milk 1 x 12 ctns' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11402607; endif;
if '[diary purchase]anchor milk powder x1' = strip(E_DESCRIPTION) then errmsg('PIERRE
CHANGED THIS'); E_COICOP = 11403005; endif;
if '[diary purchase]milk powder (anchor) x1' = strip(E_DESCRIPTION) then errmsg('PIERRE
CHANGED THIS'); E_COICOP = 11403005; endif;
if '[diary purchase]milk powder x1 (anchor)' = strip(E_DESCRIPTION) then errmsg('PIERRE
CHANGED THIS'); E_COICOP = 11403005; endif;
if '[diary purchase]peanut' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11604301; endif;
if '[diary home consume]garden peas' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 11704501; endif;
if '[diary home consume]pumpkin' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11704524; endif;
if '[diary home given]pumpkin' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 11704524; endif;
if '[diary purchase]water' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 12207101; endif;
if '[clothing exp]material' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 31109202; endif;
if '[diary purchase]material' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 31109202; endif;
if '[diary purchase]laundry' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 31409901; endif;
if '[land exp]concrete' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS'); E_COICOP
```

```
= 43111801; endif;
if '[land exp]concrete bags x6' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 43111801; endif;
if '[land exp]concrete bags' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 43111801; endif;
if '[land exp]concrete blocks' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 43111801; endif;
if '[diary purchase]1 gas cylinder 100pnds' = strip(E_DESCRIPTION) then errmsg('PIERRE
CHANGED THIS'); E_COICOP = 45212702; endif;
if '[diary purchase]gas bottle' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 45212702; endif;
if '[diary purchase]dish washer' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 56116902; endif;
if '[utility exp]petrol for scooter' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 72219601; endif;
if '[diary purchase]phone bill' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 83020904; endif;
if '[diary purchase]telephone bills' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED
THIS'); E_COICOP = 83020904; endif;
if '[hh-goods exp]xbox' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS'); E_COICOP
= 93122204; endif;
if '[utility exp]satelite tv' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[utility exp]sttelite tv' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[utility exp]satellite t.v.' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[utility exp]sky t.v.' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[utility exp]satellite sky subsription' = strip(E_DESCRIPTION) then errmsg('PIERRE
CHANGED THIS'); E_COICOP = 94224701; endif;
if '[utility exp]sttellite t.v.' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[utility exp]satelitte t.v' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 94224701; endif;
if '[edu exp]text/stationary' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 95425416; endif;
if '[clothing exp]parfum' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 121327716; endif;
if '[diary purchase]funural' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 901100102; endif;
if '[diary purchase]funeral' = strip(E_DESCRIPTION) then errmsg('PIERRE CHANGED THIS');
E_COICOP = 901100102; endif;

if E_COICOP = 11302231 then E_COICOP = 11302227; endif;

// new price for cigarettes
if ID07 in 138 then
    if E_LINE_NUMBER in 25 then
        impute(E_AMOUNT,
        setNumFlag({questName}'E_AMOUNT',{lineNo}curocc(),{imputeType}1,{newVal}7));
        impute(E_AN_AMT, E_AMOUNT * 26);
        impute(E_WGHT_AN_AMT, E_AN_AMT * WEIGHT);
        impute(E_COICOP,
        setNumFlag({questName}'E_COICOP',{lineNo}curocc(),{imputeType}1,{newVal}22108201));
    endif;
endif;

// new price for airfare
if ID07 in 209 then
    if E_LINE_NUMBER in 16 then
        impute(E_AMOUNT,
        setNumFlag({questName}'E_AMOUNT',{lineNo}curocc(),{imputeType}1,{newVal}272));
        impute(E_AN_AMT, E_AMOUNT * 1);
        impute(E_WGHT_AN_AMT, E_AN_AMT * WEIGHT);
        impute(E_COICOP,
        setNumFlag({questName}'E_COICOP',{lineNo}curocc(),{imputeType}1,{newVal}73320201));
    endif;
endif;

// new price for shampoo
```

```
if ID07 in 311 then
    if E_LINE_NUMBER in 54 then
        impute(E_AMOUNT,
        setNumFlag({questName}'E_AMOUNT',{lineNo}curocc(),{imputeType}1,{newVal}6));
        impute(E_AN_AMT, E_AMOUNT * 26);
        impute(E_WGHT_AN_AMT, E_AN_AMT * WEIGHT);
        impute(E_COICOP,
        setNumFlag({questName}'E_COICOP',{lineNo}curocc(),{imputeType}1,{newVal}121327604));
    endif;
endif;

// new price for cold power
if ID07 in 343 then
    if E_LINE_NUMBER in 34 then
        impute(E_AMOUNT,
        setNumFlag({questName}'E_AMOUNT',{lineNo}curocc(),{imputeType}1,{newVal}17));
        impute(E_AN_AMT, E_AMOUNT * 26);
        impute(E_WGHT_AN_AMT, E_AN_AMT * WEIGHT);
        impute(E_COICOP,
        setNumFlag({questName}'E_COICOP',{lineNo}curocc(),{imputeType}1,{newVal}56116920));
    endif;
endif;

// wrong COICOP for fridge
if ID07 in 304 and E_LINE_NUMBER in 7 and E_HIES_CODE in 205.302 then
    E_COICOP = 53114908;
endif;

// wrong COICOP for fittings
if ID07 in 118 and E_LINE_NUMBER in 7 then
    E_COICOP = 43111505;
endif;

// wrong COICOP for frieght
if ID07 in 305 and E_LINE_NUMBER in 8 then
    E_COICOP = 73620501;
endif;

// wrong COICOP for 6 packs of cigarettes
if ID07 in 311 and E_LINE_NUMBER in 54 then
    E_COICOP = 22108203;
endif;

// wrong COICOP for 6 packs of cigarettes
if ID07 in 311 and E_LINE_NUMBER in 54 then
    E_COICOP = 22108202;
endif;
// wrong COICOP for hotdogs
if ID07 in 220 and E_LINE_NUMBER in 19 then
    E_COICOP = 11201906;
endif;

// wrong COICOP for hotdogs
if ID07 in 324 and E_LINE_NUMBER in 28 then
    E_COICOP = 11201906;
endif;

// wrong COICOP for rainbows cereal
if ID07 in 239 and E_LINE_NUMBER in 96 then
    E_COICOP = 11201906;
endif;

// wrong COICOP for mufti change to donations non-consumption
if E_COICOP = 105026203 then
    E_COICOP = 901100601;
endif;

// wrong COICOP for gas cylinder change to gas refill
if E_COICOP = 45212701 then
    E_COICOP = 45212702;
endif;
```

```
PROC E_AMOUNT
recode  ID07    :   E_LINE_NUMBER   :   E_HIES_CODE =>  E_AMOUNT    ;
    208 :   3   :   203.031 =>  460 ;
    304 :   3   :   203.031 =>  460 ;
    306 :   3   :   203.031 =>  460 ;
    309 :   3   :   203.031 =>  280 ;
    311 :   3   :   203.031 =>  460 ;
endrecode
PROC E_MULTIPLIER
if E_COICOP in 11100101:12207506 then
    if $ in 26 then
        impute($, 26.07142857);
    else
        impute($, 52.14285714);
    endif;
else
    do varying i = 1 until i >= 316
        if E_COICOP in Selection(i,2) then
            break;
        endif
    enddo;

    {
    HIES ID Items week 1    Items week 2    Total both weeks    W1/W2 ratio Source  Multiplier
    308     1               18              19                  0.1         Week 2  52
    318     2               22              24                  0.1         Week 2  52
    118     19              6               25                  3.2         Week 1  52
    326     42              13              55                  3.2         Week 1  52
    250     45              11              56                  4.1         Week 1  52
    134     55              3               58                  18.3        Week 1  52
    105     31                              31                  0.0         Week 1  52

    week1 = P_DAY in 1:7
    week2 = P_DAY in 8:14
    }

    numeric Source = Selection(i,3);
    numeric TwoD = Selection(i,4);
    numeric OneD = Selection(i,5);
    numeric Mod = Selection(i,6);
    numeric HHOneDiary = 0;

    if ID07 in 308,318,118,326,250,134,105 then
        HHoneDiary = 1;
    endif;

    if Source in 1:3 then
        if Source = 1 then
            if E_HIES_CODE >= 500 then
                if HHOneDiary then
                    E_MULTIPLIER = OneD;
                    errmsg("source1:One Diary") summary;
                else
                    E_MULTIPLIER = TwoD;
                    errmsg("source1:Two Diary") summary;
                endif;
            else
                E_MULTIPLIER = 0;
                errmsg("source1:Module(0)") summary;
            endif;
        endif;
        if Source = 2 then
            if E_HIES_CODE < 500 then
                E_MULTIPLIER = Mod;
                errmsg("source2:Module") summary;
            else
                E_MULTIPLIER = 0;
                errmsg("source2:Diary(0)") summary;
            endif;
        endif;
        if Source = 3 then
```

```
            if HHOneDiary then
                E_MULTIPLIER = OneD;
                errmsg("source3:One Diary") summary;
            else
                E_MULTIPLIER = TwoD;
                errmsg("source3:Two Diary") summary;
            endif;
        endif;
    else
        errmsg("SOURCE/COICOP not found: %d",E_COICOP);
    endif;
endif;


if ID07 in 132 and E_LINE_NUMBER in 58 and E_HIES_CODE in 510.001 then
    E_MULTIPLIER = 0;
endif;
if ID07 in 138 and E_LINE_NUMBER in 18 and E_HIES_CODE in 304.003 then
    E_MULTIPLIER = 0;
endif;
if ID07 in 126 and E_LINE_NUMBER in 31 and E_HIES_CODE in 510.002 then
    E_MULTIPLIER = 0.963060686;
endif;
if ID07 in 347 and E_LINE_NUMBER in 10 and E_HIES_CODE in 204.331 then
    E_MULTIPLIER = 0.963060686;
endif;

E_AN_AMT = E_AMOUNT * E_MULTIPLIER;
E_WGHT_AN_AMT = E_AN_AMT * WEIGHT;


PROC E_EXP_TYPE
numeric tmp = int(E_COICOP/10000000);
if E_EXP_TYPE in 1 then
    E_CATEGORY = 100 + tmp;
elseif E_EXP_TYPE in 2 then
    E_CATEGORY = 200 + tmp;
elseif E_EXP_TYPE in 3 then
    E_CATEGORY = 300 + tmp;
elseif E_EXP_TYPE in 4 then
    E_CATEGORY = 400 + tmp;
elseif E_EXP_TYPE in 5 then
    E_CATEGORY = 500 + tmp;
elseif E_EXP_TYPE in 6 then
    E_CATEGORY = notappl;
elseif E_EXP_TYPE in 7 then
    E_CATEGORY = 612;
endif;

recode  E_COICOP    =>  E_CATEGORY  ;
    901100101   =>  701 ;
    901100102   =>  701 ;
    901100103   =>  701 ;
    901100104   =>  701 ;
    901100105   =>  701 ;
    901100106   =>  703 ;
    901100108   =>  701 ;
    901100109   =>  701 ;
    901100110   =>  702 ;
    901100301   =>  703 ;
    901100302   =>  703 ;
    901100601   =>  704 ;
    901100602   =>  704 ;
    901100603   =>  704 ;
    901100704   =>  705 ;
endrecode;

recode  E_HIES_CODE :   tmp =>  E_CATEGORY  ;
    204.202 :       =>  802 ;
    204.204 :       =>  802 ;
    204.205 :       =>  802 ;
```

```
    204.206 :   4   =>  802 ;
    204.206 :   5   =>  805 ;
endrecode;

recode E_COICOP =>  E_CATEGORY;
    55115902 {water pump} => 803 ;
    53115111 {water tank} => 803 ;
    53115109 {generator} => 803 ;
    92121801 {boat} => 803 ;
    92121801 {outboard motor} => 803 ;
endrecode;

PROC E_CATEGORY
// let do some recode to make it easier for cross tabulations
recode  E_CATEGORY  =>  E_CATEGORY  ;
    101 =>  101 ;
    102 =>  102 ;
    103 =>  103 ;
    104 =>  104 ;
    105 =>  105 ;
    106 =>  106 ;
    107 =>  107 ;
    108 =>  108 ;
    109 =>  109 ;
    110 =>  110 ;
    111 =>  111 ;
    112 =>  112 ;
    201 =>  101 ;
    301 =>  101 ;
    302 =>  102 ;
    303 =>  103 ;
    304 =>  104 ;
    305 =>  105 ;
    306 =>  106 ;
    307 =>  107 ;
    310 =>  110 ;
    311 =>  111 ;
    312 =>  112 ;
    401 =>  101 ;
    504 =>  104 ;
    612 =>  112 ;
    701 =>  201 ;
    702 =>  202 ;
    703 =>  203 ;
    704 =>  204 ;
    705 =>  205 ;
    802 =>  302 ;
    803 =>  303 ;
endrecode;
PROC E_DESCRIPTION
preproc

if E_WGHT_AN_AMT > 0 then
    i = noccurs(F_EXPENDITURE_EDT)+1;
    insert(F_EXPENDITURE_EDT(i));
    FE_LINE_NUMBER(i) = i;
    FE_HIES_CODE(i) = E_HIES_CODE;
    FE_COICOP(i) = E_COICOP;
    FE_MULTIPLIER(i) = E_MULTIPLIER;
    FE_AMOUNT(i) = E_AMOUNT;
    FE_AN_AMT(i) = E_AMOUNT * E_MULTIPLIER;
    FE_WGHT_AN_AMT(i) = FE_AN_AMT(i)*WEIGHT;
    FE_LOCATION(i) = E_LOCATION;
    FE_EXP_TYPE(i) = E_EXP_TYPE;
    FE_PROD_TYPE(i) = E_PROD_TYPE;
    FE_NUMBER(i) = E_NUMBER;
    FE_UNIT(i) = E_UNIT;
    FE_CATEGORY(i) = E_CATEGORY;
    FE_BENEFICIARY(i) = E_BENEFICIARY;
    FE_DURABILITY(i) = E_DURABILITY;
    FE_DESCRIPTION(i) = strip(tolower(E_DESCRIPTION));
```

```
endif;
PROC F_INCOME_EDT
{if sum(FI_AN_AMOUNT where FI_CATEGORY in 13 and FI_INCOME_TYPE in 1) = 0 then
    do varying i = 1 until i > noccurs($)
        if FI_CATEGORY(i) in 13 and FI_INCOME_TYPE(i) in 6 then
            FI_CATEGORY(i) = 17;
        endif;
    enddo;
endif;
if sum(FI_AN_AMOUNT where FI_CATEGORY in 14 and FI_INCOME_TYPE in 1) = 0 then
    do varying i = 1 until i > noccurs($)
        if FI_CATEGORY(i) in 14 and FI_INCOME_TYPE(i) in 6 then
            FI_CATEGORY(i) = 17;
        endif;
    enddo;
endif;
if sum(FI_AN_AMOUNT where FI_CATEGORY in 15 and FI_INCOME_TYPE in 1) = 0 then
    do varying i = 1 until i > noccurs($)
        if FI_CATEGORY(i) in 15 and FI_INCOME_TYPE(i) in 6 then
            FI_CATEGORY(i) = 17;
        endif;
    enddo;
endif;
if sum(FI_AN_AMOUNT where FI_CATEGORY in 16 and FI_INCOME_TYPE in 1) = 0 then
    do varying i = 1 until i > noccurs($)
        if FI_CATEGORY(i) in 16 and FI_INCOME_TYPE(i) in 6 then
            FI_CATEGORY(i) = 17;
        endif;
    enddo;
endif;}
if ID07 in 111,117,210,233,307,341 then
    i = seek (FI_COICOP = 42211401);
    if i then
        delete($(i));
        errmsg("(INC) PIERRE DELETED IMPUTED RENT FOR THOSE THAT DECLARED IT IN IN-KIND
        INCOME");
    endif;
endif;
PROC FI_AMOUNT

recode  ID07    :   FI_LINE_NUMBER  =>  FI_AMOUNT   ;
    102 :   1   =>  212.3835616 ;
    103 :   1   =>  166.5753424 ;
    104 :   1   =>  293.1726027 ;
    105 :   1   =>  187.3972602 ;
    106 :   1   =>  183.2328767 ;
    108 :   1   =>  224.8767123 ;
    109 :   1   =>  142.4219178 ;
    113 :   1   =>  483.068493  ;
    114 :   1   =>  179.9013698 ;
    116 :   1   =>  229.8739726 ;
    118 :   1   =>  267.3534246 ;
    119 :   1   =>  334.8164383 ;
    120 :   1   =>  179.0684931 ;
    121 :   1   =>  261.1068493 ;
    124 :   1   =>  261.1068493 ;
    125 :   1   =>  217.3808219 ;
    126 :   1   =>  266.9369862 ;
    127 :   1   =>  229.0410958 ;
    132 :   1   =>  251.5287671 ;
    134 :   1   =>  248.8635616 ;
    135 :   1   =>  209.4684931 ;
    136 :   1   =>  264.8547945 ;
    138 :   1   =>  293.1726027 ;
    139 :   1   =>  259.6076712 ;
    140 :   1   =>  329.819178  ;
    141 :   1   =>  319.8246575 ;
    142 :   1   =>  187.8136986 ;
    143 :   1   =>  427.7654794 ;
    147 :   1   =>  171.5726027 ;
    148 :   1   =>  303.9999999 ;
```

```
150 :   1   =>  285.5101369 ;
151 :   1   =>  235.2876712 ;
154 :   1   =>  227.3753424 ;
155 :   1   =>  284.8438356 ;
158 :   1   =>  179.9013698 ;
159 :   1   =>  249.8630136 ;
160 :   1   =>  229.8739726 ;
201 :   1   =>  274.849315  ;
203 :   1   =>  235.4542465 ;
204 :   1   =>  204.0547945 ;
206 :   1   =>  170.739726  ;
207 :   1   =>  182.899726  ;
208 :   1   =>  166.5753424 ;
209 :   1   =>  187.2306849 ;
211 :   1   =>  199.8904109 ;
213 :   1   =>  123.6821918 ;
217 :   1   =>  152.249863  ;
218 :   1   =>  195.7260274 ;
219 :   1   =>  243.2832876 ;
220 :   1   =>  294.8383561 ;
221 :   1   =>  215.2986301 ;
222 :   1   =>  183.649315  ;
223 :   1   =>  310.1632876 ;
224 :   1   =>  195.1430137 ;
226 :   1   =>  226.1260273 ;
227 :   1   =>  187.3972602 ;
232 :   1   =>  188.0635616 ;
234 :   1   =>  168.9073972 ;
235 :   1   =>  166.1589041 ;
236 :   1   =>  289.8410958 ;
237 :   1   =>  206.1369863 ;
238 :   1   =>  195.7260274 ;
239 :   1   =>  201.9726027 ;
240 :   1   =>  167.3249315 ;
241 :   1   =>  219.4630137 ;
246 :   1   =>  164.909589  ;
247 :   1   =>  195.7260274 ;
249 :   1   =>  254.0273972 ;
250 :   1   =>  208.219178  ;
252 :   1   =>  261.1068493 ;
253 :   1   =>  278.7638356 ;
254 :   1   =>  344.3945205 ;
255 :   1   =>  236.9534246 ;
256 :   1   =>  171.239452  ;
258 :   1   =>  278.5972602 ;
301 :   1   =>  179.9013698 ;
302 :   1   =>  154.0821917 ;
304 :   1   =>  137.4246575 ;
305 :   1   =>  224.8767123 ;
306 :   1   =>  311.4958903 ;
308 :   1   =>  220.7123287 ;
309 :   1   =>  436.4273972 ;
310 :   1   =>  231.0399999 ;
311 :   1   =>  174.9041096 ;
318 :   1   =>  224.8767123 ;
319 :   1   =>  732.9315067 ;
320 :   1   =>  229.8739726 ;
322 :   1   =>  309.8301369 ;
323 :   1   =>  487.2328766 ;
324 :   1   =>  238.2027397 ;
326 :   1   =>  284.8438356 ;
327 :   1   =>  417.2712328 ;
328 :   1   =>  209.0520547 ;
329 :   1   =>  151.1671233 ;
332 :   1   =>  239.8684931 ;
333 :   1   =>  255.5265753 ;
336 :   1   =>  224.4602739 ;
337 :   1   =>  174.0712328 ;
338 :   1   =>  183.2328767 ;
339 :   1   =>  330.5687671 ;
340 :   1   =>  229.8739726 ;
```

```
    342 :   1   =>  319.8246575 ;
    343 :   1   =>  319.8246575 ;
    346 :   1   =>  458.49863   ;
    347 :   1   =>  493.0630136 ;
    348 :   1   =>  195.7260274 ;
    349 :   1   =>  302.3342465 ;
    350 :   1   =>  265.6876712 ;
    351 :   1   =>  261.1068493 ;
    352 :   1   =>  158.2465753 ;
    353 :   1   =>  367.2986301 ;
    354 :   1   =>  365.2164383 ;
    356 :   1   =>  238.2027397 ;
endrecode;



FI_AN_AMOUNT = FI_AMOUNT * FI_MULTIPLIER;
FI_WGHT_AN_AMOUNT = FI_AN_AMOUNT * WEIGHT;




if FI_INCOME_TYPE = 6 then
    $ = $ * (-1);
endif;
PROC FI_AN_AMOUNT
if FI_INCOME_TYPE = 6 then
    $ = $ * (-1);
endif;
PROC FI_WGHT_AN_AMOUNT
if FI_INCOME_TYPE = 6 then
    $ = $ * (-1);
endif;
PROC FI_NUMBER
// Corrections to NUMBER of UNIT
if ID07 in 126 and FI_LINE_NUMBER in 16 then FI_NUMBER = 1; endif;
if ID07 in 125 and FI_LINE_NUMBER in 12 then FI_NUMBER = 2; endif;
if ID07 in 125 and FI_LINE_NUMBER in 14 then FI_NUMBER = 2; endif;
if ID07 in 121 and FI_LINE_NUMBER in 10 then FI_NUMBER = 2; endif;
if ID07 in 125 and FI_LINE_NUMBER in 5 then FI_NUMBER = 3; endif;
if ID07 in 125 and FI_LINE_NUMBER in 10 then FI_NUMBER = 3; endif;
if ID07 in 125 and FI_LINE_NUMBER in 7 then FI_NUMBER = 4; endif;
if ID07 in 125 and FI_LINE_NUMBER in 11 then FI_NUMBER = 5; endif;
if ID07 in 121 and FI_LINE_NUMBER in 11 then FI_NUMBER = 5; endif;
if ID07 in 127 and FI_LINE_NUMBER in 18 then FI_NUMBER = 5; endif;
if ID07 in 126 and FI_LINE_NUMBER in 5 then FI_NUMBER = 5; endif;
if ID07 in 121 and FI_LINE_NUMBER in 8 then FI_NUMBER = 5; endif;
if ID07 in 125 and FI_LINE_NUMBER in 9 then FI_NUMBER = 7; endif;
if ID07 in 125 and FI_LINE_NUMBER in 15 then FI_NUMBER = 10; endif;
if ID07 in 127 and FI_LINE_NUMBER in 17 then FI_NUMBER = 10; endif;
if ID07 in 127 and FI_LINE_NUMBER in 15 then FI_NUMBER = 10; endif;
if ID07 in 127 and FI_LINE_NUMBER in 12 then FI_NUMBER = 10; endif;
if ID07 in 125 and FI_LINE_NUMBER in 6 then FI_NUMBER = 10; endif;
if ID07 in 116 and FI_LINE_NUMBER in 10 then FI_NUMBER = 10; endif;
if ID07 in 116 and FI_LINE_NUMBER in 8 then FI_NUMBER = 10; endif;
if ID07 in 124 and FI_LINE_NUMBER in 18 then FI_NUMBER = 20; endif;
if ID07 in 203 and FI_LINE_NUMBER in 8 then FI_NUMBER = 20; endif;
if ID07 in 119 and FI_LINE_NUMBER in 10 then FI_NUMBER = 20; endif;
if ID07 in 126 and FI_LINE_NUMBER in 14 then FI_NUMBER = 50; endif;
if ID07 in 119 and FI_LINE_NUMBER in 9 then FI_NUMBER = 50; endif;
if ID07 in 126 and FI_LINE_NUMBER in 13 then FI_NUMBER = 70; endif;
if ID07 in 124 and FI_LINE_NUMBER in 16 then FI_NUMBER = 75; endif;
if ID07 in 116 and FI_LINE_NUMBER in 9 then FI_NUMBER = 100; endif;
if ID07 in 126 and FI_LINE_NUMBER in 9 then FI_NUMBER = 500; endif;

// Corrections to UNIT
if ID07 in 111 and FI_LINE_NUMBER in 32 then FI_UNIT = 1; endif;
if ID07 in 121 and FI_LINE_NUMBER in 10 then FI_UNIT = 2; endif;
if ID07 in 121 and FI_LINE_NUMBER in 11 then FI_UNIT = 2; endif;
if ID07 in 127 and FI_LINE_NUMBER in 18 then FI_UNIT = 2; endif;
if ID07 in 127 and FI_LINE_NUMBER in 17 then FI_UNIT = 2; endif;
```

```
if ID07 in 127 and FI_LINE_NUMBER in 15 then FI_UNIT = 2; endif;
if ID07 in 127 and FI_LINE_NUMBER in 12 then FI_UNIT = 2; endif;
if ID07 in 116 and FI_LINE_NUMBER in 10 then FI_UNIT = 2; endif;
if ID07 in 116 and FI_LINE_NUMBER in 8 then FI_UNIT = 2; endif;
if ID07 in 116 and FI_LINE_NUMBER in 9 then FI_UNIT = 2; endif;
if ID07 in 323 and FI_LINE_NUMBER in 16 then FI_UNIT = 2; endif;
if ID07 in 135 and FI_LINE_NUMBER in 15 then FI_UNIT = 2; endif;
if ID07 in 135 and FI_LINE_NUMBER in 14 then FI_UNIT = 2; endif;
if ID07 in 155 and FI_LINE_NUMBER in 11 then FI_UNIT = 2; endif;
if ID07 in 155 and FI_LINE_NUMBER in 10 then FI_UNIT = 2; endif;
if ID07 in 221 and FI_LINE_NUMBER in 4 then FI_UNIT = 2; endif;
if ID07 in 154 and FI_LINE_NUMBER in 12 then FI_UNIT = 2; endif;
if ID07 in 124 and FI_LINE_NUMBER in 13 then FI_UNIT = 2; endif;
if ID07 in 108 and FI_LINE_NUMBER in 16 then FI_UNIT = 2; endif;
if ID07 in 109 and FI_LINE_NUMBER in 11 then FI_UNIT = 2; endif;
if ID07 in 118 and FI_LINE_NUMBER in 8 then FI_UNIT = 2; endif;
if ID07 in 109 and FI_LINE_NUMBER in 15 then FI_UNIT = 2; endif;
if ID07 in 108 and FI_LINE_NUMBER in 22 then FI_UNIT = 2; endif;
if ID07 in 217 and FI_LINE_NUMBER in 6 then FI_UNIT = 2; endif;
if ID07 in 111 and FI_LINE_NUMBER in 18 then FI_UNIT = 2; endif;
if ID07 in 108 and FI_LINE_NUMBER in 24 then FI_UNIT = 2; endif;
if ID07 in 111 and FI_LINE_NUMBER in 19 then FI_UNIT = 2; endif;
if ID07 in 124 and FI_LINE_NUMBER in 14 then FI_UNIT = 2; endif;
if ID07 in 108 and FI_LINE_NUMBER in 12 then FI_UNIT = 2; endif;
if ID07 in 124 and FI_LINE_NUMBER in 10 then FI_UNIT = 2; endif;
if ID07 in 103 and FI_LINE_NUMBER in 7 then FI_UNIT = 2; endif;
if ID07 in 109 and FI_LINE_NUMBER in 9 then FI_UNIT = 2; endif;
if ID07 in 126 and FI_LINE_NUMBER in 16 then FI_UNIT = 26; endif;
if ID07 in 125 and FI_LINE_NUMBER in 10 then FI_UNIT = 26; endif;
if ID07 in 126 and FI_LINE_NUMBER in 5 then FI_UNIT = 26; endif;
if ID07 in 121 and FI_LINE_NUMBER in 8 then FI_UNIT = 26; endif;
if ID07 in 125 and FI_LINE_NUMBER in 9 then FI_UNIT = 26; endif;
if ID07 in 125 and FI_LINE_NUMBER in 6 then FI_UNIT = 26; endif;
if ID07 in 119 and FI_LINE_NUMBER in 10 then FI_UNIT = 26; endif;
if ID07 in 155 and FI_LINE_NUMBER in 7 then FI_UNIT = 26; endif;
if ID07 in 151 and FI_LINE_NUMBER in 13 then FI_UNIT = 26; endif;
if ID07 in 206 and FI_LINE_NUMBER in 9 then FI_UNIT = 26; endif;
if ID07 in 127 and FI_LINE_NUMBER in 16 then FI_UNIT = 26; endif;
if ID07 in 125 and FI_LINE_NUMBER in 13 then FI_UNIT = 26; endif;
if ID07 in 109 and FI_LINE_NUMBER in 7 then FI_UNIT = 26; endif;
if ID07 in 117 and FI_LINE_NUMBER in 21 then FI_UNIT = 26; endif;
if ID07 in 117 and FI_LINE_NUMBER in 14 then FI_UNIT = 26; endif;
if ID07 in 108 and FI_LINE_NUMBER in 11 then FI_UNIT = 26; endif;
if ID07 in 109 and FI_LINE_NUMBER in 8 then FI_UNIT = 26; endif;
if ID07 in 109 and FI_LINE_NUMBER in 13 then FI_UNIT = 26; endif;
if ID07 in 338 and FI_LINE_NUMBER in 8 then FI_UNIT = 26; endif;
if ID07 in 332 and FI_LINE_NUMBER in 11 then FI_UNIT = 26; endif;
if ID07 in 108 and FI_LINE_NUMBER in 15 then FI_UNIT = 26; endif;
if ID07 in 126 and FI_LINE_NUMBER in 7 then FI_UNIT = 26; endif;
if ID07 in 126 and FI_LINE_NUMBER in 6 then FI_UNIT = 26; endif;
if ID07 in 108 and FI_LINE_NUMBER in 20 then FI_UNIT = 26; endif;
if ID07 in 104 and FI_LINE_NUMBER in 7 then FI_UNIT = 26; endif;
PROC FI_CATEGORY
if FI_CATEGORY in 13:15 and FI_INCOME_TYPE in 6 then
    FI_CATEGORY = 17;
endif;
PROC F_EXPENDITURE_EDT
if ID07 = 134 and FE_LINE_NUMBER = 6 then
    FE_AMOUNT = 296;
    FE_AN_AMT = FE_AMOUNT * FE_MULTIPLIER;
    FE_WGHT_AN_AMT = FE_AN_AMT * WEIGHT;
endif;

if ID07 = 332 and FE_LINE_NUMBER = 5 then
    FE_AMOUNT = 82;
    FE_AN_AMT = FE_AMOUNT * FE_MULTIPLIER;
    FE_WGHT_AN_AMT = FE_AN_AMT * WEIGHT;
endif;


if ID07 in 111,117,210,233,307,341 then
```

```
    i = seek (FE_COICOP = 42211401);
    if i then
        delete($(i));
        errmsg("PIERRE DELETED IMPUTED RENT FOR THOSE THAT DECLARED IT IN IN-KIND INCOME");
    endif;
endif;
PROC FE_AMOUNT
if ID07 in 140 and FE_LINE_NUMBER in 32 then FE_AMOUNT = 2; endif;
if ID07 in 109 and FE_LINE_NUMBER in 52 then FE_AMOUNT = 3.6; endif;
if ID07 in 114 and FE_LINE_NUMBER in 24 then FE_AMOUNT = 3.6; endif;
if ID07 in 340 and FE_LINE_NUMBER in 56 then FE_AMOUNT = 3.6; endif;
if ID07 in 124 and FE_LINE_NUMBER in 47 then FE_AMOUNT = 3.6; endif;
if ID07 in 108 and FE_LINE_NUMBER in 60 then FE_AMOUNT = 5; endif;
if ID07 in 108 and FE_LINE_NUMBER in 59 then FE_AMOUNT = 5; endif;
if ID07 in 247 and FE_LINE_NUMBER in 20 then FE_AMOUNT = 10; endif;


recode  ID07    :   FE_LINE_NUMBER  =>  FE_AMOUNT   ;
    102 :   1   =>  212.3835616 ;
    103 :   1   =>  166.5753424 ;
    104 :   1   =>  293.1726027 ;
    105 :   1   =>  187.3972602 ;
    106 :   1   =>  183.2328767 ;
    108 :   1   =>  224.8767123 ;
    109 :   1   =>  142.4219178 ;
    113 :   1   =>  483.068493  ;
    114 :   1   =>  179.9013698 ;
    116 :   1   =>  229.8739726 ;
    118 :   1   =>  267.3534246 ;
    119 :   1   =>  334.8164383 ;
    120 :   1   =>  179.0684931 ;
    121 :   1   =>  261.1068493 ;
    124 :   1   =>  261.1068493 ;
    125 :   1   =>  217.3808219 ;
    126 :   1   =>  266.9369862 ;
    127 :   1   =>  229.0410958 ;
    132 :   1   =>  251.5287671 ;
    134 :   1   =>  248.8635616 ;
    135 :   1   =>  209.4684931 ;
    136 :   1   =>  264.8547945 ;
    138 :   1   =>  293.1726027 ;
    139 :   1   =>  259.6076712 ;
    140 :   1   =>  329.819178  ;
    141 :   1   =>  319.8246575 ;
    142 :   1   =>  187.8136986 ;
    143 :   1   =>  427.7654794 ;
    147 :   1   =>  171.5726027 ;
    148 :   1   =>  303.9999999 ;
    150 :   1   =>  285.5101369 ;
    151 :   1   =>  235.2876712 ;
    154 :   1   =>  227.3753424 ;
    155 :   1   =>  284.8438356 ;
    158 :   1   =>  179.9013698 ;
    159 :   1   =>  249.8630136 ;
    160 :   1   =>  229.8739726 ;
    201 :   1   =>  274.849315  ;
    203 :   1   =>  235.4542465 ;
    204 :   1   =>  204.0547945 ;
    206 :   1   =>  170.739726  ;
    207 :   1   =>  182.899726  ;
    208 :   1   =>  166.5753424 ;
    209 :   1   =>  187.2306849 ;
    211 :   1   =>  199.8904109 ;
    213 :   1   =>  123.6821918 ;
    217 :   1   =>  152.249863  ;
    218 :   1   =>  195.7260274 ;
    219 :   1   =>  243.2832876 ;
    220 :   1   =>  294.8383561 ;
    221 :   1   =>  215.2986301 ;
    222 :   1   =>  183.649315  ;
    223 :   1   =>  310.1632876 ;
```

```
    224 :   1   =>  195.1430137 ;
    226 :   1   =>  226.1260273 ;
    227 :   1   =>  187.3972602 ;
    232 :   1   =>  188.0635616 ;
    234 :   1   =>  168.9073972 ;
    235 :   1   =>  166.1589041 ;
    236 :   1   =>  289.8410958 ;
    237 :   1   =>  206.1369863 ;
    238 :   1   =>  195.7260274 ;
    239 :   1   =>  201.9726027 ;
    240 :   1   =>  167.3249315 ;
    241 :   1   =>  219.4630137 ;
    246 :   1   =>  164.909589  ;
    247 :   1   =>  195.7260274 ;
    249 :   1   =>  254.0273972 ;
    250 :   1   =>  208.219178  ;
    252 :   1   =>  261.1068493 ;
    253 :   1   =>  278.7638356 ;
    254 :   1   =>  344.3945205 ;
    255 :   1   =>  236.9534246 ;
    256 :   1   =>  171.239452  ;
    258 :   1   =>  278.5972602 ;
    301 :   1   =>  179.9013698 ;
    302 :   1   =>  154.0821917 ;
    304 :   1   =>  137.4246575 ;
    305 :   1   =>  224.8767123 ;
    306 :   1   =>  311.4958903 ;
    308 :   1   =>  220.7123287 ;
    309 :   1   =>  436.4273972 ;
    310 :   1   =>  231.0399999 ;
    311 :   1   =>  174.9041096 ;
    318 :   1   =>  224.8767123 ;
    319 :   1   =>  732.9315067 ;
    320 :   1   =>  229.8739726 ;
    322 :   1   =>  309.8301369 ;
    323 :   1   =>  487.2328766 ;
    324 :   1   =>  238.2027397 ;
    326 :   1   =>  284.8438356 ;
    327 :   1   =>  417.2712328 ;
    328 :   1   =>  209.0520547 ;
    329 :   1   =>  151.1671233 ;
    332 :   1   =>  239.8684931 ;
    333 :   1   =>  255.5265753 ;
    336 :   1   =>  224.4602739 ;
    337 :   1   =>  174.0712328 ;
    338 :   1   =>  183.2328767 ;
    339 :   1   =>  330.5687671 ;
    340 :   1   =>  229.8739726 ;
    342 :   1   =>  319.8246575 ;
    343 :   1   =>  319.8246575 ;
    346 :   1   =>  458.49863   ;
    347 :   1   =>  493.0630136 ;
    348 :   1   =>  195.7260274 ;
    349 :   1   =>  302.3342465 ;
    350 :   1   =>  265.6876712 ;
    351 :   1   =>  261.1068493 ;
    352 :   1   =>  158.2465753 ;
    353 :   1   =>  367.2986301 ;
    354 :   1   =>  365.2164383 ;
    356 :   1   =>  238.2027397 ;
endrecode;

FE_AN_AMT = FE_AMOUNT * FE_MULTIPLIER;
FE_WGHT_AN_AMT = FE_AN_AMT * WEIGHT;
PROC FE_NUMBER

if ID07 in 121 and FE_LINE_NUMBER in 27 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 319 and FE_LINE_NUMBER in 48 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 306 and FE_LINE_NUMBER in 25 then FE_NUMBER =
```

```
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2.25); endif;
if ID07 in 121 and FE_LINE_NUMBER in 39 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2.5); endif;
if ID07 in 119 and FE_LINE_NUMBER in 26 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}4); endif;
if ID07 in 120 and FE_LINE_NUMBER in 13 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}4); endif;
if ID07 in 127 and FE_LINE_NUMBER in 63 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}4); endif;
if ID07 in 109 and FE_LINE_NUMBER in 54 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}4); endif;
if ID07 in 124 and FE_LINE_NUMBER in 52 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}5); endif;
if ID07 in 126 and FE_LINE_NUMBER in 28 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}5); endif;
if ID07 in 240 and FE_LINE_NUMBER in 83 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}5.5); endif;
if ID07 in 124 and FE_LINE_NUMBER in 39 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}6); endif;
if ID07 in 127 and FE_LINE_NUMBER in 60 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}6); endif;
if ID07 in 227 and FE_LINE_NUMBER in 50 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}6); endif;
if ID07 in 119 and FE_LINE_NUMBER in 28 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}6); endif;
if ID07 in 127 and FE_LINE_NUMBER in 38 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}8); endif;
if ID07 in 127 and FE_LINE_NUMBER in 45 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}8); endif;
if ID07 in 127 and FE_LINE_NUMBER in 54 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}8); endif;
if ID07 in 127 and FE_LINE_NUMBER in 66 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}10); endif;
if ID07 in 132 and FE_LINE_NUMBER in 66 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}10); endif;
if ID07 in 113 and FE_LINE_NUMBER in 40 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}10); endif;
if ID07 in 108 and FE_LINE_NUMBER in 48 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}15); endif;
if ID07 in 306 and FE_LINE_NUMBER in 23 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}20); endif;
if ID07 in 256 and FE_LINE_NUMBER in 46 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}45); endif;
if ID07 in 108 and FE_LINE_NUMBER in 58 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}50); endif;
if ID07 in 210 and FE_LINE_NUMBER in 40 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}50); endif;
if ID07 in 127 and FE_LINE_NUMBER in 40 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}50); endif;
if ID07 in 102 and FE_LINE_NUMBER in 19 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}60); endif;
if ID07 in 339 and FE_LINE_NUMBER in 48 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}60); endif;
if ID07 in 338 and FE_LINE_NUMBER in 28 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}60); endif;
if ID07 in 108 and FE_LINE_NUMBER in 66 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 332 and FE_LINE_NUMBER in 32 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 338 and FE_LINE_NUMBER in 36 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 121 and FE_LINE_NUMBER in 36 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 127 and FE_LINE_NUMBER in 73 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 109 and FE_LINE_NUMBER in 67 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 338 and FE_LINE_NUMBER in 35 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
if ID07 in 338 and FE_LINE_NUMBER in 44 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}100); endif;
```

```
if ID07 in 211 and FE_LINE_NUMBER in 33 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}180); endif;
if ID07 in 206 and FE_LINE_NUMBER in 36 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}180); endif;
if ID07 in 109 and FE_LINE_NUMBER in 58 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}200); endif;
if ID07 in 106 and FE_LINE_NUMBER in 30 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}200); endif;
if ID07 in 102 and FE_LINE_NUMBER in 38 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}200); endif;
if ID07 in 113 and FE_LINE_NUMBER in 38 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}200); endif;
if ID07 in 206 and FE_LINE_NUMBER in 31 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}240); endif;
if ID07 in 206 and FE_LINE_NUMBER in 24 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}240); endif;
if ID07 in 102 and FE_LINE_NUMBER in 25 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}250); endif;
if ID07 in 125 and FE_LINE_NUMBER in 50 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}325); endif;
if ID07 in 255 and FE_LINE_NUMBER in 24 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}375); endif;
if ID07 in 124 and FE_LINE_NUMBER in 57 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}390); endif;
if ID07 in 109 and FE_LINE_NUMBER in 27 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}400); endif;
if ID07 in 109 and FE_LINE_NUMBER in 49 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}400); endif;
if ID07 in 108 and FE_LINE_NUMBER in 67 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 108 and FE_LINE_NUMBER in 71 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 105 and FE_LINE_NUMBER in 27 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 217 and FE_LINE_NUMBER in 44 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 136 and FE_LINE_NUMBER in 19 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 109 and FE_LINE_NUMBER in 48 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}500); endif;
if ID07 in 109 and FE_LINE_NUMBER in 50 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}555); endif;
if ID07 in 127 and FE_LINE_NUMBER in 53 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}600); endif;
if ID07 in 102 and FE_LINE_NUMBER in 29 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}600); endif;
if ID07 in 111 and FE_LINE_NUMBER in 44 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}600); endif;
if ID07 in 109 and FE_LINE_NUMBER in 25 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}650); endif;
if ID07 in 206 and FE_LINE_NUMBER in 27 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}900); endif;
if ID07 in 105 and FE_LINE_NUMBER in 26 then FE_NUMBER =
setNumFlag({questName}'FE_NUMBER',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}3.6); endif;

if ID07 in 211 and FE_LINE_NUMBER in 33 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 206 and FE_LINE_NUMBER in 36 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 224 and FE_LINE_NUMBER in 67 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 208 and FE_LINE_NUMBER in 53 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 121 and FE_LINE_NUMBER in 62 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 127 and FE_LINE_NUMBER in 70 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 121 and FE_LINE_NUMBER in 46 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}1); endif;
if ID07 in 306 and FE_LINE_NUMBER in 25 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
```

```
if ID07 in 306 and FE_LINE_NUMBER in 23 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 338 and FE_LINE_NUMBER in 25 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 111 and FE_LINE_NUMBER in 24 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 116 and FE_LINE_NUMBER in 42 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 117 and FE_LINE_NUMBER in 45 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 332 and FE_LINE_NUMBER in 31 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}2); endif;
if ID07 in 108 and FE_LINE_NUMBER in 72 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}4); endif;
if ID07 in 108 and FE_LINE_NUMBER in 48 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}7); endif;
if ID07 in 125 and FE_LINE_NUMBER in 54 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}7); endif;
if ID07 in 104 and FE_LINE_NUMBER in 33 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}12); endif;
if ID07 in 119 and FE_LINE_NUMBER in 26 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 120 and FE_LINE_NUMBER in 13 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 63 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 124 and FE_LINE_NUMBER in 39 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 60 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 38 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 45 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 54 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 66 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 127 and FE_LINE_NUMBER in 36 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}33); endif;
if ID07 in 124 and FE_LINE_NUMBER in 52 then FE_UNIT =
setNumFlag({questName}'FE_UNIT',{lineNo}FE_LINE_NUMBER,{imputeType}4,{newVal}38); endif;

PROC FE_UNIT

PROC SAMPLE_WEIGHT
SAMPLE_WEIGHT = WEIGHT;     {Weight}
PROC SAMPLE_SET
SAMPLE_SET = CV2;       {Sample Set A, B, or C}
PROC CENSUS_CODE
CENSUS_CODE = CV5B;     {Census Code}
PROC ISLAND_CODE
ISLAND_CODE = CV11;     {Island Code}
PROC VILLAGE_CODE
VILLAGE_CODE = CV13;        {Village Code}
PROC MODULE_START_DATE_DD
MODULE_START_DATE_DD = CV18A;       {Module Start Date - Day}
PROC MODULE_START_DATE_MM
MODULE_START_DATE_MM = CV18B;       {Module Start Date - Month}
PROC MODULE_START_DATE_YYYY
MODULE_START_DATE_YYYY = CV18C;     {Module Start Date - Year}
PROC DIARY_START_DATE_DD
DIARY_START_DATE_DD = S500_D_A; {Diary Start Date - Day}
PROC DIARY_START_DATE_MM
DIARY_START_DATE_MM = S500_M_A;     {Diary Start Date - Month}
PROC DIARY_START_DATE_YYYY
DIARY_START_DATE_YYYY   = S500_Y_A;     {Diary Start Date - Year}
PROC C_HH_INC_QURT_G_TOT
recode  ID07    =>  $   ;
    120 =>  1   ;
    247 =>  1   ;
```

```
114 =>  1   ;
246 =>  1   ;
143 =>  1   ;
348 =>  1   ;
302 =>  1   ;
222 =>  1   ;
308 =>  1   ;
207 =>  1   ;
142 =>  1   ;
104 =>  1   ;
140 =>  1   ;
337 =>  1   ;
329 =>  1   ;
136 =>  1   ;
336 =>  1   ;
351 =>  1   ;
138 =>  1   ;
305 =>  1   ;
232 =>  1   ;
106 =>  1   ;
201 =>  1   ;
333 =>  1   ;
356 =>  1   ;
256 =>  1   ;
158 =>  1   ;
102 =>  1   ;
105 =>  1   ;
255 =>  2   ;
220 =>  2   ;
235 =>  2   ;
147 =>  2   ;
211 =>  2   ;
338 =>  2   ;
349 =>  2   ;
301 =>  2   ;
237 =>  2   ;
139 =>  2   ;
204 =>  2   ;
160 =>  2   ;
310 =>  2   ;
209 =>  2   ;
213 =>  2   ;
304 =>  2   ;
141 =>  2   ;
350 =>  2   ;
353 =>  2   ;
113 =>  2   ;
155 =>  2   ;
103 =>  2   ;
206 =>  2   ;
352 =>  2   ;
159 =>  2   ;
221 =>  2   ;
307 =>  2   ;
226 =>  2   ;
318 =>  2   ;
250 =>  2   ;
210 =>  3   ;
240 =>  3   ;
306 =>  3   ;
324 =>  3   ;
218 =>  3   ;
121 =>  3   ;
227 =>  3   ;
148 =>  3   ;
339 =>  3   ;
323 =>  3   ;
258 =>  3   ;
311 =>  3   ;
108 =>  3   ;
124 =>  3   ;
```

```
    234 =>  3   ;
    328 =>  3   ;
    219 =>  3   ;
    116 =>  3   ;
    208 =>  3   ;
    249 =>  3   ;
    342 =>  3   ;
    346 =>  3   ;
    238 =>  3   ;
    125 =>  3   ;
    109 =>  3   ;
    322 =>  3   ;
    253 =>  3   ;
    341 =>  3   ;
    332 =>  3   ;
    151 =>  3   ;
    135 =>  4   ;
    134 =>  4   ;
    319 =>  4   ;
    154 =>  4   ;
    132 =>  4   ;
    118 =>  4   ;
    309 =>  4   ;
    233 =>  4   ;
    320 =>  4   ;
    111 =>  4   ;
    117 =>  4   ;
    150 =>  4   ;
    347 =>  4   ;
    203 =>  4   ;
    217 =>  4   ;
    241 =>  4   ;
    239 =>  4   ;
    252 =>  4   ;
    254 =>  4   ;
    119 =>  4   ;
    223 =>  4   ;
    126 =>  4   ;
    343 =>  4   ;
    327 =>  4   ;
    326 =>  4   ;
    340 =>  4   ;
    224 =>  4   ;
    236 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_INC_QURT_G_CASH
recode  ID07    =>  $   ;
    120 =>  1   ;
    143 =>  1   ;
    247 =>  1   ;
    220 =>  1   ;
    104 =>  1   ;
    114 =>  1   ;
    308 =>  1   ;
    246 =>  1   ;
    348 =>  1   ;
    140 =>  1   ;
    136 =>  1   ;
    337 =>  1   ;
    138 =>  1   ;
    305 =>  1   ;
    222 =>  1   ;
    302 =>  1   ;
    207 =>  1   ;
    232 =>  1   ;
    142 =>  1   ;
    351 =>  1   ;
    201 =>  1   ;
    336 =>  1   ;
```

```
329 =>  1   ;
333 =>  1   ;
106 =>  1   ;
356 =>  1   ;
158 =>  1   ;
256 =>  1   ;
349 =>  1   ;
102 =>  2   ;
155 =>  2   ;
255 =>  2   ;
105 =>  2   ;
139 =>  2   ;
310 =>  2   ;
235 =>  2   ;
338 =>  2   ;
211 =>  2   ;
147 =>  2   ;
353 =>  2   ;
237 =>  2   ;
301 =>  2   ;
113 =>  2   ;
160 =>  2   ;
206 =>  2   ;
141 =>  2   ;
350 =>  2   ;
103 =>  2   ;
204 =>  2   ;
209 =>  2   ;
304 =>  2   ;
213 =>  2   ;
159 =>  2   ;
118 =>  2   ;
352 =>  2   ;
221 =>  2   ;
226 =>  2   ;
148 =>  2   ;
347 =>  2   ;
307 =>  3   ;
250 =>  3   ;
324 =>  3   ;
318 =>  3   ;
306 =>  3   ;
116 =>  3   ;
323 =>  3   ;
240 =>  3   ;
328 =>  3   ;
121 =>  3   ;
210 =>  3   ;
124 =>  3   ;
339 =>  3   ;
108 =>  3   ;
258 =>  3   ;
218 =>  3   ;
249 =>  3   ;
227 =>  3   ;
311 =>  3   ;
322 =>  3   ;
346 =>  3   ;
319 =>  3   ;
203 =>  3   ;
132 =>  3   ;
208 =>  3   ;
219 =>  3   ;
234 =>  3   ;
342 =>  3   ;
125 =>  3   ;
109 =>  3   ;
253 =>  4   ;
135 =>  4   ;
154 =>  4   ;
134 =>  4   ;
```

```
    151 =>  4   ;
    238 =>  4   ;
    241 =>  4   ;
    332 =>  4   ;
    341 =>  4   ;
    309 =>  4   ;
    320 =>  4   ;
    111 =>  4   ;
    126 =>  4   ;
    117 =>  4   ;
    150 =>  4   ;
    233 =>  4   ;
    254 =>  4   ;
    119 =>  4   ;
    239 =>  4   ;
    252 =>  4   ;
    217 =>  4   ;
    223 =>  4   ;
    327 =>  4   ;
    343 =>  4   ;
    326 =>  4   ;
    340 =>  4   ;
    236 =>  4   ;
    224 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_INC_QURT_G_TOT_LESS_RENT
recode  ID07    =>  $   ;
    120 =>  1   ;
    143 =>  1   ;
    247 =>  1   ;
    114 =>  1   ;
    246 =>  1   ;
    140 =>  1   ;
    348 =>  1   ;
    104 =>  1   ;
    308 =>  1   ;
    222 =>  1   ;
    302 =>  1   ;
    138 =>  1   ;
    207 =>  1   ;
    142 =>  1   ;
    136 =>  1   ;
    351 =>  1   ;
    201 =>  1   ;
    336 =>  1   ;
    305 =>  1   ;
    337 =>  1   ;
    333 =>  1   ;
    232 =>  1   ;
    329 =>  1   ;
    106 =>  1   ;
    356 =>  1   ;
    256 =>  1   ;
    102 =>  1   ;
    158 =>  1   ;
    220 =>  1   ;
    255 =>  2   ;
    105 =>  2   ;
    349 =>  2   ;
    235 =>  2   ;
    147 =>  2   ;
    211 =>  2   ;
    338 =>  2   ;
    139 =>  2   ;
    113 =>  2   ;
    301 =>  2   ;
    237 =>  2   ;
    141 =>  2   ;
    160 =>  2   ;
```

```
204 =>  2   ;
310 =>  2   ;
353 =>  2   ;
209 =>  2   ;
350 =>  2   ;
155 =>  2   ;
213 =>  2   ;
304 =>  2   ;
103 =>  2   ;
206 =>  2   ;
159 =>  2   ;
221 =>  2   ;
352 =>  2   ;
226 =>  2   ;
318 =>  2   ;
250 =>  2   ;
307 =>  2   ;
306 =>  3   ;
240 =>  3   ;
323 =>  3   ;
324 =>  3   ;
148 =>  3   ;
210 =>  3   ;
121 =>  3   ;
218 =>  3   ;
339 =>  3   ;
227 =>  3   ;
258 =>  3   ;
108 =>  3   ;
124 =>  3   ;
311 =>  3   ;
219 =>  3   ;
328 =>  3   ;
346 =>  3   ;
319 =>  3   ;
116 =>  3   ;
234 =>  3   ;
249 =>  3   ;
208 =>  3   ;
342 =>  3   ;
322 =>  3   ;
253 =>  3   ;
125 =>  3   ;
238 =>  3   ;
332 =>  3   ;
151 =>  3   ;
109 =>  3   ;
135 =>  4   ;
134 =>  4   ;
309 =>  4   ;
341 =>  4   ;
132 =>  4   ;
154 =>  4   ;
118 =>  4   ;
347 =>  4   ;
320 =>  4   ;
233 =>  4   ;
150 =>  4   ;
111 =>  4   ;
203 =>  4   ;
117 =>  4   ;
241 =>  4   ;
217 =>  4   ;
239 =>  4   ;
254 =>  4   ;
252 =>  4   ;
119 =>  4   ;
223 =>  4   ;
126 =>  4   ;
343 =>  4   ;
327 =>  4   ;
```

```
    326 =>  4   ;
    340 =>  4   ;
    224 =>  4   ;
    236 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_INC_QURT_N_TOT
recode  ID07    =>  $   ;
    120 =>  1   ;
    247 =>  1   ;
    143 =>  1   ;
    114 =>  1   ;
    246 =>  1   ;
    302 =>  1   ;
    348 =>  1   ;
    222 =>  1   ;
    308 =>  1   ;
    201 =>  1   ;
    104 =>  1   ;
    106 =>  1   ;
    207 =>  1   ;
    142 =>  1   ;
    140 =>  1   ;
    256 =>  1   ;
    138 =>  1   ;
    136 =>  1   ;
    337 =>  1   ;
    329 =>  1   ;
    336 =>  1   ;
    351 =>  1   ;
    305 =>  1   ;
    232 =>  1   ;
    333 =>  1   ;
    356 =>  1   ;
    102 =>  1   ;
    105 =>  1   ;
    158 =>  1   ;
    235 =>  2   ;
    147 =>  2   ;
    255 =>  2   ;
    220 =>  2   ;
    211 =>  2   ;
    338 =>  2   ;
    349 =>  2   ;
    301 =>  2   ;
    237 =>  2   ;
    139 =>  2   ;
    204 =>  2   ;
    160 =>  2   ;
    310 =>  2   ;
    304 =>  2   ;
    213 =>  2   ;
    209 =>  2   ;
    141 =>  2   ;
    113 =>  2   ;
    350 =>  2   ;
    155 =>  2   ;
    353 =>  2   ;
    103 =>  2   ;
    206 =>  2   ;
    352 =>  2   ;
    159 =>  2   ;
    226 =>  2   ;
    318 =>  2   ;
    221 =>  2   ;
    307 =>  2   ;
    250 =>  2   ;
    240 =>  3   ;
    210 =>  3   ;
    306 =>  3   ;
```

```
    227 =>  3   ;
    323 =>  3   ;
    324 =>  3   ;
    121 =>  3   ;
    218 =>  3   ;
    148 =>  3   ;
    108 =>  3   ;
    258 =>  3   ;
    339 =>  3   ;
    311 =>  3   ;
    124 =>  3   ;
    234 =>  3   ;
    116 =>  3   ;
    328 =>  3   ;
    208 =>  3   ;
    249 =>  3   ;
    219 =>  3   ;
    322 =>  3   ;
    238 =>  3   ;
    342 =>  3   ;
    341 =>  3   ;
    253 =>  3   ;
    346 =>  3   ;
    125 =>  3   ;
    109 =>  3   ;
    134 =>  3   ;
    151 =>  3   ;
    332 =>  4   ;
    135 =>  4   ;
    319 =>  4   ;
    154 =>  4   ;
    132 =>  4   ;
    309 =>  4   ;
    117 =>  4   ;
    118 =>  4   ;
    233 =>  4   ;
    111 =>  4   ;
    320 =>  4   ;
    239 =>  4   ;
    150 =>  4   ;
    347 =>  4   ;
    203 =>  4   ;
    217 =>  4   ;
    241 =>  4   ;
    254 =>  4   ;
    119 =>  4   ;
    252 =>  4   ;
    223 =>  4   ;
    126 =>  4   ;
    343 =>  4   ;
    327 =>  4   ;
    326 =>  4   ;
    340 =>  4   ;
    224 =>  4   ;
    236 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_INC_QURT_N_CASH
recode  ID07    =>  $   ;
    120 =>  1   ;
    143 =>  1   ;
    247 =>  1   ;
    104 =>  1   ;
    220 =>  1   ;
    114 =>  1   ;
    140 =>  1   ;
    201 =>  1   ;
    308 =>  1   ;
    246 =>  1   ;
    138 =>  1   ;
```

```
    136 =>  1   ;
    348 =>  1   ;
    302 =>  1   ;
    337 =>  1   ;
    305 =>  1   ;
    222 =>  1   ;
    207 =>  1   ;
    232 =>  1   ;
    106 =>  1   ;
    142 =>  1   ;
    351 =>  1   ;
    256 =>  1   ;
    336 =>  1   ;
    329 =>  1   ;
    333 =>  1   ;
    356 =>  1   ;
    158 =>  1   ;
    349 =>  1   ;
    102 =>  2   ;
    105 =>  2   ;
    155 =>  2   ;
    235 =>  2   ;
    139 =>  2   ;
    255 =>  2   ;
    310 =>  2   ;
    147 =>  2   ;
    338 =>  2   ;
    237 =>  2   ;
    211 =>  2   ;
    301 =>  2   ;
    353 =>  2   ;
    113 =>  2   ;
    160 =>  2   ;
    141 =>  2   ;
    206 =>  2   ;
    204 =>  2   ;
    103 =>  2   ;
    350 =>  2   ;
    209 =>  2   ;
    304 =>  2   ;
    213 =>  2   ;
    159 =>  2   ;
    352 =>  2   ;
    323 =>  2   ;
    226 =>  2   ;
    118 =>  2   ;
    318 =>  2   ;
    221 =>  2   ;
    148 =>  3   ;
    347 =>  3   ;
    306 =>  3   ;
    250 =>  3   ;
    116 =>  3   ;
    240 =>  3   ;
    324 =>  3   ;
    307 =>  3   ;
    328 =>  3   ;
    124 =>  3   ;
    121 =>  3   ;
    108 =>  3   ;
    210 =>  3   ;
    227 =>  3   ;
    258 =>  3   ;
    339 =>  3   ;
    249 =>  3   ;
    322 =>  3   ;
    218 =>  3   ;
    311 =>  3   ;
    319 =>  3   ;
    346 =>  3   ;
    132 =>  3   ;
```

```
    208 =>  3   ;
    203 =>  3   ;
    234 =>  3   ;
    219 =>  3   ;
    134 =>  3   ;
    253 =>  3   ;
    154 =>  3   ;
    135 =>  4   ;
    342 =>  4   ;
    125 =>  4   ;
    109 =>  4   ;
    238 =>  4   ;
    151 =>  4   ;
    341 =>  4   ;
    241 =>  4   ;
    332 =>  4   ;
    309 =>  4   ;
    320 =>  4   ;
    117 =>  4   ;
    126 =>  4   ;
    111 =>  4   ;
    233 =>  4   ;
    239 =>  4   ;
    150 =>  4   ;
    119 =>  4   ;
    254 =>  4   ;
    252 =>  4   ;
    217 =>  4   ;
    223 =>  4   ;
    327 =>  4   ;
    343 =>  4   ;
    326 =>  4   ;
    340 =>  4   ;
    236 =>  4   ;
    224 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_INC_QURT_N_TOT_LESS_RENT
recode  ID07    =>  $   ;
    120 =>  1   ;
    143 =>  1   ;
    247 =>  1   ;
    114 =>  1   ;
    104 =>  1   ;
    140 =>  1   ;
    201 =>  1   ;
    246 =>  1   ;
    348 =>  1   ;
    308 =>  1   ;
    302 =>  1   ;
    222 =>  1   ;
    138 =>  1   ;
    136 =>  1   ;
    106 =>  1   ;
    207 =>  1   ;
    142 =>  1   ;
    351 =>  1   ;
    256 =>  1   ;
    336 =>  1   ;
    305 =>  1   ;
    337 =>  1   ;
    333 =>  1   ;
    232 =>  1   ;
    329 =>  1   ;
    356 =>  1   ;
    102 =>  1   ;
    105 =>  1   ;
    158 =>  1   ;
    235 =>  2   ;
    220 =>  2   ;
```

```
255 =>  2   ;
349 =>  2   ;
147 =>  2   ;
211 =>  2   ;
139 =>  2   ;
338 =>  2   ;
113 =>  2   ;
237 =>  2   ;
301 =>  2   ;
141 =>  2   ;
160 =>  2   ;
310 =>  2   ;
204 =>  2   ;
353 =>  2   ;
209 =>  2   ;
155 =>  2   ;
304 =>  2   ;
350 =>  2   ;
213 =>  2   ;
103 =>  2   ;
206 =>  2   ;
159 =>  2   ;
352 =>  2   ;
226 =>  2   ;
318 =>  2   ;
221 =>  2   ;
323 =>  2   ;
250 =>  2   ;
306 =>  3   ;
240 =>  3   ;
307 =>  3   ;
227 =>  3   ;
324 =>  3   ;
121 =>  3   ;
148 =>  3   ;
210 =>  3   ;
339 =>  3   ;
218 =>  3   ;
258 =>  3   ;
108 =>  3   ;
124 =>  3   ;
311 =>  3   ;
319 =>  3   ;
116 =>  3   ;
249 =>  3   ;
234 =>  3   ;
328 =>  3   ;
322 =>  3   ;
219 =>  3   ;
208 =>  3   ;
346 =>  3   ;
342 =>  3   ;
253 =>  3   ;
238 =>  3   ;
125 =>  3   ;
134 =>  3   ;
151 =>  3   ;
332 =>  3   ;
109 =>  4   ;
135 =>  4   ;
341 =>  4   ;
309 =>  4   ;
154 =>  4   ;
132 =>  4   ;
118 =>  4   ;
347 =>  4   ;
117 =>  4   ;
320 =>  4   ;
233 =>  4   ;
150 =>  4   ;
239 =>  4   ;
```

```
    111 =>  4   ;
    203 =>  4   ;
    241 =>  4   ;
    217 =>  4   ;
    254 =>  4   ;
    119 =>  4   ;
    223 =>  4   ;
    252 =>  4   ;
    126 =>  4   ;
    343 =>  4   ;
    327 =>  4   ;
    326 =>  4   ;
    340 =>  4   ;
    224 =>  4   ;
    236 =>  4   ;
    354 =>  4   ;
    127 =>  4   ;
endrecode;
PROC C_HH_EXP_QURT_TOT
{Calculated - Expenditure Deciles (Total)}
recode  ID07    =>  $   ;
    348 =>  1   ;
    302 =>  1   ;
    207 =>  1   ;
    351 =>  1   ;
    120 =>  1   ;
    337 =>  1   ;
    204 =>  1   ;
    329 =>  1   ;
    336 =>  1   ;
    356 =>  1   ;
    350 =>  1   ;
    235 =>  1   ;
    246 =>  1   ;
    247 =>  1   ;
    305 =>  1   ;
    213 =>  1   ;
    142 =>  1   ;
    304 =>  1   ;
    158 =>  1   ;
    114 =>  1   ;
    232 =>  1   ;
    211 =>  1   ;
    333 =>  1   ;
    237 =>  1   ;
    222 =>  1   ;
    255 =>  1   ;
    136 =>  1   ;
    206 =>  1   ;
    338 =>  1   ;
    307 =>  1   ;
    139 =>  2   ;
    138 =>  2   ;
    140 =>  2   ;
    209 =>  2   ;
    349 =>  2   ;
    218 =>  2   ;
    102 =>  2   ;
    141 =>  2   ;
    147 =>  2   ;
    308 =>  2   ;
    220 =>  2   ;
    353 =>  2   ;
    221 =>  2   ;
    159 =>  2   ;
    310 =>  2   ;
    306 =>  2   ;
    253 =>  2   ;
    226 =>  2   ;
    301 =>  2   ;
    311 =>  2   ;
```

```
    339 =>  2   ;
    104 =>  2   ;
    324 =>  2   ;
    208 =>  2   ;
    124 =>  2   ;
    227 =>  2   ;
    318 =>  2   ;
    252 =>  2   ;
    105 =>  2   ;
    151 =>  2   ;
    322 =>  3   ;
    103 =>  3   ;
    106 =>  3   ;
    160 =>  3   ;
    148 =>  3   ;
    258 =>  3   ;
    121 =>  3   ;
    135 =>  3   ;
    328 =>  3   ;
    116 =>  3   ;
    143 =>  3   ;
    217 =>  3   ;
    117 =>  3   ;
    309 =>  3   ;
    113 =>  3   ;
    323 =>  3   ;
    341 =>  3   ;
    119 =>  3   ;
    111 =>  3   ;
    352 =>  3   ;
    210 =>  3   ;
    125 =>  3   ;
    256 =>  3   ;
    320 =>  3   ;
    240 =>  3   ;
    127 =>  3   ;
    343 =>  3   ;
    150 =>  3   ;
    346 =>  3   ;
    154 =>  4   ;
    340 =>  4   ;
    201 =>  4   ;
    118 =>  4   ;
    233 =>  4   ;
    234 =>  4   ;
    126 =>  4   ;
    238 =>  4   ;
    254 =>  4   ;
    249 =>  4   ;
    203 =>  4   ;
    239 =>  4   ;
    332 =>  4   ;
    250 =>  4   ;
    326 =>  4   ;
    319 =>  4   ;
    241 =>  4   ;
    354 =>  4   ;
    347 =>  4   ;
    327 =>  4   ;
    223 =>  4   ;
    342 =>  4   ;
    109 =>  4   ;
    219 =>  4   ;
    155 =>  4   ;
    236 =>  4   ;
    108 =>  4   ;
    132 =>  4   ;
    224 =>  4   ;
    134 =>  4   ;
endrecode;
PROC C_HH_EXP_QURT_CASH
```

```
{Calculated - Expenditure Deciles (Cash + In-kind)}
recode  ID07    =>  $   ;
    348 =>  1   ;
    351 =>  1   ;
    302 =>  1   ;
    337 =>  1   ;
    207 =>  1   ;
    120 =>  1   ;
    350 =>  1   ;
    204 =>  1   ;
    336 =>  1   ;
    329 =>  1   ;
    356 =>  1   ;
    305 =>  1   ;
    235 =>  1   ;
    246 =>  1   ;
    247 =>  1   ;
    232 =>  1   ;
    206 =>  1   ;
    158 =>  1   ;
    142 =>  1   ;
    304 =>  1   ;
    213 =>  1   ;
    333 =>  1   ;
    211 =>  1   ;
    220 =>  1   ;
    114 =>  1   ;
    139 =>  1   ;
    237 =>  1   ;
    136 =>  1   ;
    349 =>  1   ;
    138 =>  1   ;
    255 =>  2   ;
    338 =>  2   ;
    222 =>  2   ;
    140 =>  2   ;
    353 =>  2   ;
    307 =>  2   ;
    209 =>  2   ;
    141 =>  2   ;
    118 =>  2   ;
    218 =>  2   ;
    102 =>  2   ;
    310 =>  2   ;
    253 =>  2   ;
    308 =>  2   ;
    306 =>  2   ;
    159 =>  2   ;
    116 =>  2   ;
    124 =>  2   ;
    147 =>  2   ;
    322 =>  2   ;
    328 =>  2   ;
    221 =>  2   ;
    324 =>  2   ;
    339 =>  2   ;
    226 =>  2   ;
    311 =>  2   ;
    148 =>  2   ;
    252 =>  2   ;
    104 =>  2   ;
    301 =>  2   ;
    208 =>  3   ;
    135 =>  3   ;
    103 =>  3   ;
    151 =>  3   ;
    119 =>  3   ;
    258 =>  3   ;
    318 =>  3   ;
    323 =>  3   ;
    227 =>  3   ;
```

```
    121 =>  3   ;
    203 =>  3   ;
    347 =>  3   ;
    160 =>  3   ;
    143 =>  3   ;
    105 =>  3   ;
    309 =>  3   ;
    113 =>  3   ;
    106 =>  3   ;
    126 =>  3   ;
    117 =>  3   ;
    320 =>  3   ;
    111 =>  3   ;
    217 =>  3   ;
    125 =>  3   ;
    154 =>  3   ;
    346 =>  3   ;
    341 =>  3   ;
    352 =>  3   ;
    343 =>  3   ;
    127 =>  4   ;
    210 =>  4   ;
    256 =>  4   ;
    150 =>  4   ;
    240 =>  4   ;
    241 =>  4   ;
    340 =>  4   ;
    249 =>  4   ;
    201 =>  4   ;
    254 =>  4   ;
    354 =>  4   ;
    319 =>  4   ;
    234 =>  4   ;
    238 =>  4   ;
    233 =>  4   ;
    239 =>  4   ;
    332 =>  4   ;
    326 =>  4   ;
    250 =>  4   ;
    327 =>  4   ;
    155 =>  4   ;
    223 =>  4   ;
    109 =>  4   ;
    342 =>  4   ;
    236 =>  4   ;
    219 =>  4   ;
    132 =>  4   ;
    108 =>  4   ;
    224 =>  4   ;
    134 =>  4   ;
endrecode;
PROC C_HH_EXP_QURT_TOT_LESS_RENT
{Calculated - Expenditure Deciles (Total less rent)}
recode  ID07    =>  $   ;
    348 =>  1   ;
    351 =>  1   ;
    302 =>  1   ;
    207 =>  1   ;
    120 =>  1   ;
    337 =>  1   ;
    204 =>  1   ;
    336 =>  1   ;
    350 =>  1   ;
    356 =>  1   ;
    329 =>  1   ;
    235 =>  1   ;
    246 =>  1   ;
    305 =>  1   ;
    247 =>  1   ;
    142 =>  1   ;
    213 =>  1   ;
```

```
333 =>  1   ;
158 =>  1   ;
232 =>  1   ;
211 =>  1   ;
114 =>  1   ;
304 =>  1   ;
136 =>  1   ;
237 =>  1   ;
255 =>  1   ;
222 =>  1   ;
138 =>  1   ;
139 =>  1   ;
140 =>  1   ;
206 =>  2   ;
338 =>  2   ;
349 =>  2   ;
209 =>  2   ;
141 =>  2   ;
218 =>  2   ;
102 =>  2   ;
307 =>  2   ;
353 =>  2   ;
220 =>  2   ;
147 =>  2   ;
308 =>  2   ;
159 =>  2   ;
221 =>  2   ;
306 =>  2   ;
253 =>  2   ;
310 =>  2   ;
226 =>  2   ;
339 =>  2   ;
301 =>  2   ;
104 =>  2   ;
311 =>  2   ;
324 =>  2   ;
124 =>  2   ;
252 =>  2   ;
318 =>  2   ;
208 =>  2   ;
227 =>  2   ;
322 =>  2   ;
151 =>  2   ;
148 =>  3   ;
143 =>  3   ;
258 =>  3   ;
105 =>  3   ;
160 =>  3   ;
113 =>  3   ;
121 =>  3   ;
106 =>  3   ;
323 =>  3   ;
103 =>  3   ;
309 =>  3   ;
135 =>  3   ;
328 =>  3   ;
116 =>  3   ;
217 =>  3   ;
119 =>  3   ;
117 =>  3   ;
352 =>  3   ;
125 =>  3   ;
346 =>  3   ;
341 =>  3   ;
320 =>  3   ;
256 =>  3   ;
111 =>  3   ;
343 =>  3   ;
127 =>  3   ;
150 =>  3   ;
210 =>  3   ;
```

```
    240 =>  3   ;
    154 =>  4   ;
    201 =>  4   ;
    340 =>  4   ;
    118 =>  4   ;
    126 =>  4   ;
    234 =>  4   ;
    319 =>  4   ;
    254 =>  4   ;
    238 =>  4   ;
    233 =>  4   ;
    249 =>  4   ;
    203 =>  4   ;
    332 =>  4   ;
    239 =>  4   ;
    250 =>  4   ;
    326 =>  4   ;
    354 =>  4   ;
    347 =>  4   ;
    241 =>  4   ;
    327 =>  4   ;
    223 =>  4   ;
    342 =>  4   ;
    219 =>  4   ;
    109 =>  4   ;
    155 =>  4   ;
    236 =>  4   ;
    108 =>  4   ;
    132 =>  4   ;
    224 =>  4   ;
    134 =>  4   ;
endrecode;
PROC C_HH_TOT_EXP
C_HH_TOT_EXP = sum(FE_AN_AMT);
PROC C_HH_PC_EXP
C_HH_PC_EXP = sum(FE_AN_AMT) / count(S1010_EDT where sex in 1:2);
PROC C_HH_TOT_INC
C_HH_TOT_INC = sum(FI_AN_AMOUNT);
PROC C_HH_PC_INC
C_HH_PC_INC = sum(FI_AN_AMOUNT) / count(S1010_EDT where sex in 1:2);
PROC C_HH_COMPOSITION
//C_HH_COMPOSITION      {Calculated - Household Composition}
if count(S1010_EDT where relate in 2) >= 1 then
    C_HH_COMPOSITION = 5; // husband and wife with other
    if count(S1010_EDT) = 2 then
        C_HH_COMPOSITION = 1;
    endif;
    if count(S1010_EDT where relate in 3,7 and age in 0:5) >= 1 then
        C_HH_COMPOSITION = 2;
    endif;
    if count(S1010_EDT where relate in 3,7 and age in 6:17) >= 1 then
        C_HH_COMPOSITION = 3;
    endif;
    if count(S1010_EDT where relate in 3,7 and age in 18:95) >= 1 then
        C_HH_COMPOSITION = 4;
    endif;
else
    if count(S1010_EDT where relate in 3,7 and age in 0:17) >= 1 then
        C_HH_COMPOSITION = 6;
    else
        C_HH_COMPOSITION = 7;
    endif;
endif;
PROC C_HH_TOT_PER
C_HH_TOT_PER = count(S1010_EDT where sex in 1:2);   {Calculated - Total Number of Persons in
the Household}
PROC C_HH_TOT_MALES
C_HH_TOT_MALES = count(S1010_EDT where sex in 1);   {Calculated - Total Number of Males in
the Household}
PROC C_HH_TOT_FEMALES
C_HH_TOT_FEMALES = count(S1010_EDT where sex in 2);     {Calculated - Total Number of
```

```
Females in the Household}
PROC C_HH_TOT_PER_LESS_THAN_6
C_HH_TOT_PER_LESS_THAN_6 = count(S1010_EDT where age in 0:5);       {Calculated - Total
Number of Persons < 6 years of age}
PROC C_HH_TOT_PER_6_TO_17
C_HH_TOT_PER_6_TO_17 = count(S1010_EDT where age in 6:17);      {Calculated - Total Number
of Persons 6 years to 17 years of age}
PROC C_HH_TOT_PER_LESS_THAN_18
C_HH_TOT_PER_LESS_THAN_18 = count(S1010_EDT where age in 0:18);     {Calculated - Total
Number of Persons < 18 years of age}
PROC C_HH_TOT_PER_18_AND_OVER
C_HH_TOT_PER_18_AND_OVER = count(S1010_EDT where age in 18:99);     {Calculated - Total
Number of Persons >= 18 years of age}
PROC C_HH_TOT_PER_LESS_THAN_15
C_HH_TOT_PER_LESS_THAN_15 = count(S1010_EDT where age in 0:14);     {Calculated - Total
Number of Persons < 15 years of age}
PROC C_HH_TOT_PER_15_AND_OVER
C_HH_TOT_PER_15_AND_OVER = count(S1010_EDT where age in 15:99);     {Calculated - Total
Number of Persons >= 15 years of age}
PROC C_HH_TOT_PER_LESS_THAN_64
C_HH_TOT_PER_LESS_THAN_64 = count(S1010_EDT where age in 0:63);     {Calculated - Total
Number of Persons < 64 years of age}
PROC C_HH_TOT_PER_64_AND_OVER
C_HH_TOT_PER_64_AND_OVER = count(S1010_EDT where age in 64:99);     {Calculated - Total
Number of Persons >= 64 years of age}
PROC HC_Q20111
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20111_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20111 <> 6 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20112A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20112A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20112A <> 5 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20112B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20112B_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20112B <> 5 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20112C
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20112C_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20112C <> 4 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20113
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20113_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20113 <> 5 then
```

```
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20114
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20114_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20114 <> 5 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20115
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20116
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20117
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20121
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20122
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20123
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20123_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20123 <> 4 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20124
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20124_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20124 <> 6 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20125
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if hc_q20124 = hc_q20125 then
    impute (hc_q20125, 7);
    errmsg("DOUBLE");
endif;
PROC HC_Q20125_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20125 <> 6 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20126
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20126_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20126 <> 8 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20127
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
```

```
if hc_q20126 = hc_q20127 then
    impute (hc_q20127, 9);
    errmsg("DOUBLE");
endif;
PROC HC_Q20127_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20127 <> 8 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20131_1A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20131_1A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_1A <> 7 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20131_1B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if hc_q20131_1A = hc_q20131_1B then
    impute (hc_q20131_1B, 8);
    errmsg("DOUBLE");
endif;
PROC HC_Q20131_1B_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_1B <> 7 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20131_2A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20131_2A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_2A <> 7 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20131_2B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if hc_q20131_2A = hc_q20131_2B then
    impute (hc_q20131_2B, 8);
    errmsg("DOUBLE");
endif;
PROC HC_Q20131_2B_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_2B <> 7 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20131_3A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20131_3A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_3A <> 7 then
    if $ <> "" then
        $ = "";
    endif;
```

```
endif;
PROC HC_Q20131_3B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if hc_q20131_3A = hc_q20131_3B then
    impute (hc_q20131_3B, 8);
    errmsg("DOUBLE");
endif;
PROC HC_Q20131_3B_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20131_3B <> 7 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20132
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20132_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20132 <> 5 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20133_A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20133_A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20133_A <> 6 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20133_B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if hc_q20133_A = hc_q20133_B then
    impute (hc_q20133_B, 8);
    errmsg("DOUBLE");
endif;
PROC HC_Q20133_B_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20141
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20142
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20143
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20144
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20145
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20146
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20145 in 1 then
    if $ in 1:2 then
        putdeck (hd_20146, $);
    else
        impute($, getdeck(hd_20146));
    endif;
    if $ in default then
        impute($, notappl);
        impute(HC_Q20145, 2);
    endif;
else
    if $ <> notappl then
        impute ($, notappl);
```

```
    endif;
endif;
PROC HC_Q20146_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);

if HC_Q20146 in 1 then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC HC_Q20211
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20212
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20213
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20214
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20215
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20215_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20216
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20217
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20218
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20219
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20220
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20221
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20222
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20223
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20224
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20225
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20231
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);

if ! $ in 1:2 then
    impute($, 2);
endif;
PROC HC_Q20232
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q20502_1
i = 0;
$ = Q20502(inc(i));
PROC HC_Q20502_2
$ = Q20502(inc(i));
PROC HC_Q20502_3
$ = Q20502(inc(i));
PROC HC_Q20502_4
$ = Q20502(inc(i));
PROC HC_Q20502_5
$ = Q20502(inc(i));
PROC HC_Q20502_6
$ = Q20502(inc(i));
PROC HC_Q20502_7
$ = Q20502(inc(i));
PROC HC_Q20502_8
$ = Q20502(inc(i));
PROC HC_Q20502_9
$ = Q20502(inc(i));
PROC HC_Q20502_10
$ = Q20502(inc(i));
```

```
PROC HC_Q20502_11
$ = Q20502(inc(i));
PROC HC_Q20502_12
$ = Q20502(inc(i));
PROC HC_Q20502_13
$ = Q20502(inc(i));
PROC HC_Q20502_14
$ = Q20502(inc(i));
PROC HC_Q20502_15
$ = Q20502(inc(i));
PROC HC_Q20502_16
$ = Q20502(inc(i));
PROC HC_Q20502_17
$ = Q20502(inc(i));
PROC HC_Q20502_18
$ = Q20502(inc(i));
PROC HC_Q20502_19
$ = Q20502(inc(i));
PROC HC_Q20502_20
$ = Q20502(inc(i));
PROC HC_Q20502_21
$ = Q20502(inc(i));
PROC HC_Q20502_22
$ = Q20502(inc(i));
PROC HC_Q20502_23
$ = Q20502(inc(i));
PROC HC_Q20502_24
$ = Q20502(inc(i));
PROC HC_Q20502_25
$ = Q20502(inc(i));
PROC HC_Q20502_26
$ = Q20502(inc(i));
PROC HC_Q20502_27
$ = Q20502(inc(i));
PROC HC_Q20502_28
$ = Q20502(inc(i));
PROC HC_Q20502_29
$ = Q20502(inc(i));
PROC HC_Q20502_30
$ = Q20502(inc(i));
PROC HC_Q20502_31
$ = Q20502(inc(i));
PROC HC_Q20502_32
$ = Q20502(inc(i));
PROC HC_Q20502_33
$ = Q20502(inc(i));
PROC HC_Q20502_34
$ = Q20502(inc(i));
PROC HC_Q20502_35
$ = Q20502(inc(i));
PROC HC_Q20502_36
$ = Q20502(inc(i));
PROC HC_Q20502_37
$ = Q20502(inc(i));
PROC HC_Q20502_38
$ = Q20502(inc(i));
PROC HC_Q20502_39
$ = Q20502(inc(i));
PROC HC_Q20502_40
$ = Q20502(inc(i));
PROC HC_Q20502_41
$ = Q20502(inc(i));
if $ in notappl then
    impute($, 2);
endif;
PROC HC_Q20602_1
i = 0;
$ = Q20602(inc(i));
PROC HC_Q20602_2
$ = Q20602(inc(i));
PROC HC_Q20602_3
```

```
$ = Q20602(inc(i));
PROC HC_Q20602_4
$ = Q20602(inc(i));
PROC HC_Q20602_5
$ = Q20602(inc(i));
PROC HC_Q20602_6
$ = Q20602(inc(i));
PROC HC_Q20602_7
$ = Q20602(inc(i));
PROC HC_Q20602_8
$ = Q20602(inc(i));
PROC HC_Q20602_9
$ = Q20602(inc(i));
PROC HC_Q20602_10
$ = Q20602(inc(i));
PROC HC_Q20602_11
$ = Q20602(inc(i));
PROC HC_Q40201_1
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_2
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_3
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_4
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_5
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_6
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_7
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_8
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_9
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40201_10
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40202
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40203
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40204
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40205
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40206
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40207
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_1
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_2
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_3
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_4
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_5
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_6
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_7
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_8
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_9
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40301_10
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40302
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
```

```
PROC HC_Q40303A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303C
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303D
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303E
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303F
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303G
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303G_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303H
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303I
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303J
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303K
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303K_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303L
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303M
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40603N
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40303N_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304C
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304D
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304E
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304F
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304G
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304G_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304H
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304I
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304J
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304K
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304L
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304L_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304M
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304N
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40304_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40305
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40306
```

```
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40307
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_1
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_2
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_3
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_4
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_5
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_6
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_7
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_8
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_9
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40401_10
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40402
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40403
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_1A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_1B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_2A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_2B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_3A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_3A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_3B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_4A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_4B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_5A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_5B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_6A
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_6A_O
$ = getvaluealpha(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40404_6B
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_1
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_2
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_3
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_4
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_5
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_6
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_7
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_8
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
```

```
PROC HC_Q40501_9
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40501_10
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC HC_Q40502
$ = getvaluenumeric(getsymbol($)[4:length(getsymbol($))]);
PROC PERSON_LEVEL_EDT
postproc
    // populate some HH variables that are dependent on the person record
    C_HH_EQUIV_OECD = sum(C_PSN_EQUIV_OECD);
    C_HH_EQUIV_FAO = sum(C_PSN_EQUIV_FAO);
    C_HH_EQUIV_WB = sum(C_PSN_EQUIV_WB);
    M_HH_FIRST_NAME = PSN_Q10102B(1);
    M_HH_LAST_NAME = PSN_Q10102A(1);
    M_HH_SEX = PSN_Q10103(1);
    M_HH_AGE = PSN_Q10104(1);
    M_HH_CITIZENSHIP = PSN_Q10108(1);
    M_HH_MARITAL_STATUS = PSN_Q10109(1);
    M_HH_MAIN_ACTIVITY = PSN_Q10201(1);
    M_HH_MAIN_OCCUPATION = PSN_Q10202_C(1);
    M_HH_MAIN_INDUSTRY = PSN_Q10203_C(1);
    M_HH_SECOND_ACTIVITY = PSN_Q10206(1);
    M_HH_SECOND_OCCUPATION = PSN_Q10207_C(1);
    M_HH_SECOND_INDUSTRY = PSN_Q10208_C(1);
    M_HH_ATTEND_EDU = PSN_Q10301(1);
    M_HH_REASON_NOT_ATTEND_EDU = PSN_Q10302(1);

    C_HH_TOT_STUDENTS  = count(PERSON_LEVEL_EDT where PSN_Q10303 in 1);     {Calculated -
    Total Number of Students}

    // Calculated - Total Number of Persons Employed
    C_HH_TOT_PER_EMPLOYED = count(PERSON_LEVEL_EDT where C_PSN_LFS1 in 1:8);
    // Calculated - Total Number of Persons Unemployed
    C_HH_TOT_PER_UNEMPLPOYED = count(PERSON_LEVEL_EDT where C_PSN_LFS1 in 9);
    // Calculated - Total Number of Persons Not in the Labour Force
    C_HH_TOT_PER_NOT_IN_LF = count(PERSON_LEVEL_EDT where C_PSN_LFS1 in 10:14);
PROC C_PSN_EQUIV_OECD
if curocc() in 1 then
    $ = 1.00;
else
    if age(curocc()) >= 15 then
        $ = 0.50;
    else
        $ = 0.30;
    endif;
endif;
PROC C_PSN_EQUIV_FAO
if sex(curocc()) in 1 then
    recode age(curocc()) => $;
        0               => 0.27;
        1,2,3           => 0.45;
        4,5,6           => 0.61;
        7,8,9           => 0.73;
        10,11,12        => 0.86;
        13,14,15        => 0.96;
        16,17,18,19     => 1.02;
        20-95           => 1.00;
    endrecode;
endif;
if sex(curocc()) in 2 then
    recode age(curocc()) => $;
        0               => 0.27;
        1,2,3           => 0.45;
        4,5,6           => 0.61;
        7,8,9           => 0.73;
        10,11,12        => 0.78;
        13,14,15        => 0.83;
        16,17,18,19     => 0.77;
        20-95           => 0.73;
    endrecode;
endif;
```

```
PROC C_PSN_EQUIV_WB
if age(curocc()) >= 15 then
    $ = 1.00;
else
    $ = 0.50;
endif;
PROC PSN_Q10101
$ = curocc();
PROC PSN_Q10102A
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10102B
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10103
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10104
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10105
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

if rcdage in 0:15 then
    if $ in 6 then
        impute ($, 9);
    endif;
endif;
PROC PSN_Q10105_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10106A
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10106B
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10106C
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if $ in notappl then
    impute($, MODULE_START_DATE_YYYY - psn_q10104);
endif;
PROC PSN_Q10107
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10107_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10108
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10108_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10109
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10110
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if $ in notappl then
    impute($, 1);
endif;
PROC PSN_Q10201
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

// manually change the activity type to student, wrongly coded
if ID07 in 339 and psn_q10101 = 6 then
    impute($, 9);
endif;
recode  ID07    :   PSN_Q10101  =>  PSN_Q10201  ;
    138 :   1   =>  11  ;
    150 :   3   =>  11  ;
    155 :   3   =>  11  ;
    158 :   1   =>  11  ;
    159 :   1   =>  11  ;
    151 :   4   =>  11  ;
    155 :   4   =>  11  ;
    154 :   4   =>  11  ;
    151 :   3   =>  11  ;
    155 :   1   =>  11  ;
    143 :   1   =>  11  ;
endrecode;
recode  ID07    :   PSN_Q10101  =>  PSN_Q10201  ;
```

```
    150 :   1   =>  3   ;
    150 :   2   =>  3   ;
    151 :   2   =>  3   ;
    155 :   2   =>  3   ;
    155 :   5   =>  3   ;
    206 :   2   =>  3   ;
    234 :   2   =>  3   ;
    234 :   3   =>  3   ;
    234 :   4   =>  3   ;
    234 :   5   =>  3   ;
    237 :   1   =>  3   ;
    237 :   2   =>  3   ;
    352 :   1   =>  3   ;
    352 :   2   =>  3   ;
    336 :   1   =>  8   ;
    340 :   1   =>  8   ;
    150 :   4   =>  11  ;
    105 :   1   =>  5   ;
    124 :   1   =>  11  ;
    217 :   1   =>  3   ;
endrecode;


if rcdAge < 15 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;

PROC PSN_Q10202
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10201(curocc()) in 9:14 or rcdAge < 15 then
    $ = "";
endif;
PROC PSN_Q10202_C
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

recode  ID07    :   PSN_Q10101  =>  PSN_Q10202_C    ;
    116 :   2   =>  111 ;
    118 :   1   =>  111 ;
    121 :   2   =>  111 ;
    127 :   3   =>  111 ;
    135 :   2   =>  111 ;
    150 :   2   =>  111 ;
    155 :   2   =>  111 ;
    217 :   1   =>  111 ;
    249 :   2   =>  111 ;
    253 :   1   =>  111 ;
    255 :   1   =>  111 ;
    319 :   1   =>  111 ;
    320 :   1   =>  111 ;
    322 :   1   =>  111 ;
    324 :   1   =>  111 ;
    117 :   1   =>  112 ;
    127 :   2   =>  112 ;
    224 :   2   =>  112 ;
    322 :   3   =>  112 ;
    332 :   2   =>  112 ;
    354 :   1   =>  112 ;
    105 :   4   =>  121 ;
    342 :   1   =>  122 ;
    103 :   2   =>  132 ;
    347 :   2   =>  132 ;
    119 :   2   =>  134 ;
    127 :   5   =>  134 ;
    346 :   7   =>  141 ;
    141 :   1   =>  214 ;
    148 :   2   =>  214 ;
    148 :   4   =>  214 ;
    154 :   1   =>  222 ;
    241 :   3   =>  222 ;
```

```
341 :   1   =>  222 ;
116 :   1   =>  226 ;
125 :   2   =>  226 ;
233 :   2   =>  231 ;
109 :   1   =>  233 ;
109 :   2   =>  233 ;
134 :   3   =>  233 ;
201 :   3   =>  233 ;
206 :   2   =>  233 ;
210 :   2   =>  233 ;
233 :   1   =>  233 ;
236 :   1   =>  233 ;
236 :   3   =>  233 ;
319 :   2   =>  233 ;
324 :   2   =>  233 ;
329 :   1   =>  233 ;
113 :   1   =>  234 ;
127 :   1   =>  234 ;
136 :   1   =>  234 ;
139 :   1   =>  234 ;
208 :   2   =>  234 ;
209 :   2   =>  234 ;
241 :   1   =>  234 ;
253 :   2   =>  234 ;
307 :   1   =>  234 ;
352 :   1   =>  234 ;
160 :   1   =>  235 ;
160 :   2   =>  235 ;
239 :   2   =>  241 ;
224 :   1   =>  242 ;
343 :   6   =>  252 ;
219 :   2   =>  261 ;
151 :   2   =>  262 ;
340 :   1   =>  263 ;
223 :   1   =>  315 ;
125 :   1   =>  322 ;
126 :   2   =>  322 ;
150 :   1   =>  322 ;
219 :   3   =>  322 ;
320 :   5   =>  322 ;
323 :   2   =>  322 ;
343 :   3   =>  322 ;
349 :   2   =>  322 ;
350 :   2   =>  322 ;
154 :   2   =>  335 ;
158 :   2   =>  335 ;
203 :   1   =>  335 ;
217 :   3   =>  335 ;
221 :   1   =>  335 ;
224 :   3   =>  335 ;
239 :   6   =>  335 ;
250 :   1   =>  335 ;
327 :   5   =>  335 ;
332 :   1   =>  335 ;
339 :   2   =>  335 ;
347 :   1   =>  335 ;
147 :   2   =>  341 ;
328 :   2   =>  341 ;
354 :   2   =>  341 ;
134 :   2   =>  351 ;
102 :   1   =>  412 ;
221 :   2   =>  412 ;
134 :   1   =>  422 ;
322 :   4   =>  422 ;
356 :   1   =>  422 ;
318 :   2   =>  512 ;
340 :   5   =>  512 ;
343 :   9   =>  512 ;
121 :   1   =>  522 ;
124 :   2   =>  522 ;
142 :   1   =>  522 ;
```

```
151 :   1   =>  522 ;
209 :   1   =>  522 ;
213 :   1   =>  522 ;
219 :   4   =>  522 ;
234 :   1   =>  522 ;
239 :   3   =>  522 ;
256 :   1   =>  522 ;
346 :   6   =>  522 ;
135 :   1   =>  531 ;
141 :   2   =>  531 ;
327 :   6   =>  531 ;
339 :   5   =>  531 ;
108 :   1   =>  541 ;
108 :   4   =>  541 ;
118 :   5   =>  541 ;
121 :   4   =>  541 ;
132 :   2   =>  541 ;
210 :   1   =>  541 ;
240 :   1   =>  541 ;
249 :   1   =>  541 ;
306 :   2   =>  541 ;
310 :   1   =>  541 ;
124 :   4   =>  711 ;
126 :   1   =>  711 ;
159 :   2   =>  711 ;
204 :   2   =>  711 ;
226 :   1   =>  711 ;
237 :   1   =>  711 ;
239 :   1   =>  711 ;
241 :   2   =>  711 ;
301 :   1   =>  711 ;
304 :   1   =>  711 ;
327 :   1   =>  711 ;
327 :   4   =>  711 ;
349 :   1   =>  711 ;
351 :   1   =>  711 ;
353 :   1   =>  711 ;
124 :   3   =>  712 ;
258 :   1   =>  712 ;
121 :   5   =>  723 ;
160 :   3   =>  723 ;
227 :   3   =>  723 ;
217 :   2   =>  731 ;
217 :   4   =>  731 ;
218 :   1   =>  731 ;
218 :   2   =>  731 ;
218 :   3   =>  731 ;
218 :   4   =>  731 ;
219 :   1   =>  731 ;
220 :   1   =>  731 ;
223 :   2   =>  731 ;
226 :   2   =>  731 ;
227 :   2   =>  731 ;
235 :   1   =>  731 ;
246 :   1   =>  731 ;
250 :   2   =>  731 ;
252 :   2   =>  731 ;
258 :   3   =>  731 ;
301 :   2   =>  731 ;
304 :   2   =>  731 ;
305 :   1   =>  731 ;
306 :   4   =>  731 ;
309 :   2   =>  731 ;
309 :   4   =>  731 ;
320 :   2   =>  731 ;
322 :   2   =>  731 ;
326 :   1   =>  731 ;
326 :   2   =>  731 ;
327 :   3   =>  731 ;
332 :   3   =>  731 ;
343 :   2   =>  731 ;
```

```
343 :   7   =>  731 ;
343 :   8   =>  731 ;
353 :   2   =>  731 ;
155 :   5   =>  741 ;
203 :   2   =>  741 ;
204 :   1   =>  741 ;
249 :   3   =>  741 ;
253 :   3   =>  741 ;
346 :   1   =>  741 ;
119 :   3   =>  833 ;
139 :   2   =>  833 ;
140 :   1   =>  911 ;
206 :   1   =>  911 ;
223 :   3   =>  911 ;
258 :   2   =>  911 ;
318 :   1   =>  911 ;
326 :   3   =>  911 ;
336 :   1   =>  911 ;
338 :   2   =>  911 ;
340 :   2   =>  911 ;
338 :   1   =>  921 ;
346 :   2   =>  921 ;
150 :   4   =>  961 ;
207 :   1   =>  961 ;
208 :   1   =>  961 ;
208 :   4   =>  961 ;
211 :   1   =>  961 ;
213 :   3   =>  961 ;
227 :   1   =>  961 ;
232 :   1   =>  961 ;
234 :   2   =>  961 ;
234 :   3   =>  961 ;
234 :   4   =>  961 ;
236 :   2   =>  961 ;
237 :   2   =>  961 ;
238 :   1   =>  961 ;
238 :   4   =>  961 ;
240 :   3   =>  961 ;
254 :   3   =>  961 ;
319 :   3   =>  961 ;
323 :   1   =>  961 ;
348 :   1   =>  961 ;
102 :   4   =>  962 ;
103 :   4   =>  962 ;
106 :   4   =>  962 ;
108 :   7   =>  962 ;
108 :   8   =>  962 ;
109 :   8   =>  962 ;
119 :   4   =>  962 ;
132 :   3   =>  962 ;
135 :   3   =>  962 ;
138 :   2   =>  962 ;
211 :   2   =>  962 ;
218 :   5   =>  962 ;
218 :   6   =>  962 ;
222 :   1   =>  962 ;
234 :   5   =>  962 ;
235 :   2   =>  962 ;
238 :   2   =>  962 ;
238 :   3   =>  962 ;
252 :   1   =>  962 ;
254 :   1   =>  962 ;
302 :   1   =>  962 ;
306 :   3   =>  962 ;
306 :   5   =>  962 ;
308 :   1   =>  962 ;
309 :   1   =>  962 ;
309 :   3   =>  962 ;
309 :   5   =>  962 ;
311 :   1   =>  962 ;
311 :   3   =>  962 ;
```

```
    320 :   3   =>  962 ;
    320 :   4   =>  962 ;
    328 :   1   =>  962 ;
    333 :   1   =>  962 ;
    337 :   1   =>  962 ;
    339 :   1   =>  962 ;
    340 :   4   =>  962 ;
    343 :   1   =>  962 ;
    350 :   1   =>  962 ;
    352 :   2   =>  962 ;
endrecode;

if ID07 in 104 and psn_q10101 = 3 then
    impute($, 341);
endif;

if psn_q10201 in 9:14 or rcdAge < 15 then
    $ = notappl;
endif;
PROC PSN_Q10203
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10201 in 9:14 or rcdAge < 15 then
    $ = "";
endif;

PROC PSN_Q10203_C
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

recode  ID07    :   PSN_Q10101  =>  PSN_Q10203_C    ;
    116 :   2   =>  841 ;
    118 :   1   =>  841 ;
    121 :   2   =>  841 ;
    127 :   3   =>  841 ;
    135 :   2   =>  841 ;
    143 :   2   =>  841 ;
    150 :   2   =>  841 ;
    155 :   2   =>  841 ;
    217 :   1   =>  841 ;
    249 :   2   =>  841 ;
    253 :   1   =>  841 ;
    255 :   1   =>  841 ;
    319 :   1   =>  841 ;
    320 :   1   =>  841 ;
    322 :   1   =>  841 ;
    324 :   1   =>  841 ;
    117 :   1   =>  841 ;
    127 :   2   =>  841 ;
    224 :   2   =>  841 ;
    322 :   3   =>  841 ;
    332 :   2   =>  841 ;
    354 :   1   =>  841 ;
    105 :   4   =>  841 ;
    342 :   1   =>  471 ;
    103 :   2   =>  471 ;
    347 :   2   =>  471 ;
    119 :   2   =>  841 ;
    127 :   5   =>  841 ;
    346 :   7   =>  551 ;
    141 :   1   =>  351 ;
    148 :   2   =>  351 ;
    148 :   4   =>  351 ;
    154 :   1   =>  861 ;
    241 :   3   =>  861 ;
    341 :   1   =>  861 ;
    116 :   1   =>  861 ;
    125 :   2   =>  861 ;
    233 :   2   =>  351 ;
    109 :   1   =>  852 ;
    109 :   2   =>  852 ;
    134 :   3   =>  852 ;
    201 :   3   =>  852 ;
```

```
206 :   2   =>  852 ;
210 :   2   =>  852 ;
233 :   1   =>  852 ;
236 :   1   =>  852 ;
236 :   3   =>  852 ;
319 :   2   =>  852 ;
324 :   2   =>  852 ;
329 :   1   =>  852 ;
113 :   1   =>  851 ;
127 :   1   =>  851 ;
136 :   1   =>  851 ;
139 :   1   =>  851 ;
208 :   2   =>  851 ;
209 :   2   =>  851 ;
241 :   1   =>  851 ;
253 :   2   =>  851 ;
307 :   1   =>  851 ;
352 :   1   =>  851 ;
160 :   1   =>  852 ;
160 :   2   =>  852 ;
239 :   2   =>  841 ;
224 :   1   =>  841 ;
343 :   6   =>  841 ;
219 :   2   =>  841 ;
151 :   2   =>  852 ;
340 :   1   =>  949 ;
223 :   1   =>  841 ;
125 :   1   =>  861 ;
126 :   2   =>  861 ;
150 :   1   =>  861 ;
219 :   3   =>  861 ;
320 :   5   =>  861 ;
323 :   2   =>  861 ;
343 :   3   =>  861 ;
349 :   2   =>  861 ;
350 :   2   =>  861 ;
154 :   2   =>  841 ;
158 :   2   =>  841 ;
203 :   1   =>  841 ;
217 :   3   =>  841 ;
221 :   1   =>  841 ;
224 :   3   =>  841 ;
239 :   6   =>  841 ;
250 :   1   =>  841 ;
327 :   5   =>  841 ;
332 :   1   =>  841 ;
339 :   2   =>  841 ;
347 :   1   =>  841 ;
147 :   2   =>  841 ;
328 :   2   =>  841 ;
354 :   2   =>  841 ;
134 :   2   =>  841 ;
102 :   1   =>  851 ;
221 :   2   =>  841 ;
134 :   1   =>  841 ;
322 :   4   =>  841 ;
356 :   1   =>  841 ;
318 :   2   =>  841 ;
340 :   5   =>  841 ;
343 :   9   =>  841 ;
121 :   1   =>  471 ;
124 :   2   =>  471 ;
142 :   1   =>  471 ;
151 :   1   =>  471 ;
209 :   1   =>  471 ;
213 :   1   =>  471 ;
219 :   4   =>  471 ;
234 :   1   =>  471 ;
239 :   3   =>  471 ;
256 :   1   =>  471 ;
346 :   6   =>  471 ;
```

```
135 :   1   =>  851 ;
141 :   2   =>  851 ;
327 :   6   =>  851 ;
339 :   5   =>  851 ;
108 :   1   =>  842 ;
108 :   4   =>  842 ;
118 :   5   =>  842 ;
121 :   4   =>  842 ;
132 :   2   =>  842 ;
210 :   1   =>  842 ;
240 :   1   =>  842 ;
249 :   1   =>  842 ;
306 :   2   =>  842 ;
310 :   1   =>  842 ;
124 :   4   =>  410 ;
126 :   1   =>  841 ;
159 :   2   =>  410 ;
204 :   2   =>  410 ;
226 :   1   =>  410 ;
237 :   1   =>  410 ;
239 :   1   =>  410 ;
241 :   2   =>  410 ;
301 :   1   =>  841 ;
304 :   1   =>  410 ;
327 :   1   =>  410 ;
327 :   4   =>  410 ;
349 :   1   =>  410 ;
351 :   1   =>  410 ;
353 :   1   =>  410 ;
124 :   3   =>  841 ;
258 :   1   =>  841 ;
121 :   5   =>  841 ;
160 :   3   =>  351 ;
227 :   3   =>  841 ;
217 :   2   =>  131 ;
217 :   4   =>  131 ;
218 :   1   =>  131 ;
218 :   2   =>  131 ;
218 :   3   =>  131 ;
218 :   4   =>  131 ;
219 :   1   =>  131 ;
220 :   1   =>  131 ;
223 :   2   =>  131 ;
226 :   2   =>  131 ;
227 :   2   =>  131 ;
235 :   1   =>  131 ;
246 :   1   =>  131 ;
250 :   2   =>  131 ;
252 :   2   =>  131 ;
258 :   3   =>  131 ;
301 :   2   =>  131 ;
304 :   2   =>  131 ;
305 :   1   =>  131 ;
306 :   4   =>  131 ;
309 :   2   =>  131 ;
309 :   4   =>  131 ;
320 :   2   =>  131 ;
322 :   2   =>  131 ;
326 :   1   =>  131 ;
326 :   2   =>  131 ;
327 :   3   =>  131 ;
332 :   3   =>  131 ;
343 :   2   =>  131 ;
343 :   7   =>  131 ;
343 :   8   =>  131 ;
353 :   2   =>  131 ;
155 :   5   =>  351 ;
203 :   2   =>  351 ;
204 :   1   =>  351 ;
249 :   3   =>  841 ;
253 :   3   =>  351 ;
```

```
346 :   1   =>  351 ;
119 :   3   =>  841 ;
139 :   2   =>  841 ;
140 :   1   =>  970 ;
206 :   1   =>  551 ;
223 :   3   =>  551 ;
258 :   2   =>  551 ;
318 :   1   =>  851 ;
326 :   3   =>  841 ;
336 :   1   =>  841 ;
338 :   2   =>  970 ;
340 :   2   =>  949 ;
338 :   1   =>  841 ;
346 :   2   =>  841 ;
150 :   4   =>  841 ;
207 :   1   =>  841 ;
208 :   1   =>  841 ;
208 :   4   =>  841 ;
211 :   1   =>  841 ;
213 :   3   =>  841 ;
227 :   1   =>  841 ;
232 :   1   =>  841 ;
234 :   2   =>  841 ;
234 :   3   =>  841 ;
234 :   4   =>  841 ;
236 :   2   =>  841 ;
237 :   2   =>  841 ;
238 :   1   =>  841 ;
238 :   4   =>  841 ;
240 :   3   =>  841 ;
254 :   3   =>  841 ;
319 :   3   =>  841 ;
323 :   1   =>  841 ;
348 :   1   =>  841 ;
102 :   4   =>  841 ;
103 :   4   =>  841 ;
106 :   4   =>  841 ;
108 :   7   =>  841 ;
108 :   8   =>  841 ;
109 :   8   =>  841 ;
119 :   4   =>  841 ;
132 :   3   =>  841 ;
135 :   3   =>  841 ;
138 :   2   =>  841 ;
211 :   2   =>  841 ;
218 :   5   =>  841 ;
218 :   6   =>  841 ;
222 :   1   =>  841 ;
234 :   5   =>  841 ;
235 :   2   =>  841 ;
238 :   2   =>  841 ;
238 :   3   =>  841 ;
252 :   1   =>  841 ;
254 :   1   =>  841 ;
302 :   1   =>  841 ;
306 :   3   =>  841 ;
306 :   5   =>  841 ;
308 :   1   =>  841 ;
309 :   1   =>  841 ;
309 :   3   =>  841 ;
309 :   5   =>  841 ;
311 :   1   =>  841 ;
311 :   3   =>  841 ;
320 :   3   =>  841 ;
320 :   4   =>  841 ;
328 :   1   =>  841 ;
333 :   1   =>  841 ;
337 :   1   =>  841 ;
339 :   1   =>  841 ;
340 :   4   =>  841 ;
343 :   1   =>  841 ;
```

```
    350 :   1   =>  841 ;
    352 :   2   =>  841 ;
endrecode;
if ID07 in 104 and psn_q10101 = 3 then
    impute($, 949);
endif;
if psn_q10201 in 9:14 or rcdAge < 15 then
    $ = notappl;
endif;
PROC PSN_Q10204
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10201 in 9:14 or rcdAge < 15 then
    $ = notappl;
endif;
PROC PSN_Q10205
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10201 in 9:14 then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
if psn_q10201 in 1:8 then
    if psn_q10204 >= 30 then
        if $ <> notappl then
            impute($, notappl);
        endif;
    else
        if ! $ in 1:2 then
            impute($, 9);
        endif;
    endif;
endif;

if PSN_Q10201(curocc()) in 9:14 then
    PSN_Q10202(curocc())        = "";
    PSN_Q10202_C(curocc())  = notappl;
    PSN_Q10203(curocc())        = "";
    PSN_Q10203_C(curocc())  = notappl;
    PSN_Q10204(curocc())        = notappl;
    PSN_Q10205(curocc())        = notappl;
endif;
PROC PSN_Q10206
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

recode  ID07    :   PSN_Q10101  =>  PSN_Q10206  ;
    356 :   1   =>  11  ;
    234 :   6   =>  11  ;
    239 :   7   =>  11  ;
    252 :   4   =>  11  ;
    223 :   3   =>  11  ;
    348 :   1   =>  11  ;
    204 :   1   =>  11  ;
    253 :   3   =>  11  ;
    213 :   3   =>  11  ;
    238 :   3   =>  11  ;
    252 :   1   =>  11  ;
    341 :   1   =>  11  ;
    350 :   2   =>  11  ;
    208 :   4   =>  11  ;
    227 :   1   =>  11  ;
    237 :   2   =>  11  ;
    142 :   1   =>  11  ;
    234 :   1   =>  11  ;
    139 :   1   =>  11  ;
    208 :   2   =>  11  ;
    250 :   2   =>  11  ;
    258 :   3   =>  11  ;
    219 :   5   =>  11  ;
    220 :   2   =>  11  ;
    236 :   5   =>  11  ;
    241 :   4   =>  11  ;
```

```
103 :   3   =>  11  ;
121 :   3   =>  11  ;
147 :   1   =>  11  ;
151 :   3   =>  11  ;
127 :   5   =>  11  ;
104 :   4   =>  11  ;
221 :   1   =>  11  ;
114 :   1   =>  11  ;
105 :   1   =>  11  ;
160 :   1   =>  11  ;
219 :   2   =>  11  ;
126 :   2   =>  11  ;
102 :   1   =>  11  ;
109 :   1   =>  11  ;
217 :   1   =>  11  ;
136 :   1   =>  11  ;
233 :   1   =>  11  ;
236 :   1   =>  11  ;
141 :   2   =>  11  ;
232 :   1   =>  11  ;
217 :   2   =>  11  ;
219 :   1   =>  11  ;
306 :   6   =>  11  ;
218 :   3   =>  11  ;
208 :   5   =>  11  ;
240 :   5   =>  11  ;
208 :   3   =>  11  ;
239 :   6   =>  11  ;
211 :   1   =>  11  ;
258 :   2   =>  11  ;
235 :   2   =>  11  ;
249 :   2   =>  11  ;
238 :   1   =>  11  ;
254 :   3   =>  11  ;
234 :   4   =>  11  ;
132 :   1   =>  11  ;
209 :   2   =>  11  ;
253 :   2   =>  11  ;
207 :   1   =>  11  ;
223 :   2   =>  11  ;
226 :   2   =>  11  ;
227 :   2   =>  11  ;
326 :   4   =>  11  ;
327 :   4   =>  11  ;
320 :   3   =>  11  ;
322 :   4   =>  11  ;
326 :   2   =>  11  ;
154 :   3   =>  11  ;
221 :   3   =>  11  ;
221 :   5   =>  11  ;
232 :   2   =>  11  ;
236 :   6   =>  11  ;
236 :   4   =>  11  ;
111 :   2   =>  11  ;
118 :   3   =>  11  ;
119 :   1   =>  11  ;
124 :   5   =>  11  ;
126 :   3   =>  11  ;
148 :   5   =>  11  ;
155 :   1   =>  11  ;
327 :   2   =>  11  ;
140 :   1   =>  11  ;
127 :   2   =>  11  ;
233 :   2   =>  11  ;
306 :   4   =>  11  ;
104 :   3   =>  11  ;
241 :   3   =>  11  ;
125 :   1   =>  11  ;
217 :   3   =>  11  ;
108 :   1   =>  11  ;
113 :   1   =>  11  ;
```

```
221 :   2   =>  11  ;
121 :   1   =>  11  ;
219 :   4   =>  11  ;
217 :   4   =>  11  ;
220 :   1   =>  11  ;
252 :   2   =>  11  ;
218 :   6   =>  11  ;
218 :   5   =>  11  ;
218 :   2   =>  11  ;
218 :   1   =>  11  ;
354 :   2   =>  11  ;
308 :   2   =>  11  ;
332 :   2   =>  11  ;
308 :   1   =>  11  ;
226 :   4   =>  5   ;
226 :   1   =>  5   ;
255 :   1   =>  5   ;
105 :   2   =>  5   ;
109 :   8   =>  5   ;
160 :   3   =>  5   ;
138 :   2   =>  5   ;
143 :   2   =>  5   ;
234 :   7   =>  5   ;
139 :   2   =>  5   ;
148 :   2   =>  5   ;
148 :   4   =>  5   ;
135 :   3   =>  5   ;
132 :   3   =>  5   ;
254 :   1   =>  5   ;
234 :   5   =>  5   ;
227 :   3   =>  5   ;
158 :   2   =>  5   ;
147 :   2   =>  5   ;
132 :   2   =>  5   ;
150 :   2   =>  5   ;
154 :   5   =>  5   ;
160 :   2   =>  5   ;
141 :   1   =>  5   ;
154 :   2   =>  5   ;
352 :   2   =>  5   ;
150 :   4   =>  5   ;
239 :   1   =>  5   ;
103 :   1   =>  5   ;
138 :   1   =>  5   ;
102 :   2   =>  5   ;
104 :   2   =>  5   ;
113 :   2   =>  5   ;
103 :   4   =>  5   ;
151 :   2   =>  5   ;
127 :   3   =>  5   ;
116 :   1   =>  8   ;
119 :   2   =>  8   ;
118 :   2   =>  8   ;
127 :   4   =>  8   ;
124 :   1   =>  8   ;
320 :   2   =>  8   ;
154 :   4   =>  11  ;
339 :   6   =>  11  ;
349 :   1   =>  11  ;
353 :   1   =>  11  ;
338 :   2   =>  11  ;
346 :   1   =>  11  ;
346 :   2   =>  11  ;
320 :   5   =>  11  ;
346 :   7   =>  11  ;
339 :   5   =>  11  ;
353 :   2   =>  11  ;
339 :   2   =>  11  ;
328 :   1   =>  11  ;
328 :   4   =>  11  ;
327 :   1   =>  11  ;
```

```
    349 :   2   =>  11  ;
    135 :   4   =>  11  ;
    135 :   2   =>  11  ;
    135 :   1   =>  11  ;
    347 :   1   =>  11  ;
    343 :   9   =>  11  ;
    343 :   3   =>  11  ;
    332 :   3   =>  11  ;
    343 :   8   =>  11  ;
    328 :   2   =>  11  ;
    211 :   2   =>  11  ;
    209 :   1   =>  11  ;
    203 :   1   =>  11  ;
    203 :   2   =>  11  ;
    319 :   2   =>  11  ;
    329 :   1   =>  11  ;
    323 :   2   =>  11  ;
    306 :   1   =>  11  ;
    324 :   2   =>  11  ;
    352 :   1   =>  11  ;
    301 :   2   =>  11  ;
    143 :   1   =>  11  ;
    102 :   4   =>  5   ;
    106 :   3   =>  11  ;
    111 :   1   =>  11  ;
    309 :   3   =>  8   ;
    237 :   1   =>  8   ;
    159 :   2   =>  5   ;
    151 :   1   =>  11  ;
    351 :   1   =>  8   ;
    240 :   3   =>  8   ;
    240 :   4   =>  11  ;
    252 :   3   =>  11  ;
    239 :   3   =>  11  ;
    256 :   1   =>  11  ;
    234 :   3   =>  11  ;
    118 :   4   =>  11  ;
    219 :   6   =>  11  ;
    233 :   4   =>  11  ;
    241 :   5   =>  11  ;
    114 :   2   =>  11  ;
    151 :   4   =>  11  ;
    155 :   3   =>  11  ;
    158 :   1   =>  11  ;
    159 :   1   =>  11  ;
    106 :   1   =>  11  ;
    104 :   1   =>  11  ;
    219 :   3   =>  11  ;
    150 :   1   =>  11  ;
    121 :   4   =>  11  ;
    154 :   1   =>  11  ;
    127 :   1   =>  11  ;
    236 :   3   =>  11  ;
    240 :   2   =>  5   ;
    235 :   1   =>  5   ;
    246 :   1   =>  5   ;
    234 :   2   =>  5   ;
    150 :   3   =>  5   ;
    134 :   1   =>  5   ;
    343 :   2   =>  5   ;
    343 :   7   =>  5   ;
    239 :   2   =>  5   ;
    340 :   2   =>  8   ;
    340 :   3   =>  8   ;
    155 :   2   =>  3   ;
endrecode;
PROC PSN_Q10207
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> "" then
        $ = "";
```

```
    endif;
endif;
PROC PSN_Q10207_C
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> notappl then
        impute($, notappl);
    endif;
else
    recode  ID07    :   PSN_Q10101  =>  PSN_Q10207_C    ;
        342 :   1   =>  111 ;
        155 :   2   =>  111 ;
        340 :   1   =>  111 ;
        307 :   1   =>  234 ;
        318 :   1   =>  512 ;
        318 :   2   =>  512 ;
        224 :   2   =>  512 ;
        206 :   1   =>  512 ;
        206 :   2   =>  512 ;
        333 :   1   =>  514 ;
        155 :   4   =>  531 ;
        226 :   1   =>  631 ;
        226 :   4   =>  631 ;
        255 :   1   =>  631 ;
        311 :   3   =>  631 ;
        239 :   1   =>  631 ;
        302 :   1   =>  631 ;
        304 :   1   =>  631 ;
        309 :   1   =>  631 ;
        102 :   2   =>  631 ;
        103 :   1   =>  631 ;
        103 :   4   =>  631 ;
        104 :   2   =>  631 ;
        113 :   2   =>  631 ;
        127 :   3   =>  631 ;
        138 :   1   =>  631 ;
        151 :   2   =>  631 ;
        322 :   1   =>  631 ;
        322 :   2   =>  631 ;
        159 :   2   =>  631 ;
        105 :   2   =>  632 ;
        109 :   8   =>  632 ;
        138 :   2   =>  632 ;
        160 :   3   =>  632 ;
        102 :   4   =>  632 ;
        309 :   6   =>  632 ;
        327 :   5   =>  633 ;
        327 :   6   =>  633 ;
        106 :   4   =>  634 ;
        108 :   4   =>  634 ;
        108 :   7   =>  634 ;
        108 :   8   =>  634 ;
        103 :   2   =>  634 ;
        105 :   4   =>  634 ;
        109 :   2   =>  634 ;
        116 :   2   =>  634 ;
        118 :   1   =>  634 ;
        118 :   5   =>  634 ;
        121 :   2   =>  634 ;
        132 :   2   =>  634 ;
        132 :   3   =>  634 ;
        135 :   3   =>  634 ;
        139 :   2   =>  634 ;
        143 :   2   =>  634 ;
        147 :   2   =>  634 ;
        148 :   2   =>  634 ;
        148 :   4   =>  634 ;
        158 :   2   =>  634 ;
        227 :   3   =>  634 ;
        234 :   5   =>  634 ;
        234 :   7   =>  634 ;
```

```
254 :  1  =>  634 ;
309 :  5  =>  634 ;
311 :  1  =>  634 ;
323 :  1  =>  634 ;
117 :  1  =>  634 ;
119 :  3  =>  634 ;
119 :  4  =>  634 ;
124 :  2  =>  634 ;
124 :  3  =>  634 ;
124 :  4  =>  634 ;
125 :  2  =>  634 ;
126 :  1  =>  634 ;
141 :  1  =>  634 ;
150 :  2  =>  634 ;
150 :  4  =>  634 ;
154 :  2  =>  634 ;
154 :  5  =>  634 ;
155 :  5  =>  634 ;
160 :  2  =>  634 ;
208 :  1  =>  634 ;
301 :  1  =>  634 ;
320 :  1  =>  634 ;
320 :  4  =>  634 ;
322 :  3  =>  634 ;
326 :  1  =>  634 ;
327 :  3  =>  634 ;
332 :  1  =>  634 ;
338 :  1  =>  634 ;
339 :  1  =>  634 ;
343 :  1  =>  634 ;
343 :  6  =>  634 ;
346 :  6  =>  634 ;
347 :  2  =>  634 ;
352 :  2  =>  634 ;
354 :  1  =>  634 ;
121 :  5  =>  634 ;
237 :  1  =>  723 ;
108 :  3  =>  731 ;
116 :  1  =>  731 ;
118 :  2  =>  731 ;
119 :  2  =>  731 ;
124 :  1  =>  731 ;
127 :  4  =>  731 ;
320 :  2  =>  731 ;
108 :  2  =>  731 ;
108 :  5  =>  731 ;
108 :  6  =>  731 ;
305 :  1  =>  731 ;
234 :  2  =>  731 ;
235 :  1  =>  731 ;
240 :  2  =>  731 ;
246 :  1  =>  731 ;
134 :  1  =>  731 ;
150 :  3  =>  731 ;
326 :  3  =>  731 ;
343 :  2  =>  731 ;
343 :  7  =>  731 ;
239 :  2  =>  731 ;
310 :  1  =>  731 ;
210 :  2  =>  911 ;
340 :  2  =>  911 ;
340 :  3  =>  911 ;
340 :  4  =>  911 ;
350 :  1  =>  911 ;
304 :  2  =>  911 ;
309 :  4  =>  911 ;
324 :  1  =>  911 ;
336 :  1  =>  911 ;
340 :  5  =>  911 ;
309 :  2  =>  911 ;
241 :  1  =>  911 ;
```

```
        241 :   2   =>  911 ;
        249 :   3   =>  931 ;
        224 :   1   =>  931 ;
        319 :   1   =>  931 ;
        319 :   3   =>  931 ;
        238 :   2   =>  931 ;
        240 :   1   =>  931 ;
        249 :   1   =>  931 ;
        258 :   1   =>  931 ;
        201 :   3   =>  931 ;
        204 :   2   =>  931 ;
        236 :   2   =>  931 ;
        106 :   2   =>  931 ;
        222 :   1   =>  931 ;
        223 :   1   =>  931 ;
        253 :   1   =>  931 ;
        250 :   1   =>  931 ;
        309 :   3   =>  931 ;
        306 :   2   =>  961 ;
        306 :   3   =>  961 ;
        306 :   5   =>  961 ;
        224 :   3   =>  961 ;
        337 :   1   =>  961 ;
        240 :   3   =>  961 ;
        351 :   1   =>  961 ;
        114 :   3   =>  961 ;
    endrecode;
endif;

recode  ID07    :   PSN_Q10101  =>  PSN_Q10202_C    ;
    103 :   1   =>  731 ;
    309 :   6   =>  911 ;
    340 :   3   =>  911 ;
endrecode;
recode  ID07    :   PSN_Q10101  =>  PSN_Q10201  ;
    103 :   1   =>  8   ;
    103 :   3   =>  11  ;
    106 :   3   =>  11  ;
    108 :   3   =>  11  ;
    108 :   5   =>  11  ;
    108 :   6   =>  11  ;
    111 :   2   =>  11  ;
    114 :   2   =>  11  ;
    118 :   2   =>  11  ;
    118 :   3   =>  11  ;
    119 :   1   =>  11  ;
    121 :   3   =>  11  ;
    124 :   5   =>  11  ;
    126 :   3   =>  11  ;
    127 :   4   =>  11  ;
    148 :   5   =>  11  ;
    234 :   7   =>  11  ;
endrecode;

PROC PSN_Q10208
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> "" then
        $ = "";
    endif;
endif;
PROC PSN_Q10208_C
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> notappl then
        impute($, notappl);
    endif;
else
    recode  ID07    :   PSN_Q10101  =>  PSN_Q10208_C    ;
        342 :   1   =>  841 ;
        155 :   2   =>  841 ;
```

```
340 :   1   =>  841 ;
307 :   1   =>  851 ;
318 :   1   =>  652 ;
318 :   2   =>  652 ;
224 :   2   =>  970 ;
206 :   1   =>  652 ;
206 :   2   =>  652 ;
333 :   1   =>  960 ;
155 :   4   =>  970 ;
226 :   1   =>  11  ;
226 :   4   =>  11  ;
255 :   1   =>  11  ;
311 :   3   =>  11  ;
239 :   1   =>  11  ;
302 :   1   =>  11  ;
304 :   1   =>  11  ;
309 :   1   =>  11  ;
102 :   2   =>  11  ;
103 :   1   =>  11  ;
103 :   4   =>  11  ;
104 :   2   =>  11  ;
113 :   2   =>  11  ;
127 :   3   =>  11  ;
138 :   1   =>  11  ;
151 :   2   =>  11  ;
322 :   1   =>  11  ;
322 :   2   =>  11  ;
159 :   2   =>  11  ;
105 :   2   =>  14  ;
109 :   8   =>  14  ;
138 :   2   =>  14  ;
160 :   3   =>  14  ;
102 :   4   =>  14  ;
309 :   6   =>  14  ;
327 :   5   =>  15  ;
327 :   6   =>  15  ;
106 :   4   =>  31  ;
108 :   4   =>  31  ;
108 :   7   =>  31  ;
108 :   8   =>  31  ;
103 :   2   =>  31  ;
105 :   4   =>  31  ;
109 :   2   =>  31  ;
116 :   2   =>  31  ;
118 :   1   =>  31  ;
118 :   5   =>  31  ;
121 :   2   =>  31  ;
132 :   2   =>  31  ;
132 :   3   =>  31  ;
135 :   3   =>  31  ;
139 :   2   =>  31  ;
143 :   2   =>  31  ;
147 :   2   =>  31  ;
148 :   2   =>  31  ;
148 :   4   =>  31  ;
158 :   2   =>  31  ;
227 :   3   =>  31  ;
234 :   5   =>  31  ;
234 :   7   =>  31  ;
254 :   1   =>  31  ;
309 :   5   =>  31  ;
311 :   1   =>  31  ;
323 :   1   =>  31  ;
117 :   1   =>  31  ;
119 :   3   =>  31  ;
119 :   4   =>  31  ;
124 :   2   =>  31  ;
124 :   3   =>  31  ;
124 :   4   =>  31  ;
125 :   2   =>  31  ;
126 :   1   =>  31  ;
```

```
141 :   1   =>  31  ;
150 :   2   =>  31  ;
150 :   4   =>  31  ;
154 :   2   =>  31  ;
154 :   5   =>  31  ;
155 :   5   =>  31  ;
160 :   2   =>  31  ;
208 :   1   =>  31  ;
301 :   1   =>  31  ;
320 :   1   =>  31  ;
320 :   4   =>  31  ;
322 :   3   =>  31  ;
326 :   1   =>  31  ;
327 :   3   =>  31  ;
332 :   1   =>  31  ;
338 :   1   =>  31  ;
339 :   1   =>  31  ;
343 :   1   =>  31  ;
343 :   6   =>  31  ;
346 :   6   =>  31  ;
347 :   2   =>  31  ;
352 :   2   =>  31  ;
354 :   1   =>  31  ;
121 :   5   =>  31  ;
237 :   1   =>  452 ;
108 :   3   =>  162 ;
116 :   1   =>  162 ;
118 :   2   =>  162 ;
119 :   2   =>  162 ;
124 :   1   =>  162 ;
127 :   4   =>  162 ;
320 :   2   =>  162 ;
108 :   2   =>  162 ;
108 :   5   =>  162 ;
108 :   6   =>  162 ;
305 :   1   =>  162 ;
234 :   2   =>  162 ;
235 :   1   =>  162 ;
240 :   2   =>  162 ;
246 :   1   =>  162 ;
134 :   1   =>  162 ;
150 :   3   =>  162 ;
326 :   3   =>  162 ;
343 :   2   =>  162 ;
343 :   7   =>  162 ;
239 :   2   =>  162 ;
310 :   1   =>  162 ;
210 :   2   =>  812 ;
340 :   2   =>  949 ;
340 :   3   =>  949 ;
340 :   4   =>  949 ;
350 :   1   =>  949 ;
304 :   2   =>  949 ;
309 :   4   =>  949 ;
324 :   1   =>  949 ;
336 :   1   =>  949 ;
340 :   5   =>  949 ;
309 :   2   =>  949 ;
241 :   1   =>  949 ;
241 :   2   =>  949 ;
249 :   3   =>  410 ;
224 :   1   =>  410 ;
319 :   1   =>  410 ;
319 :   3   =>  410 ;
238 :   2   =>  410 ;
240 :   1   =>  410 ;
249 :   1   =>  410 ;
258 :   1   =>  410 ;
201 :   3   =>  410 ;
204 :   2   =>  410 ;
236 :   2   =>  410 ;
```

```
        106 :   2   =>  410 ;
        222 :   1   =>  410 ;
        223 :   1   =>  410 ;
        253 :   1   =>  410 ;
        250 :   1   =>  410 ;
        309 :   3   =>  410 ;
        306 :   2   =>  429 ;
        306 :   3   =>  429 ;
        306 :   5   =>  429 ;
        224 :   3   =>  381 ;
        337 :   1   =>  381 ;
        240 :   3   =>  381 ;
        351 :   1   =>  381 ;
        114 :   3   =>  841 ;
    endrecode;
endif;
PROC PSN_Q10209
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC PSN_Q10210
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if !(psn_q10206 in 1:8) then
    if $ <> notappl then
        impute($(curocc()), notappl);
    endif;
else
    if psn_q10209 >= 30 then
        if $ <> notappl then
            impute($, notappl);
        endif;
    endif;
endif;

if psn_q10209 in 0:29 then
    if $(curocc())in notappl then
        impute($, 2);
    endif;
endif;

PROC PSN_Q10211
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10212
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if not(psn_q10211 in 2) then
    if $ <> notappl then
        impute ($, notappl);
    endif;
endif;

PROC PSN_Q10212_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
if not(psn_q10211 in 2) then
    if $ <> "" then
        $ = "";
    endif;
endif;

PROC PSN_Q10213
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));

// PRIMARY
if psn_q10201 in 1:14 then
    c_psn_lfs1 = psn_q10201;
endif;
// let combine part and full time student
if psn_q10201 in 9 then
    c_psn_lfs1 = 10;
```

```
endif;

// lets clear the umemp so that we can properly reassign
if c_psn_lfs1 in 9 then
    c_psn_lfs1 = notappl;
endif;

// lets calculate the unemp rate
if c_psn_lfs1 in 10:14 then
    // looked for and able to work are put back into the LF:UNEMP
    if psn_q10211 in 1 then
        if psn_q10213 in 1 then
            c_psn_lfs1 = 9;
        endif;
    else
        // discouraged / weather reasons are put back into the LF:UNEMP
        if psn_q10212 in 5,6,8 and psn_q10213 in 1 then
            c_psn_lfs1 = 9;
        endif;
    endif;
endif;

//SECONDARY
if psn_q10206 in 1:14 then
    c_psn_lfs2 = psn_q10206;
endif;
// let combine part and full time student
if psn_q10206 in 9 then
    c_psn_lfs2 = 10;
endif;

// lets clear the umemp so that we can properly reassign
if c_psn_lfs2 in 9 then
    c_psn_lfs2 = notappl;
endif;

// lets calculate the unemp rate
{ NOT APPLICABLE FOR SECONDARY
if c_psn_lfs1 in 10:14 and c_psn_lfs2 in 10:14 then
    // looked for and able to work are put back into the LF:UNEMP
    if psn_q10211 in 1 then
        if psn_q10213 in 1 then
            c_psn_lfs2 = 9;
        endif;
    else
        // discouraged / weather reasons are put back into the LF:UNEMP
        if psn_q10212 in 5,6,8 and psn_q10213 in 1 then
            c_psn_lfs2 = 9;
        endif;
    endif;
endif;
}
PROC PSN_Q10301
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if PSN_Q10201(curocc()) in 9:14 then
    PSN_Q10202(curocc())        = "";
    PSN_Q10202_C(curocc())  = notappl;
    PSN_Q10203(curocc())        = "";
    PSN_Q10203_C(curocc())  = notappl;
    PSN_Q10204(curocc())        = notappl;
    PSN_Q10205(curocc())        = notappl;
endif;
PROC PSN_Q10302
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10302_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10303
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10304
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10304_O
```

```
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10305
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10305_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10306
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
numeric iFlag = 0;
if psn_q10104 in 3:90 and PSN_Q10303 in 1 then
    if psn_q10104 in 3 and !(psn_q10306 in 31) then
        iFlag = 1;
    endif;
    if psn_q10104 in 4 and !(psn_q10306 in 31,1) then
        iFlag = 1;
    endif;
    if psn_q10104 in 5 and !(psn_q10306 in 31,1,2) then
        iFlag = 1;
    endif;
    if psn_q10104 in 6 and !(psn_q10306 in 1,2,3) then
        iFlag = 1;
    endif;
    if psn_q10104 in 7 and !(psn_q10306 in 2,3,4) then
        iFlag = 1;
    endif;
    if psn_q10104 in 8 and !(psn_q10306 in 4,5,6) then
        iFlag = 1;
    endif;
    if psn_q10104 in 9 and !(psn_q10306 in 5,6,7) then
        iFlag = 1;
    endif;
    if psn_q10104 in 10 and !(psn_q10306 in 5,6,7) then
        iFlag = 1;
    endif;
    if psn_q10104 in 11 and !(psn_q10306 in 6,7,8) then
        iFlag = 1;
    endif;
    if psn_q10104 in 12 and !(psn_q10306 in 7,8,9) then
        iFlag = 1;
    endif;
    if psn_q10104 in 12 and !(psn_q10306 in 7,8,9) then
        iFlag = 1;
    endif;
    if psn_q10104 in 13,14 and !(psn_q10306 in 9,10,11) then
        iFlag = 1;
    endif;
    if psn_q10104 in 15 and !(psn_q10306 in 11,12,13) then
        iFlag = 1;
    endif;
    if psn_q10104 in 16,17 and !(psn_q10306 in 12,13) then
        iFlag = 1;
    endif;
    if psn_q10104 in 18 and !(psn_q10306 in 11,12,13,14,15,16) then
        iFlag = 1;
    endif;
    if psn_q10104 in 19,20,21 and !(psn_q10306 in 12,13,14,15,16) then
        iFlag = 1;
    endif;
    if psn_q10104 in 22 and !(psn_q10306 in 13,14,15,16) then
        iFlag = 1;
    endif;
    if psn_q10104 in 23 and !(psn_q10306 in 14,15,16,17,18) then
        iFlag = 1;
    endif;
    if psn_q10104 in 24:40 and !(psn_q10306 in 14,15,16,17,18,19,20) then
        iFlag = 1;
    endif;
    // lets impute
    if iFlag = 0 then
        putdeck(hd_10306_sex_age, $);
    else
        impute($,getdeck(hd_10306_sex_age));
```

```
    endif;
else
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC PSN_Q10306_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10307
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10104 in 3:90 and PSN_Q10303 in 1 then
    if $ in 1:3 then
        putdeck(hd_10307_sex_age, $);
    else
        impute($,getdeck(hd_10307_sex_age));
    endif;
else
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC PSN_Q10307_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10501
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10104 in 10:95 then
    if $ in 1:2 then
        putdeck(hd_10501_sex_age, $);
    else
        impute($,getdeck(hd_10501_sex_age));
    endif;
else
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC PSN_Q10502
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10501 in 1 then
    if $ in 1:7 then
        putdeck(hd_10502_sex_age, $);
    else
        impute($,getdeck(hd_10502_sex_age));
    endif;
else
    if $ <> notappl then
        impute($, notappl);
    endif;
endif;
PROC PSN_Q10502_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10503
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10503_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q10504
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
if psn_q10501 in 1 then
    if psn_q10503 in notappl then
        impute (psn_q10503, 9);
    endif;
    if psn_q10504 in notappl then
        impute (psn_q10504, 9);
    endif;
    if psn_q10503 = psn_q10504 then
        impute (psn_q10504, 9);
    endif;
    if psn_q10502 = psn_q10503 then
        impute (psn_q10503, 9);
    endif;
    if psn_q10502 = psn_q10504 then
```

```
        impute (psn_q10504, 9);
    endif;
    if psn_q10504 in 1:7 and psn_q10503 in 9 then
        impute(psn_q10503, psn_q10504);
        impute(psn_q10504, 9);
    endif;
else
    if psn_q10503 <> notappl then
        impute(psn_q10503, notappl);
    endif;
    if psn_q10504 <> notappl then
        impute(psn_q10504, notappl);
    endif;
endif;
if psn_q10501 in 1 then
    if psn_q10503 in 9 and psn_q10504 in 9 then
        psn_q10504 = notappl;
    endif;
endif;
PROC PSN_Q10504_O
$ = getvaluealpha(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q30602
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q30603
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q30604
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC PSN_Q30605
$ = getvaluenumeric(getsymbol($)[5:length(getsymbol($))],curocc($));
PROC POWER_EDT
preproc
    do varying i = totocc(POWER_EDT) until i <= 0 by (-1)
        if Q21404(i) in 2,notappl then
            delete(POWER_EDT(i));
        endif;
    enddo;


PROC Q21403
$ = getlabel(Q21402,Q21402);
```