

```
{ Application 'STLUCIALFS2011' logic file generated by CPro }  
PROC GLOBAL
```

```
{***** DECLARATION OF SPECIFIC VARIABLES FOR ST.LUCIA LABOUR FORCE QUESTIONNAIRE*****}
```

```
numeric  
j k housecnt popcnt emigcnt TOTPERSCOUNT HEADSCOUNT HEADPT SPOUSE SPOUSEPT settlecount sex i  
age1;
```

```
function ToggleSex(x);  
  if x = 1 then { male, make female }  
    ToggleSex = 2;  
  else { female, make male }  
    ToggleSex = 1;  
  endif;  
end; { function end }
```

```
PROC STLUCIALFS2011_FF
```

```
PROC STLUCIALFS2011_QUEST  
{ Application 'STLUCIALFS2011' logic file generated by CPro }  
PROC DISTRICT
```

```
if !($ in 1:12) then  
  recode ED => DISTRICT;  
  03002 => 02;  
  00600 => 01;  
  00801 => 02;  
  05001 => 02;  
  00901 => 01;  
  03001 => 02;  
  02504 => 02;  
  03102 => 01;  
  03105 => 01;  
  02702 => 02;  
  01102 => 01;  
  00802 => 02;  
  02203 => 02;  
  02704 => 02;  
  02706 => 02;  
  00100 => 01;  
  00200 => 01;  
  00300 => 01;  
  01405 => 02;  
  02803 => 02;  
  00502 => 01;  
  03103 => 02;  
  03203 => 01;  
  02810 => 02;  
  02901 => 02;  
  02902 => 02;  
  00400 => 01;
```

01001 => 01;  
00702 => 01;  
05002 => 02;  
05004 => 02;  
05006 => 02;  
02604 => 02;  
02607 => 02;  
02304 => 02;  
01703 => 02;  
02106 => 02;  
02107 => 02;  
03101 => 02;  
03401 => 02;  
01702 => 02;  
00701 => 02;  
00902 => 01;  
02301 => 02;  
02305 => 02;  
01201 => 01;  
02000 => 02;  
03302 => 02;  
01411 => 01;  
01401 => 02;  
02201 => 02;  
01904 => 02;  
01601 => 02;  
02101 => 02;  
02303 => 02;  
02703 => 02;  
01500 => 02;  
01412 => 02;  
01413 => 02;  
03201 => 02;  
02102 => 02;  
01101 => 01;  
01903 => 02;  
01301 => 02;  
01701 => 02;  
01406 => 02;  
00903 => 01;  
01408 => 02;  
01602 => 02;  
02302 => 02;  
01409 => 02;  
01302 => 02;  
02808 => 02;  
02809 => 02;  
01404 => 02;  
01800 => 02;  
01202 => 02;  
01402 => 02;  
03202 => 02;  
01407 => 02;  
00501 => 01;

02105 => 02;  
01403 => 02;  
01901 => 01;  
01002 => 02;  
02806 => 03;  
03506 => 03;  
03507 => 03;  
03508 => 03;  
03509 => 03;  
02805 => 03;  
02603 => 03;  
02807 => 03;  
02605 => 03;  
02606 => 03;  
02608 => 03;  
03601 => 03;  
03602 => 03;  
05003 => 03;  
05005 => 03;  
04905 => 03;  
02705 => 03;  
04907 => 03;  
04906 => 03;  
02701 => 03;  
04916 => 03;  
04600 => 03;  
04917 => 03;  
04903 => 03;  
03502 => 03;  
03802 => 03;  
03503 => 03;  
02801 => 03;  
04801 => 03;  
01410 => 03;  
03104 => 03;  
03301 => 03;  
02503 => 03;  
02505 => 03;  
04913 => 03;  
04915 => 03;  
04904 => 03;  
04902 => 03;  
03406 => 03;  
04805 => 03;  
04806 => 03;  
04807 => 03;  
04808 => 03;  
04809 => 03;  
04811 => 03;  
03504 => 03;  
04804 => 03;  
03903 => 03;  
04909 => 03;  
04201 => 03;

02104 => 03;  
03901 => 03;  
04202 => 03;  
03402 => 03;  
02802 => 03;  
04901 => 03;  
04803 => 03;  
03501 => 03;  
02403 => 03;  
02404 => 03;  
03505 => 03;  
04004 => 03;  
02804 => 03;  
04914 => 03;  
02601 => 03;  
02502 => 03;  
04001 => 03;  
03803 => 03;  
04918 => 03;  
04919 => 03;  
04920 => 03;  
04921 => 03;  
04922 => 03;  
04923 => 03;  
04802 => 03;  
04100 => 03;  
04002 => 03;  
04912 => 03;  
02202 => 03;  
04204 => 03;  
04911 => 03;  
03408 => 03;  
04908 => 03;  
03404 => 03;  
03403 => 03;  
02401 => 03;  
03405 => 03;  
03801 => 03;  
03902 => 03;  
04910 => 03;  
03702 => 03;  
04301 => 03;  
01902 => 03;  
02402 => 03;  
03703 => 03;  
03704 => 03;  
03705 => 03;  
03706 => 03;  
03904 => 03;  
03905 => 03;  
02103 => 03;  
04701 => 03;  
04702 => 03;  
04503 => 03;

04504 => 03;  
02602 => 03;  
04401 => 03;  
04203 => 03;  
03805 => 03;  
03409 => 03;  
03410 => 03;  
03407 => 03;  
04003 => 03;  
04302 => 03;  
04304 => 03;  
03701 => 03;  
04404 => 03;  
04403 => 03;  
03804 => 03;  
04402 => 03;  
05100 => 03;  
04005 => 03;  
04006 => 03;  
04007 => 03;  
02501 => 03;  
04501 => 03;  
04502 => 03;  
05802 => 04;  
05801 => 04;  
05700 => 04;  
06402 => 05;  
06201 => 04;  
05302 => 04;  
06102 => 04;  
06101 => 04;  
06203 => 04;  
06204 => 04;  
06205 => 04;  
06502 => 05;  
05200 => 04;  
06503 => 05;  
05400 => 04;  
06302 => 05;  
06401 => 05;  
06501 => 05;  
06301 => 05;  
05500 => 04;  
05901 => 04;  
06504 => 05;  
06202 => 04;  
05301 => 04;  
06001 => 04;  
05600 => 04;  
06002 => 04;  
05902 => 04;  
07101 => 06;  
06802 => 06;  
06601 => 06;

07103 => 06;  
07600 => 06;  
07503 => 06;  
06902 => 06;  
07200 => 06;  
07302 => 06;  
06602 => 06;  
07700 => 06;  
07501 => 06;  
06903 => 06;  
06904 => 06;  
07400 => 06;  
07502 => 06;  
06703 => 06;  
06704 => 06;  
07102 => 06;  
06901 => 06;  
06801 => 06;  
06701 => 06;  
07800 => 06;  
07001 => 06;  
07002 => 06;  
06702 => 06;  
07301 => 06;  
08200 => 07;  
08101 => 07;  
08602 => 07;  
08102 => 07;  
08001 => 07;  
08400 => 07;  
08601 => 07;  
07900 => 07;  
08902 => 07;  
09003 => 07;  
08502 => 07;  
08903 => 07;  
08300 => 07;  
09001 => 07;  
08700 => 07;  
08802 => 07;  
08501 => 07;  
08103 => 07;  
09004 => 07;  
09005 => 07;  
08901 => 07;  
08002 => 07;  
09002 => 07;  
08801 => 07;  
09601 => 08;  
10201 => 08;  
09500 => 08;  
10202 => 08;  
10100 => 08;  
09801 => 08;

09802 => 08;  
09602 => 08;  
09402 => 08;  
09702 => 08;  
09900 => 08;  
09102 => 08;  
09403 => 08;  
09101 => 08;  
09701 => 08;  
10000 => 08;  
09401 => 08;  
09300 => 08;  
09200 => 08;  
11402 => 09;  
10602 => 09;  
11403 => 09;  
11206 => 09;  
11207 => 10;  
11208 => 09;  
11210 => 09;  
10704 => 09;  
10705 => 09;  
10706 => 09;  
10402 => 09;  
11305 => 09;  
10301 => 09;  
11404 => 09;  
10501 => 09;  
11303 => 09;  
11209 => 09;  
11201 => 09;  
10703 => 09;  
10804 => 09;  
10805 => 09;  
11204 => 09;  
10502 => 09;  
11308 => 09;  
11203 => 09;  
11202 => 09;  
10401 => 09;  
10302 => 09;  
11401 => 09;  
11211 => 09;  
11212 => 09;  
11213 => 09;  
11214 => 10;  
11304 => 09;  
10601 => 09;  
11000 => 09;  
11307 => 09;  
10701 => 09;  
10803 => 09;  
11205 => 09;  
10802 => 09;

11302 => 09;  
10403 => 09;  
11100 => 09;  
11309 => 09;  
10806 => 09;  
10807 => 09;  
10808 => 09;  
10809 => 09;  
10801 => 09;  
10900 => 09;  
11301 => 09;  
10702 => 09;  
11306 => 09;  
12004 => 10;  
12110 => 10;  
12001 => 10;  
12107 => 10;  
12002 => 10;  
12115 => 10;  
12116 => 10;  
12109 => 10;  
12112 => 10;  
12105 => 10;  
11505 => 10;  
12204 => 10;  
12205 => 10;  
12003 => 10;  
11501 => 10;  
12501 => 10;  
12101 => 10;  
12502 => 10;  
12401 => 10;  
12202 => 10;  
12104 => 10;  
11904 => 10;  
12406 => 10;  
11901 => 10;  
11606 => 10;  
12402 => 10;  
12114 => 10;  
11604 => 10;  
12404 => 10;  
12106 => 10;  
11603 => 10;  
11504 => 10;  
11903 => 10;  
12403 => 10;  
12602 => 10;  
12603 => 10;  
11702 => 10;  
12102 => 10;  
11605 => 10;  
11905 => 10;  
12103 => 10;



12302 => 10;  
11800 => 10;  
12111 => 10;  
12108 => 10;  
11503 => 10;  
12301 => 10;  
11602 => 10;  
12203 => 10;  
11902 => 10;  
11502 => 10;  
11601 => 10;  
12113 => 10;  
12201 => 10;  
11701 => 10;  
12405 => 10;  
12601 => 10;  
13302 => 11;  
13702 => 11;  
14002 => 11;  
13402 => 11;  
14100 => 11;  
13600 => 11;  
13701 => 11;  
13502 => 11;  
13903 => 11;  
13304 => 11;  
12900 => 11;  
13902 => 11;  
13101 => 11;  
13004 => 11;  
13404 => 11;  
13405 => 11;  
12700 => 11;  
13403 => 11;  
13003 => 11;  
13401 => 11;  
14001 => 11;  
13301 => 11;  
14301 => 11;  
13703 => 11;  
13203 => 11;  
14200 => 11;  
13201 => 11;  
13501 => 11;  
12800 => 11;  
13202 => 11;  
13303 => 11;  
13002 => 11;  
13102 => 11;  
13103 => 11;  
14302 => 11;  
13800 => 11;  
13005 => 11;  
13001 => 11;

13901 => 11;  
14604 => 12;  
14606 => 12;  
14607 => 12;  
14904 => 12;  
14704 => 12;  
14605 => 12;  
14902 => 12;  
14603 => 12;  
14713 => 12;  
14720 => 12;  
15103 => 12;  
15104 => 12;  
14907 => 12;  
14706 => 12;  
15102 => 12;  
14903 => 12;  
14701 => 12;  
14906 => 12;  
15201 => 12;  
14806 => 12;  
15202 => 12;  
15101 => 12;  
14501 => 12;  
14703 => 12;  
14702 => 12;  
14804 => 12;  
15205 => 12;  
14705 => 12;  
14714 => 12;  
15206 => 12;  
14801 => 12;  
15203 => 12;  
14602 => 12;  
14716 => 12;  
14717 => 12;  
14718 => 12;  
14719 => 12;  
14720 => 12;  
14712 => 12;  
14711 => 12;  
15204 => 12;  
14715 => 12;  
14909 => 12;  
14910 => 12;  
14707 => 12;  
14601 => 12;  
14708 => 12;  
14905 => 12;  
14807 => 12;  
14803 => 12;  
14802 => 12;  
14709 => 12;  
15404 => 12;

```
15405 => 12;
15304 => 12;
15306 => 12;
15307 => 12;
14710 => 12;
14502 => 12;
14901 => 12;
14400 => 12;
14805 => 12;
15302 => 12;
15403 => 12;
14808 => 12;
14809 => 12;
15002 => 12;
15401 => 12;
15402 => 12;
15500 => 12;
15001 => 12;
15301 => 12;
15303 => 12;
15305 => 12;
15600 => 12;
14908 => 12;
endrecode;
endif;
```

```
if ($ in 1:12) then
recode ED => DISTRICT;
03002 => 02;
00600 => 01;
00801 => 02;
05001 => 02;
00901 => 01;
03001 => 02;
02504 => 02;
03102 => 01;
03105 => 01;
02702 => 02;
01102 => 01;
00802 => 02;
02203 => 02;
02704 => 02;
02706 => 02;
00100 => 01;
00200 => 01;
00300 => 01;
01405 => 02;
02803 => 02;
00502 => 01;
03103 => 02;
03203 => 01;
02810 => 02;
02901 => 02;
02902 => 02;
```

00400 => 01;  
01001 => 01;  
00702 => 01;  
05002 => 02;  
05004 => 02;  
05006 => 02;  
02604 => 02;  
02607 => 02;  
02304 => 02;  
01703 => 02;  
02106 => 02;  
02107 => 02;  
03101 => 02;  
03401 => 02;  
01702 => 02;  
00701 => 02;  
00902 => 01;  
02301 => 02;  
02305 => 02;  
01201 => 01;  
02000 => 02;  
03302 => 02;  
01411 => 01;  
01401 => 02;  
02201 => 02;  
01904 => 02;  
01601 => 02;  
02101 => 02;  
02303 => 02;  
02703 => 02;  
01500 => 02;  
01412 => 02;  
01413 => 02;  
03201 => 02;  
02102 => 02;  
01101 => 01;  
01903 => 02;  
01301 => 02;  
01701 => 02;  
01406 => 02;  
00903 => 01;  
01408 => 02;  
01602 => 02;  
02302 => 02;  
01409 => 02;  
01302 => 02;  
02808 => 02;  
02809 => 02;  
01404 => 02;  
01800 => 02;  
01202 => 02;  
01402 => 02;  
03202 => 02;  
01407 => 02;

00501 => 01;  
02105 => 02;  
01403 => 02;  
01901 => 01;  
01002 => 02;  
02806 => 03;  
03506 => 03;  
03507 => 03;  
03508 => 03;  
03509 => 03;  
02805 => 03;  
02603 => 03;  
02807 => 03;  
02605 => 03;  
02606 => 03;  
02608 => 03;  
03601 => 03;  
03602 => 03;  
05003 => 03;  
05005 => 03;  
04905 => 03;  
02705 => 03;  
04907 => 03;  
04906 => 03;  
02701 => 03;  
04916 => 03;  
04600 => 03;  
04917 => 03;  
04903 => 03;  
03502 => 03;  
03802 => 03;  
03503 => 03;  
02801 => 03;  
04801 => 03;  
01410 => 03;  
03104 => 03;  
03301 => 03;  
02503 => 03;  
02505 => 03;  
04913 => 03;  
04915 => 03;  
04904 => 03;  
04902 => 03;  
03406 => 03;  
04805 => 03;  
04806 => 03;  
04807 => 03;  
04808 => 03;  
04809 => 03;  
04811 => 03;  
03504 => 03;  
04804 => 03;  
03903 => 03;  
04909 => 03;

04201 => 03;  
02104 => 03;  
03901 => 03;  
04202 => 03;  
03402 => 03;  
02802 => 03;  
04901 => 03;  
04803 => 03;  
03501 => 03;  
02403 => 03;  
02404 => 03;  
03505 => 03;  
04004 => 03;  
02804 => 03;  
04914 => 03;  
02601 => 03;  
02502 => 03;  
04001 => 03;  
03803 => 03;  
04918 => 03;  
04919 => 03;  
04920 => 03;  
04921 => 03;  
04922 => 03;  
04923 => 03;  
04802 => 03;  
04100 => 03;  
04002 => 03;  
04912 => 03;  
02202 => 03;  
04204 => 03;  
04911 => 03;  
03408 => 03;  
04908 => 03;  
03404 => 03;  
03403 => 03;  
02401 => 03;  
03405 => 03;  
03801 => 03;  
03902 => 03;  
04910 => 03;  
03702 => 03;  
04301 => 03;  
01902 => 03;  
02402 => 03;  
03703 => 03;  
03704 => 03;  
03705 => 03;  
03706 => 03;  
03904 => 03;  
03905 => 03;  
02103 => 03;  
04701 => 03;  
04702 => 03;

04503 => 03;  
04504 => 03;  
02602 => 03;  
04401 => 03;  
04203 => 03;  
03805 => 03;  
03409 => 03;  
03410 => 03;  
03407 => 03;  
04003 => 03;  
04302 => 03;  
04304 => 03;  
03701 => 03;  
04404 => 03;  
04403 => 03;  
03804 => 03;  
04402 => 03;  
05100 => 03;  
04005 => 03;  
04006 => 03;  
04007 => 03;  
02501 => 03;  
04501 => 03;  
04502 => 03;  
05802 => 04;  
05801 => 04;  
05700 => 04;  
06402 => 05;  
06201 => 04;  
05302 => 04;  
06102 => 04;  
06101 => 04;  
06203 => 04;  
06204 => 04;  
06205 => 04;  
06502 => 05;  
05200 => 04;  
06503 => 05;  
05400 => 04;  
06302 => 05;  
06401 => 05;  
06501 => 05;  
06301 => 05;  
05500 => 04;  
05901 => 04;  
06504 => 05;  
06202 => 04;  
05301 => 04;  
06001 => 04;  
05600 => 04;  
06002 => 04;  
05902 => 04;  
07101 => 06;  
06802 => 06;

06601 => 06;  
07103 => 06;  
07600 => 06;  
07503 => 06;  
06902 => 06;  
07200 => 06;  
07302 => 06;  
06602 => 06;  
07700 => 06;  
07501 => 06;  
06903 => 06;  
06904 => 06;  
07400 => 06;  
07502 => 06;  
06703 => 06;  
06704 => 06;  
07102 => 06;  
06901 => 06;  
06801 => 06;  
06701 => 06;  
07800 => 06;  
07001 => 06;  
07002 => 06;  
06702 => 06;  
07301 => 06;  
08200 => 07;  
08101 => 07;  
08602 => 07;  
08102 => 07;  
08001 => 07;  
08400 => 07;  
08601 => 07;  
07900 => 07;  
08902 => 07;  
09003 => 07;  
08502 => 07;  
08903 => 07;  
08300 => 07;  
09001 => 07;  
08700 => 07;  
08802 => 07;  
08501 => 07;  
08103 => 07;  
09004 => 07;  
09005 => 07;  
08901 => 07;  
08002 => 07;  
09002 => 07;  
08801 => 07;  
09601 => 08;  
10201 => 08;  
09500 => 08;  
10202 => 08;  
10100 => 08;



09801 => 08;  
09802 => 08;  
09602 => 08;  
09402 => 08;  
09702 => 08;  
09900 => 08;  
09102 => 08;  
09403 => 08;  
09101 => 08;  
09701 => 08;  
10000 => 08;  
09401 => 08;  
09300 => 08;  
09200 => 08;  
11402 => 09;  
10602 => 09;  
11403 => 09;  
11206 => 09;  
11207 => 10;  
11208 => 09;  
11210 => 09;  
10704 => 09;  
10705 => 09;  
10706 => 09;  
10402 => 09;  
11305 => 09;  
10301 => 09;  
11404 => 09;  
10501 => 09;  
11303 => 09;  
11209 => 09;  
11201 => 09;  
10703 => 09;  
10804 => 09;  
10805 => 09;  
11204 => 09;  
10502 => 09;  
11308 => 09;  
11203 => 09;  
11202 => 09;  
10401 => 09;  
10302 => 09;  
11401 => 09;  
11211 => 09;  
11212 => 09;  
11213 => 09;  
11214 => 10;  
11304 => 09;  
10601 => 09;  
11000 => 09;  
11307 => 09;  
10701 => 09;  
10803 => 09;  
11205 => 09;

10802 => 09;  
11302 => 09;  
10403 => 09;  
11100 => 09;  
11309 => 09;  
10806 => 09;  
10807 => 09;  
10808 => 09;  
10809 => 09;  
10801 => 09;  
10900 => 09;  
11301 => 09;  
10702 => 09;  
11306 => 09;  
12004 => 10;  
12110 => 10;  
12001 => 10;  
12107 => 10;  
12002 => 10;  
12115 => 10;  
12116 => 10;  
12109 => 10;  
12112 => 10;  
12105 => 10;  
11505 => 10;  
12204 => 10;  
12205 => 10;  
12003 => 10;  
11501 => 10;  
12501 => 10;  
12101 => 10;  
12502 => 10;  
12401 => 10;  
12202 => 10;  
12104 => 10;  
11904 => 10;  
12406 => 10;  
11901 => 10;  
11606 => 10;  
12402 => 10;  
12114 => 10;  
11604 => 10;  
12404 => 10;  
12106 => 10;  
11603 => 10;  
11504 => 10;  
11903 => 10;  
12403 => 10;  
12602 => 10;  
12603 => 10;  
11702 => 10;  
12102 => 10;  
11605 => 10;  
11905 => 10;

12103 => 10;  
12302 => 10;  
11800 => 10;  
12111 => 10;  
12108 => 10;  
11503 => 10;  
12301 => 10;  
11602 => 10;  
12203 => 10;  
11902 => 10;  
11502 => 10;  
11601 => 10;  
12113 => 10;  
12201 => 10;  
11701 => 10;  
12405 => 10;  
12601 => 10;  
13302 => 11;  
13702 => 11;  
14002 => 11;  
13402 => 11;  
14100 => 11;  
13600 => 11;  
13701 => 11;  
13502 => 11;  
13903 => 11;  
13304 => 11;  
12900 => 11;  
13902 => 11;  
13101 => 11;  
13004 => 11;  
13404 => 11;  
13405 => 11;  
12700 => 11;  
13403 => 11;  
13003 => 11;  
13401 => 11;  
14001 => 11;  
13301 => 11;  
14301 => 11;  
13703 => 11;  
13203 => 11;  
14200 => 11;  
13201 => 11;  
13501 => 11;  
12800 => 11;  
13202 => 11;  
13303 => 11;  
13002 => 11;  
13102 => 11;  
13103 => 11;  
14302 => 11;  
13800 => 11;  
13005 => 11;

13001 => 11;  
13901 => 11;  
14604 => 12;  
14606 => 12;  
14607 => 12;  
14904 => 12;  
14704 => 12;  
14605 => 12;  
14902 => 12;  
14603 => 12;  
14713 => 12;  
14720 => 12;  
15103 => 12;  
15104 => 12;  
14907 => 12;  
14706 => 12;  
15102 => 12;  
14903 => 12;  
14701 => 12;  
14906 => 12;  
15201 => 12;  
14806 => 12;  
15202 => 12;  
15101 => 12;  
14501 => 12;  
14703 => 12;  
14702 => 12;  
14804 => 12;  
15205 => 12;  
14705 => 12;  
14714 => 12;  
15206 => 12;  
14801 => 12;  
15203 => 12;  
14602 => 12;  
14716 => 12;  
14717 => 12;  
14718 => 12;  
14719 => 12;  
14720 => 12;  
14712 => 12;  
14711 => 12;  
15204 => 12;  
14715 => 12;  
14909 => 12;  
14910 => 12;  
14707 => 12;  
14601 => 12;  
14708 => 12;  
14905 => 12;  
14807 => 12;  
14803 => 12;  
14802 => 12;  
14709 => 12;

```

15404 => 12;
15405 => 12;
15304 => 12;
15306 => 12;
15307 => 12;
14710 => 12;
14502 => 12;
14901 => 12;
14400 => 12;
14805 => 12;
15302 => 12;
15403 => 12;
14808 => 12;
14809 => 12;
15002 => 12;
15401 => 12;
15402 => 12;
15500 => 12;
15001 => 12;
15301 => 12;
15303 => 12;
15305 => 12;
15600 => 12;
14908 => 12;
endrecode;
endif;
PROC WU
if DISTRICT=01 then
    impute ($, 52.6252608);
endif;

if DISTRICT in 02,03 then
    impute ($, 71.7934582);
endif;

if DISTRICT in 04,05 then
    impute ($, 70.5556399);
endif;
if DISTRICT=06 then
    impute ($, 61.0541471);
endif;
if DISTRICT=07 then
    impute ($, 66.9586858);
endif;
if DISTRICT=08 then
    impute ($, 71.1436036);
endif;
if DISTRICT=09 then
    impute ($, 71.5428382);
endif;
if DISTRICT=10 then
    impute ($, 60.8087784);
endif;
if DISTRICT=11 then

```

```

        impute ($, 78.9728040);
    endif;
    if DISTRICT=12 then
        impute ($, 79.3651294);
    endif;
PROC PERSON_EDT

preproc
popcnt = popcnt + 1;
sex = 1;
age1 = 15; {minimum age to be head}

{do varying i = 1 until i >= totocc(PERSON_EDT)
  if !(P6(i) in 1:98) THEN
      errmsg ("*P00-1* Deleted records - ED - [%2d], - Household -[%2d], - Pno -[%2d], pers rec
[%2d]",ED,HHNO,Pno(i),i) denom = popcnt summary;
      write ("*P00-1* Deleted records - ED - %d - HH -%d - Pno -%d pers rec %d ",ED,HHNO,Pno(i),i);

      delete (PERSON_EDT(i)); { remove "blank" person records }
  endif;
enddo;

}
{
The following procedure deletes superfluous records

}
{
  i = 1;
  WHILE i <= totocc(PERSON_EDT) DO

      IF //(P2(i) = 0) AND
      (P3(i) = 1) OR (P3(i) = 2) THEN
          //(P6(i) = 0) then
          errmsg ("*P00-1* Deleted records - ED - [%2d], - Household -[%2d], - Pno -[%2d], pers rec
[%2d]",ED,HHNO,Pno(i),i) denom = popcnt summary;
          write ("*P00-1* Deleted records - ED - %d - HH -%d - Pno -%d pers rec %d ",ED,HHNO,Pno(i),i);
          delete (PERSON_EDT(i)); { remove "blank" population records }

      ENDIF;

      i = i + 1;

  enddo; }
PROC P2
{Head 1
Spouse 2
Child 3
Parent 4
Other Relative 5
Employee 6
Non Relative 7
Other 9

```

Not Stated    0    }

```
if !($ in 1:7) and !($ in 8) then
    impute ($, 8);
endif;
```

```
errmsg (" ***** RELATIONSHIP ***** "), summary;
```

```
HEADS = 0;
```

```
HEADS = count(PERSON_EDT where P2 = 1);    {. Count the heads}
HEADPT = 0;
```

```
if HEADS <> 0 then
```

```
    do varying i = 1 until i > TOTOCC(PERSON_EDT)
        if P2(i) = 1 AND P6(i) >= age1 then
            HEADPT = i;
            break; {index for person with head found so break from loop}
        endif;
    enddo;
```

```
endif;
```

```
if TOTOCC (PERSON_EDT) = 1 then
```

```
    if P2 <> 1 then
```

```
        errmsg ("*35-1* Scanner problem: making only person in house head of household"), denom = popcnt summary;
        write ("*35-1* Scanner problem: making only person in house head of household");
        impute(P2, 1);
```

```
    endif;
```

```
endif;
```

```
if HEADS > 1 then
```

```
    errmsg("*35-3_1* Too many heads of household") denom = popcnt summary;
    write("*35-3_1* Too many heads of household, spouses= [%02d] ", HEADS);
    {if there is more than one code above for HEAD > 0 has already set the pointer}
```

```
    do varying k = 1 until k > TOTOCC(PERSON_EDT)
        if k <> HEADPT then
            if P2 (k) = 1 then {eliminate remaining heads}
```

```
                errmsg ("*35-3_2* Remaining spouses made other relative, pn [%d]",Pno(j)) denom = popcnt
```

```
summary;
```

```
                write ("*35-3_2* Remaining spouses made other relative, pn= [%02d], relat= [%02d]",Pno(j),P2(j));
                impute (P2(k),5);
                HEADS = HEADS - 1;
```

```
            endif;
```

```
        endif;
```

```
    enddo;
```

```

endif;

if not P2 in 1:9 then
  errmsg ("*35-2* Illegal P2_RELATIONSHIP code, made other relative") denom = popcnt summary;
  write ("*35-2* Illegal P2_RELATIONSHIP code, made other relative, pn = [%2d], relat = [%2d]",Pno,P2);
  impute (P2,5);

{Head  1
Spouse 2
Child  3
Parent 4
Other Relative 5
Employee 6
Non Relative 7
Other 9
Not Stated 0  }

elseif P2 in 1:9 then

  if P2 = 2 then {if person is spouse or partner, check if there are more than one in record}
  { Checking for spouses -- count spouses and partners }

    SPOUSE = count(PERSON_EDT where P2 =2);

    {write ("Number of spouses = ", SPOUSE); }

    SPOUSEPT = 0;

    { If more than one spouse, edit out extra spouses }

    if SPOUSE > 0 then

      {write ("Spouse pointer because only one spouse %2d",SPOUSEPT);}

      do varying i = 1 until i > TOTOCC (PERSON_EDT)
      if P2(i) = 2 then
        SPOUSEPT = i;
        break;
      endif;
    enddo;
  endif;

  if SPOUSE > 1 then

    errmsg("*35-3_1* Too many spouses of household") denom = popcnt summary;
    write("*35-3_1* Too many spouses of household, spouses= [%02d] ", SPOUSE);
    {if there is more than one then code above for spouse > 0 has already set the pointer}

    do varying j = 1 until j > TOTOCC(PERSON_EDT)
    if j <> SPOUSEPT then
      if P2 (j) = 2 then {eliminate remaining spouses or partners }

```





```

if $=0 then
    impute($,9);
endif;
else
    impute($,8);
endif;
PROC P8
if (P6 in 15:98) then
    if $=0 then
        impute($,99);
    endif;
else
    impute($,88);
endif;

If (P6 in 0:14) then
    impute($,88);
endif;
PROC P9
if $=0 then
    impute($,9);
endif;
PROC P10
if (P6 in 15:98) then
    if $=0 then
        impute($,9);
    endif;
endif;
PROC P11C
if $<1000 then
    impute($,9999);
endif;
PROC P123
if ($ > 26) then
    if !(P13 in 1:6) then
        impute(P13, 9);
    endif;
endif;
PROC P13
if $=0 then
    impute($,8);
endif;
PROC P14
{ 14. ECONOMIC ACTIVITY

```

Did you work for pay, profit or family gain, during the week ending.../.../...?

NB. Domestic work at home is excluded

1. Yes (go to PART 3, Q.21)
2. No}

{If P14 is 0 and the age is legal and if P21 is 1 or 2 then P14 must be 1 else it is 2}

```
{If P14 is 1 and the age is legal then if P21 is not 1 or 2 then make it 1}
{If P14 is 1 then all variables P15 to P20 must be assigned not applicable}
```

```
if (P6 in 15:98) and $=0 then
  if P21=1 or P21=2 then
    impute($,1);
  else
    impute($,2);
  endif;
endif;
```

```
if (P6 in 15:98) and P14=1 then
  if !(P21 in 1:2) then
    impute(P21, 1);
  endif;
endif;
```

```
if (P6 in 15:98) and P14=1 then
  impute(P15,8);
  impute(P16,8);
  impute(P17,8);
  impute(P18,8);
  impute(P181,8);
  impute(P182,8);
  impute(P183,8);
  impute(P184,8);
  impute(P185,8);
  impute(P186,8);
  impute(P187,8);
  impute(P188,8);
  impute(P189,8);
  impute(P1810,8);
  impute(P19,88);
  impute(P20,8);
endif;
```

#### PROC P15

{P15 - Temporary Absence

Yes 1

No 2

Not Stated 9

Not Stated 0

Not Applicable 8 }

{This procedure imputed a value of 9, not stated where the value of P14 is 2,  
if the value is not 2 then P15 is not applicable}

```
if P14=2 then
  if !($ in 1,2) then
    impute($,9);
  endif;
else
  impute($,8);
endif;
```

PROC P16  
{ 16. WHY ABSENT

Why were you (was...) absent from work during week ending .../.../...?

1. Vacation
2. Maternity Leave
3. Sick leave
4. Temporary Lay-off
5. Other. Please specify}

```
if ($ in 1:4) then
    if !(P21 in 1:2) then
        impute(P21, 1);
    endif;
elseif $=5 then
    impute($,5);
elseif !($ in 1:5) then
    if P14=2 and P15=1 then
        impute($,9);
    endif;
endif;
```

PROC P17  
{ 17. WANTING WORK

Did you (...) want to work during week ending ... /.../...?

1. Yes
2. No (go to PART 5 QU.51)
3. Have Job(Go to Q21)}

```
If (p6 in 15:98) then
if $=2 then
    if !(p51 in 1:2) then
        impute(p51,9);
    endif;
endif;
```

```
if $=3 then
    if !(p21 in 1:2) then
        impute(p21,1);
    endif;
endif;
```

```
if P14=2 and P15=1 and !($ in 1:3) then
    impute($, 9);
```

```

endif;
endif
PROC P181

If $=1 then
    if !(P19 in 1:14) then
        impute($,99);
    endif;
elseif $=0 and (P181=1 or P182=1 or P183=1 or P184=1 or P185=1 or P186=1 or P187=1 or P188=1 or P189=1 or
P1810=1) then
    if (P19 in 0:14) then
        impute($,88);
    endif;
endif;
PROC P20
If (p6 in 15:98) then
    if P17=1 then
        if !(P20 in 1,2,3,4,5,9) then
            impute($,9);
        elseif $=5 and !(P43 in 1:4) then
            impute(P43,9);
        elseif $=9 and !(P43 in 1:4) then
            impute(P43,9);
        elseif (P20 in 1:4) and P51=0 then
            impute(P51,9);
        endif;
    endif;
endif;
PROC P21
if (P6 in 15:98) then
    if !($ in 1:2) and P14=1 then
        impute($,1);
    endif;
endif;
{if !($ in 1:2) then
    impute(P24,88);
    impute(note24,8);
    impute(P25,88);
    impute(P26,8);
    impute(P27,0);
    impute(P27a,8);
    impute(P27b,8);
    impute(P27c,8);
    impute(P27d,8);
    impute(P27e,8);
    impute(P27f,8);
    impute(P27g,8);
    impute(P28c,8888);
    impute(P29c,8888);
    impute(P30,8);
endif;}
PROC P22
{ //PROC P22
//If !(P21=1) then

```

```

//    impute($,000);
//elseif (P21=1) then
//    if P22=0 or P22>130 then
//        impute($,35);
//    endif;
//endif;}
PROC P23
{///PROC P23
//If !(P21=1) then
//    impute($,000);
//elseif (P21=1) then
//    if $=0 or $>130 then
//        impute($,35);
//    endif;
//endif;    }
PROC P24
If P21=1 then
    if P22>0 and P23>0 then
        if P22=P23 then
            impute($,1);
        elseif P22>P23 then
            if !($ in 2:6) then
                impute($,99);
            endif;
        elseif P22<P23 then
            if !($ in 11:12) then
                impute($,99);
            endif;
        endif;
    endif;
endif;

PROC NOTE24
if P23>34 then
    impute($,1);
    impute(P25,88);
endif;
PROC P25
if NOTE24=2 and $=0 then
    impute($,99);
endif;
PROC P26
if P23>34 and $=0 then
    impute($,9);
endif;
PROC P27

PROC P27A
if P27 in 1:4,8 then
    impute($,8);
elseif P27 in 5,6,8,9 then
    impute($,9);
endif;

```

```

if P27A in 1:4 then
    impute(P27B,8);
    impute(P27C,8);
    impute(P27D,8);
    impute(P27E,8);
endif;

PROC P27B
If $=0 and P27 in 1:4,8 then
    impute($,9);
endif;
PROC P27C
If $=0 and P27 in 1:4,8 then
    impute($,9);
endif;
PROC P27D
If $=0 and P27 in 1:4,8 then
    impute($,9);
endif;
PROC P27E
If $=0 and P27 in 1:4,8 then
    impute($,9);
endif;
if P27E in 1:4,9 then
    impute(P27F,8);
    impute(P27G,8);
endif;
PROC P27F
if P27 in 5,6,8,9 and $=0 then
    impute($,9);
endif;
PROC P27G
if P27 in 5,6,8,9 and $=0 then
    impute($,9);
endif;
PROC P31
if P21=1 then
    if !($ in 1:9) then
        impute($,9);
    endif;
endif;
PROC P43
if $=0 then
    impute($,8);
endif;
{if $ in 1:4 then
    impute(P51,8);
    impute(p52,8);
    impute(P53,0);
    impute(P54c,8888);
    impute(P55c,8888);
    impute(P56,8);
    impute(P57,8);
}

```

```

endif;}
PROC P44
{if !(p44 in 1:2) then
    if P43 in 1:4 then
        if p6<22 then
            impute($,2);
        else
            impute($,1);
        endif;
    endif;
elseif !(p44 in 1:2) and (P43 in 1:4) then
    impute($,9);
else
    impute($,1);
endif;

if $=2 then
    impute(P45,0);
    impute(p46,0);
    impute(p47c,8888);
    impute(p48c,8888);
endif;}
PROC P49
if p44 in 1:4 then
    if p49=0 then
        impute($,9);
    endif;
else
    impute($,8);
endif;
PROC P50
if p44 in 1:4 then
    if $=0 then
        impute($,9);
    endif;
else
    impute($,8);
endif;
PROC P51
if $=0 then
    impute($,8);
endif;
if $=2 then
    impute(p52,8);
    impute(P53,0);
    impute(P54c,8888);
    impute(P55c,8888);
endif;
PROC P56
if P51 in 1:2 then
    if $=0 then
        impute($,9);
    endif;
endif;

```



```

PROC P57
if P51 in 1:2 then
    if $=0 then
        impute($,9);
    endif;
endif;
PROC P18
If P17=1 then
    If (P181=0 and P182=0 and P183=0 and P184=0 and P185=0 and P186=0 and P187=0 and P188=0 and P189=0
and P1810=0) then
        impute($,9);
    endif;
endif;

if p6 in 15:98 then
    if !($ in 1,2) then
        if P181 = 1 then
            errmsg ("*XX-X No method of Seeking [%2d], so impute",P181) denom = popcnt summary;
            write ("*XX-X No method of Seeking [%2d], so impute [%2d], so imputed, PN = [%2d]",$,P181,
Pno);

            impute ($, 1);
        endif;
        if P182 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P182) denom = popcnt summary;
            write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P182, Pno);
            impute ($, 2);
        endif;
        if P183 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P183) denom = popcnt summary;
            write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P183, Pno);
            impute ($, 2);
        endif;
        if P184 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P184) denom = popcnt summary;
            write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P184, Pno);
            impute ($, 2);
        endif;
        if P185 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P185) denom = popcnt summary;
            write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P185, Pno);
            impute ($, 2);
        endif;
        if P186 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P186) denom = popcnt summary;
            write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P186, Pno);
            impute ($, 2);
        endif;
        if P187 = 1 then
            errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P187) denom = popcnt summary;

```

```

        write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P187, Pno);
        impute ($, 2);
    endif;
    if P188 = 1 then
        errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P188) denom = popcnt summary;
        write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P188, Pno);
        impute ($, 2);
    endif;
    if P189 = 1 then
        errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P189) denom = popcnt summary;
        write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P189, Pno);
        impute ($, 2);
    endif;
    if P1810 = 1 then
        errmsg ("*XX-X This is a method of Seeking [%2d], so impute",P1810) denom = popcnt summary;
        write ("*XX-X This is a method of Seeking [%2d], so impute [%2d], so imputed, PN =
[%2d]",$,P1810, Pno);
        impute ($, 2);
    endif;
else
    errmsg ("*XX-X No Response to this Question [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X No Response to this questoin [%2d], PN = [%2d]", $, Pno);
    impute ($, 9);
endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
endif;
PROC P18C
if P181=1 then
    impute($,1);
elseif P182=1 then
    impute($,2);
elseif P183=1 then
    impute($,3);
elseif P184=1 then
    impute($,4);
elseif P185=1 then
    impute($,5);
elseif P186=1 then
    impute($,6);
elseif P187=1 then
    impute($,7);
elseif P188=1 then
    impute($,8);
elseif P189=1 then
    impute($,9);
elseif P1810=1 then
    impute($,10);
endif;

```

PROC EMP

{if (p21=1 or p21=2 or p14=1) emp=1.

if (emp=1 and p17=1) emp=3. }

if p6 in 15:98 then

if P21 = 1 or P21 = 2 or P14 = 1 then

errmsg ("\*XX-X Person is employed [%2d], so impute",\$) denom = popcnt summary;

write ("\*XX-X Person is employed [%2d], Activity last week [%2d], Multiple Job [%2d], PN = [%2d]", \$,

P21, P14, Pno);

impute (\$, 1);

if p28c=0 then

impute (p28c, 9999);

endif;

if p29c=0 then

impute (p29c, 9999);

endif;

if p38c=0 then

impute (p38c, 9999);

endif;

if p39c=0 then

impute (p39c, 9999);

endif;

if p381c=0 then

impute (p381c, 9999);

endif;

if p391c=0 then

impute (p391c, 9999);

endif;

else

errmsg ("\*XX-X Person is not employed [%2d], so impute",\$) denom = popcnt summary;

write ("\*XX-X Person is employed [%2d], Activity last week [%2d], Multiple Job [%2d], PN = [%2d]", \$,

P21, P14, Pno);

impute (\$, 2);

if p28c=0 then

impute (p28c, 9999);

endif;

if p29c=0 then

impute (p29c, 9999);

endif;

if p38c=0 then

impute (p38c, 9999);

endif;

if p39c=0 then

impute (p39c, 9999);

endif;

if p381c=0 then

impute (p381c, 9999);

endif;

if p391c=0 then

impute (p391c, 9999);

endif;

endif;

else

errmsg ("\*XX-X Person is under 15 years [%2d], so impute",\$) denom = popcnt summary;

```

write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
impute ($, 8);
    impute (p28c, 8888);
    impute (p29c, 8888);
    impute (p38c, 8888);
    impute (p39c, 8888);
    impute (p381c, 8888);
    impute (p391c, 8888);
endif;
PROC UNEMP
{if ((p18=2) or (p19=4 or p19=5 or p19=6 or p19=7 or p19=9 or p19=10 or
p19=11 or p19=12 or p19=13 and p20=9)) unemp=1.}

if p6 in 15:98 then
    if ((p18=2) or (p19=4 or p19=5 or p19=6 or p19=7 or p19=9 or p19=10 or p19=11 or p19=12 or p19=13 and
p20=9)) then
        errmsg ("*XX-X Person is unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is Seeking [%2d], PN = [%2d]", $, P18, Pno);
        impute ($, 1);
        if p47c=0 then
            impute (p47c, 9999);
        endif;
        if p48c=0 then
            impute (p48c, 9999);
        endif;
    else
        errmsg ("*XX-X Person is not unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is Seeking [%2d], Why not Seeking [%2d], PN = [%2d]",
$, P18, P19, Pno);
        impute ($, 2);
        if p47c=0 then
            impute (p47c, 8888);
        endif;
        if p48c=0 then
            impute (p48c, 8888);
        endif;
    endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute",$) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
    if p47c=0 then
        impute (p47c, 8888);
    endif;
    if p48c=0 then
        impute (p48c, 8888);
    endif;
endif;
if p6 in 15:98 then
    if (emp = 1 and unemp=1) then
        errmsg ("*XX-X Person is unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is employed [%2d], Is unemployed [%2d], PN = [%2d]",
$, emp, unemp, Pno);
        impute ($, 2);
    endif;
endif;

```

```

        if p47c=0 then
            impute (p47c, 8888);
        endif;
        if p48c=0 then
            impute (p48c, 8888);
        endif;
    endif;
endif;
PROC UNEMP1
{if ((p18a=2) unemp=1.)}

if p6 in 15:98 then
    if (p18=2) then
        errmsg ("*XX-X Person is unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is Seeking [%2d], PN = [%2d]", $, P18, Pno);
        impute ($, 1);
        if p47c=0 then
            impute (p47c, 9999);
        endif;
        if p48c=0 then
            impute (p48c, 9999);
        endif;
    else
        errmsg ("*XX-X Person is not unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is Seeking [%2d], Why not Seeking [%2d], PN = [%2d]",
$, P18, P19, Pno);
        impute ($, 2);
        if p47c=0 then
            impute (p47c, 8888);
        endif;
        if p48c=0 then
            impute (p48c, 8888);
        endif;
    endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute",$) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
    if p47c=0 then
        impute (p47c, 8888);
    endif;
    if p48c=0 then
        impute (p48c, 8888);
    endif;
endif;
if p6 in 15:98 then
    if (emp = 1 and unemp1=1) then
        errmsg ("*XX-X Person is unemployed [%2d], so impute",$) denom = popcnt summary;
        write ("*XX-X Person is unemployed [%2d], Is employed [%2d], Is unemployed [%2d], PN = [%2d]",
$, emp, unemp, Pno);
        impute ($, 2);
        if p47c in 0:9999 then
            impute (p47c, 8888);
        endif;
    endif;
endif;

```

```

        if p48c in 0:9999 then
            impute (p48c, 8888);
        endif;
    endif;
endif;
PROC LABFOR
{if (unemp=1 or emp=1) labfor=1.}

if p6 in 15:98 then
if (unemp=1 or emp=1) then
    errmsg ("*XX-X Person is in the Labour Force [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is in the Labour Force[%2d], Is employed [%2d], Is unemployed [%2d], PN =
[%2d]", $, emp, unemp, Pno);
    impute ($, 1);
else
    errmsg ("*XX-X Person is not in the Labour Force [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is not in the Labour Force [%2d], Is employed [%2d], Is unemployed [%2d], PN =
[%2d]", $, emp, unemp, Pno);
    impute ($, 2);
endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
endif;
PROC LABFOR1
if p6 in 15:98 then
if (unemp1=1 or emp=1) then
    errmsg ("*XX-X Person is in the Labour Force [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is in the Labour Force[%2d], Is employed [%2d], Is unemployed [%2d], PN =
[%2d]", $, emp, unemp1, Pno);
    impute ($, 1);
else
    errmsg ("*XX-X Person is not in the Labour Force [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is not in the Labour Force [%2d], Is employed [%2d], Is unemployed [%2d], PN =
[%2d]", $, emp, unemp1, Pno);
    impute ($, 2);
endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
endif;
PROC NILF
{if (labfor=2) nilf=1.}
if p6 in 15:98 then
if (labfor=2) then
    impute($,1);
    if p54c=0 then
        impute (p54c, 9999);
    endif;
    if p55c=0 then
        impute (p55c, 9999);
    endif;
endif;

```

```

else
impute($,2);
  if p54c=0 then
    impute (p54c, 8888);
  endif;
  if p55c=0 then
    impute (p55c, 8888);
  endif;
endif;
else
  errmsg ("*XX-X Person is under 15 years [%2d], so impute",$) denom = popcnt summary;
  write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
  impute ($, 8);
    if p54c=0 then
      impute (p54c, 8888);
    endif;
    if p55c=0 then
      impute (p55c, 8888);
    endif;
endif;
PROC NILF1
{if (labfor1=2) nilf1=1.}
if p6 in 15:98 then
if (labfor1=2) then
  impute($,1);
    if p54c=0 then
      impute (p54c, 9999);
    endif;
    if p55c=0 then
      impute (p55c, 9999);
    endif;
  else
    impute($,2);
      if p54c=0 then
        impute (p54c, 8888);
      endif;
      if p55c=0 then
        impute (p55c, 8888);
      endif;
    endif;
  else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute",$) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
      if p54c=0 then
        impute (p54c, 8888);
      endif;
      if p55c=0 then
        impute (p55c, 8888);
      endif;
    endif;
  endif;
PROC FIRS
{if (p20=9 AND p18=2 AND p17=1 and p44=2) firs=1.}

```

```

if p6 in 15:98 then
if (p18=2 and p44=2) then
    errmsg ("*XX-X Person is a First Seeker [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is a First Seeker [%2d], Is unemployed [%2d], PN = [%2d]", $, unemp, Pno);
    impute ($, 1);
else
    errmsg ("*XX-X Person is not a First Seeker [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is not a First Seeker [%2d], Is employed [%2d], Is unemployed [%2d], PN = [%2d]",
$, emp, unemp, Pno);
    impute ($, 2);
endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
endif;
PROC FUT
{if ((p44=2) or ( p57=1)) fut=1.}
if p6 in 15:98 then
if ((p44=2) or ( p57=1)) then
    errmsg ("*XX-X Person is a Future Seeker [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is a Future Seeker [%2d], Is unemployed [%2d], PN = [%2d]", $, unemp, Pno);
    impute ($, 1);
else
    errmsg ("*XX-X Person is not a Future Seeker [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is not a Future Seeker [%2d], Is employed [%2d], Is unemployed [%2d], PN =
[%2d]", $, emp, unemp, Pno);
    impute ($, 2);
endif;
else
    errmsg ("*XX-X Person is under 15 years [%2d], so impute", $) denom = popcnt summary;
    write ("*XX-X Person is under 15 years [%2d], PN = [%2d]", P6, Pno);
    impute ($, 8);
endif;

PROC PERSONS
PERSONS = 0;
$ = count(PERSON_EDT); { . Count the Persons}
PROC HEADS
HEADS = 0;

$ = count(PERSON_EDT where P2 = 1 ); { . Count the heads}
HEADPT = 0;
HEADSCOUNT = $;
{ If one head, note head's line number with head pointer }

if HEADS <> 0 then

    for i in PERSON_EDT do
        if P2(i) = 1 then
            HEADPT = i;
            break; {index for person with head found so break from loop}

```



```
endif;  
enddo;
```

```
endif;  
PROC WT  
if DISTRICT=01 then  
    impute ($, 52.6252608);  
endif;  
  
if DISTRICT in 02,03 then  
    impute ($, 71.7934582);  
endif;  
  
if DISTRICT in 04,05 then  
    impute ($, 70.5556399);  
endif;  
if DISTRICT=06 then  
    impute ($, 61.0541471);  
endif;  
if DISTRICT=07 then  
    impute ($, 66.9586858);  
endif;  
if DISTRICT=08 then  
    impute ($, 71.1436036);  
endif;  
if DISTRICT=09 then  
    impute ($, 71.5428382);  
endif;  
if DISTRICT=10 then  
    impute ($, 60.8087784);  
endif;  
if DISTRICT=11 then  
    impute ($, 78.9728040);  
endif;  
if DISTRICT=12 then  
    impute ($, 79.3651294);  
endif;
```