NLSY79 APPENDIX 2:
TOTAL NET FAMILY INCOME VARIABLE CREATION: 1979-2000

VARIABLE CREATION: TOTAL NET FAMILY INCOME 1979–2000

/* 1979–83 /

```
   DCL   1 FAMILY_INCOME (5),              5 AFDC,            5 ALIM,            5 CHSP,
         5 CPS,                            5 ED,              5 EDSS,            5 FAMILY,
         5 FAMINC,                         5 FARM,                     5 FOOD,                     5 GIFT,
         5 INC,                            5 INCOME,                   5 LEVEL,                    5 MIL,
         5 MILS,                           5 MN (3),                   5 OTHER,                    5
POVERTY,
         5 PUBLIC,                         5 R,               5 RELREG,                   5 RELWEL,
         5 S,                              5 SEI,             5 SEIS,            5 SUMN,
         5 TABLE (3,2,15),                          5 UI,             5 UIS,             5 VET,
         5 WELF,                                    5 WPS,            5 WPSS,            5 YOUTH;
```

/* 1979 VARIABLES */

```
        INC(1)=R(1903.10);
         if R(1547.)=-4 then MIL(1)=0;                       else MIL=R(1547.);
         if R(1554.)=-4 then WPS(1)=0;                       else WPS(1)=R(1554.);
         if R(1560.)=-4 then SEI(1)=0;               else SEI(1)=R(1560.);
         if R(1588.)=-4 & R(1590.)=–4 then UI(1)=0;
        else if R(1588.)>0 & R(1590.)>0 then UI(1)=R(1588.) * R(1590.);
        else UI(1)=R(1590.);
        MILS(1)=0;
         if R(1555.)=-4 then WPSS(1)=0;                      else WPSS(1)=R(1555.);
         if R(1561.)=-4 then SEIS(1)=0;                      else SEIS(1)=R(1561.);
         if R(1589.)=-4 & R(1591.)=–4 then UIS(1)=0;
        else if R(1589.)>0 & R(1591.)>0 then UIS(1)=R(1589.) * R(1591.);
        else UIS(1)=R(1591.);
         if R(1594.)=-4 then ALIM(1)=0;                      else ALIM(1)=R(1594.);
        CHSP(1)=0;
         if R(1608.)=-4 then AFDC(1)=0;
        else do; B=0;
          do J=1596. to 1607.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(1608.)<0 then AFDC(1)=R(1608.);
          else if  B<=0 then AFDC(1)=–3;                     else AFDC(1)=R(1608.) * B;
        end;
         if R(1623.)=-4 then FOOD(1)=0;
        else do; B=0;
          do J=1610. to 1621.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if B<=0 then FOOD(1)=-3;
           else if R(1622.)>=0 & R(1623.)>0 then FOOD(1)=(R(1623.)-R(1622.)) * B; else FOOD(1)=-3;
        end;
        WELF(1)=0;
         if R(1640.)=-4 then PUBLIC(1)=0;
        else do; B=0;
          do J=1628. to 1639.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(1640.)<0 then PUBLIC(1)=R(1640.);
          else if B<=0 then PUBLIC(1)=-3;                    else PUBLIC(1)=R(1640.) * B;
        end;
         if R(1645.)=-4 then ED(1)=0;                else ED(1)=R(1645.);
         if R(1646.)=-4 then EDSS(1)=0;                      else EDSS(1)=R(1646.);
         VET(1)=0;
         if R(1649.)=-4 then GIFT(1)=0;                      else GIFT(1)=R(1649.);
         if R(1651.)=-4 then OTHER(1)=0;                     else OTHER(1)=R(1651.);
         RELWEL(1)=0;
        if R(1664.)=-4 then RELREG(1)=0;                    else RELREG(1)=R(1664.);

/*1979 FAMILY POVERTY STATUS, 1979 FAMILY POVERTY LEVEL, & 1979 FAMILY POVERTY
FLAG*/
 DCL PUBID PIC '99999';
   DCL S79          fixed DEC(9);        FAMILY79    fixed DEC(9);         FARM79          fixed
DEC(9);
   SCREENSZ         fixed DEC(9);        FLAG        fixed DEC(9);         INCOME79        float
DEC(6);
    CAL_INCOME      float DEC(6);        LEVEL79     float DEC(6);         STATE79         float
DEC(6);
    CPS79           float DEC(6);        R79         float DEC(6);         TABLE(3,2,15)   float
DEC(6);
    R(1.10)         float DEC(6);        R(2179.)    float DEC(6);         R(1691.)        float
DEC(6);
    R(1697.)        float DEC(6);        R(1743.)    float DEC(6);         R(1916.10)      float
DEC(6);
    R(1921.)        float DEC(6);        R(1919.)    float DEC(6);         FAMSZ79         float
DEC(6);
   WPS              float DEC(6);        WPSS        float DEC(6);         SEI             float
DEC(6);
   SEIS             float DEC(6);        MIL         float DEC(6);         OTHER           float
DEC(6);
    RELINC          float DEC(6);        GIFT        float DEC(6);

 LEVEL79=-4;
 do I=1 to 3;
   do J=1 to 2;
      do K=1 to 15; TABLE(I,J,K)=-4; end;
   end;
 end;
 TABLE(1,1,1)=3400;                   TABLE(2,1,1)=4270;                         TABLE(3,1,1)=3930;
 TABLE(1,2,1)=2910;                   TABLE(2,2,1)=3650;                         TABLE(3,2,1)=3350;
do J=2 to 15;
   TABLE(1,1,J)=TABLE(1,1,J-1)+1100;                                 TABLE(2,1,J)=TABLE(2,1,J-
1)+1370;
   TABLE(3,1,J)=TABLE(3,1,J-1)+1260;                                 TABLE(1,2,J)=TABLE(1,2,J-
1)+930;
   TABLE(2,2,J)=TABLE(2,2,J-1)+1160;                                 TABLE(3,2,J)=TABLE(3,2,J-
1)+1070;
 end;

 FARM79=1;
 if R(1919.)>4 then FARM79=2;
 if R(1743.)>0 then STATE79=MOD(R(1743.),100);       else STATE79=R(1743.);
 S79=1;
 if STATE79=2 then S79=2;                             else if STATE79=15 then S79=3;

/*To create family size (FAMSZ79) search thru the household enumeration increment family size, if the
relationship to the youth is a relative. Do not increase family size if the code is <0 or (ge 33 & le 36) or =45 or
=46 or *ge 50 & le 54)*/

 if FAMSZ79=0 then FAMILY79=1;
 else FAMILY79=FAMSZ79;

 INCOME79=R(2179.);
 R79=0;
 if R(1691.)>=0 then R79=R79+R(1691.);
 if R(1697.)>=0 then R79=R79+R(1697.);

 if FAMILY79>0 then do;
    LEVEL79=TABLE(S79,FARM79,FAMILY79);
    if INCOME79>LEVEL79 then CPS79=0;
    else if INCOME79>= 0 & INCOME79<=LEVEL79 then CPS79=1;
    else if R79>TABLE(S79,FARM79,FAMILY79) then CPS79=0;
    else CPS79=-3;
 end;
 else do;
    if INCOME79>TABLE(S79,FARM79,15) then CPS79=0;
    else if INCOME79>=0 & INCOME79<=TABLE(S79,FARM79,1) then CPS79=1;
    else CPS79=-3;
 end;
 FLAG=0;
 CAL_INCOME=0;
 if CPS79=-3 & INCOME79<0 then do;
    if WPS>=0 then CAL_INCOME=CAL_INCOME + WPS;
    if WPSS>=0 then CAL_INCOME=CAL_INCOME + WPSS;
    if SEI>=0 then CAL_INCOME=CAL_INCOME + SEI;
    if SEIS>=0 then CAL_INCOME=CAL_INCOME + SEIS;
    if MIL>=0 then CAL_INCOME=CAL_INCOME + MIL;
    if OTHER>=0 then CAL_INCOME=CAL_INCOME + OTHER;
    if RELINC>=0 then CAL_INCOME=CAL_INCOME + RELINC;
    if GIFT>=0 then CAL_INCOME=CAL_INCOME + GIFT;
    if CAL_INCOME>0 then do;
       if CAL_INCOME>LEVEL79 then do;
          CPS79=0;
          FLAG=1;
       end;
    end;
 end;
 if CPS79=-3 & R(1916.10)>0 & R(1921.)>0 then do;
    SCREENSZ=R(1921.);
    if SCREENSZ=-4 then FAMILY79=1;
    else FAMILY79=SCREENSZ;
    LEVEL79=TABLE(S79,FARM79,FAMILY79);
    if R(1916.10)>LEVEL79 then do;
       CPS79=0;
       FLAG=2;
    end;
    else if R(1916.10)<=LEVEL79 then do;
       CPS79=1;
       FLAG=2;
    end;
 end;

CPS79=R(2179.10);
LEVEL79=R(2179.20);
FLAG79=R(2179.30);
```

```
/*  1980 VARIABLES  */
        if R(4052.)<=0 then INC(2)=-5;                    else INC(2)=R(4045.10);
        if R(3120.)=-4 then MIL(2)=0;                     else MIL(2)=R(3120.);
        if R(3123.)=-4 then WPS(2)=0;                     else WPS(2)=R(3123.);
        if R(3126.)=-4 then SEI(2)=0;                     else SEI(2)=R(3126.);
        if R(3146.)=-4 & R(3132.)=–4 then UI(2)=0;
        else if R(3146.)>0 & R(3132.)>0 then UI(2)=R(3146.) * R(3132.);
        else UI(2)=R(3132.);
        if R(3122.)=-4 then MILS(2)=0;                    else MILS(2)=R(3122.);
        if R(3127.10)=-4 then WPSS(2)=0;                            else WPSS(2)=R(3127.10);
        if R(3130.)=-4 then SEIS(2)=0;                    else SEIS(2)=R(3130.);
        if R(3160.)=-4 & R(3161.)=–4 then UIS(2)=0;
        else if R(3160.)>0 & R(3161.)>0 then UIS(2)=R(3160.) * R(3161.);
        else UIS(2)=R(3161.);
        if R(3164.)=-4 then ALIM(2)=0;                    else ALIM(2)=R(3164.);
        CHSP(2)=0;
        if R(3178.)=-4 then AFDC(2)=0;
        else do; B=0;
          do J=3166. to 3177.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(3178.)<0 then AFDC(2)=R(3178.);
          else if B<=0 then AFDC(2)=–3;                   else AFDC(2)=R(3178.) * B;
        end;
        if R(3192.)=-4 then FOOD(2)=0;
        else do; B=0;
          do J=3180. to 3191.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(3192.)=–4 then FOOD(2)=R(3192.);
          else if B<=0 then FOOD(2)=-3;                   else FOOD(2)=R(3192.) * B;
        end;
        if R(3206.)=-4 then WELF(2)=0;
        else do; B=0;
          do J=3194. to 3205.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(3206.)<0 then WELF(2)=R(3206.);
          else if B<=0 then WELF(2)=-3;                   else WELF(2)=R(3206.) * B;
        end;
        if R(3220.)=-4 then PUBLIC(2)=0;
        else do; B=0;
          do J=3208. to 3219.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(3220.)<0 then PUBLIC(2)=R(3220.);
          else if B<=0 then PUBLIC(2)=-3;                           else PUBLIC(2)=R(3220.) * B;
        end;
        if R(3225.)=-4 then ED(2)=0;                      else ED(2)=R(3225.);
        if R(3226.)=-4 then EDSS(2)=0;                    else EDSS(2)=R(3226.);
        if R(3228.)=-4 then VET(2)=0;                     else VET(2)=R(3228.);
        if R(3233.)=-4 then GIFT(2)=0;                    else GIFT(2)=R(3233.);
        if R(3235.)=-4 then OTHER(2)=0;                   else OTHER(2)=R(3235.);
        if R(3244.)=-4 then RELWEL(2)=0;                  else RELWEL(2)=R(3244.);
        if R(3251.10)=-4 then RELREG(2)=0;                else RELREG(2)=R(3251.10);
        if R(4047.)=-4 then POVERTY(2)=0;                 else POVERTY(2)=R(4047.);
        FAMILY(2)=0;
        do J=3939. to 4023. BY 6;
           if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
             else FAMILY(2)=FAMILY(2)+1;
        end;
```

```
/*  1981 VARIABLES  */
        if R(6146.)<=0 then INC(3)=-5;                          else INC(3)=R(6138.10);
        YOUTH(3)=0;            TABLE(3,1,1,1)=4320; TABLE(3,2,1,1)=5400; TABLE(3,3,1,1)=4970;
                               TABLE(3,1,2,1)=3690; TABLE(3,2,2,1)=4660; TABLE(3,3,2,1)=4270;
        do J=2 to 15;          TABLE(3,1,1,J)=TABLE(3,1,1,J-1) + 1380; TABLE(3,2,1,J)=TABLE(3,2,1,J-
1) + 1730;
                               TABLE(3,3,1,J)=TABLE(3,3,1,J-1) + 1590; TABLE(3,1,2,J)=TABLE(3,1,2,J-
1) + 1170;
                               TABLE(3,2,2,J)=TABLE(3,2,2,J-1) + 1450; TABLE(3,3,2,J)=TABLE(3,3,2,J-
1) + 1340;
        end;
        FARM(3)=1;                                              if R(6125.)>4 then FARM(3)=2;  S(3)=1;
        if R(6028.)>0 then do; if MOD(R(6028.),100)=2 then S(3)=2; if MOD(R(6028.),100)=15 then S(3)=3;
end;
        LEVEL(3)=-4; if R(4825.)=-4 then MIL(3)=0;              else MIL(3)=R(4825.);
        if R(4826.)=-4 then WPS(3)=0;                           else WPS(3)=R(4826.);
        if R(4832.)=-4 then SEI(3)=0;                           else SEI(3)=R(4832.);
        if R(4849.)=-4 & R(4850.)=–4 then UI(3)=0;
        else if R(4849.)>0 & R(4850.)>0 then UI(3)=R(4849.) * R(4850.); else UI(3)=R(4850.);
        if R(4828.)=-4 then MILS(3)=0;                          else MILS(3)=R(4828.);
        if R(4829.10)=-4 then WPSS(3)=0;                        else WPSS(3)=R(4829.10);
        if R(4835.)=-4 then SEIS(3)=0;                          else SEIS(3)=R(4835.);
        if R(4865.)=-4 & R(4866.)=–4 then UIS(3)=0;
        else if R(4865.)>0 & R(4866.)>0 then UIS(3)=R(4865.) * R(4866.); else UIS(3)=R(4866.);
        if R(4869.)=-4 then ALIM(3)=0;                          else ALIM(3)=R(4869.);
        CHSP(3)=0;                                              if R(4883.)=-4 then AFDC(3)=0;
        else do; B=0;
          do J=4871. to 4882.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(4883.)<0 then AFDC(3)=R(4883.);
          else if B<=0 then AFDC(3)=–3;                         else AFDC(3)=R(4883.) * B;
        end;
        if R(4897.)=-4 then FOOD(3)=0;
        else do; B=0;
          do J=4885. to 4896.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(4897.)<0 then FOOD(3)=R(4897.);
          else if B<=0 then FOOD(3)=-3;                         else FOOD(3)=R(4897.) * B;
        end;
        if R(4911.)=-4 then WELF(3)=0;
        else do; B=0;
          do J=4899. to 4910.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(4911.)<0 then WELF(3)=R(4911.);
          else if B<=0 then WELF(3)=-3;                         else WELF(3)=R(4911.) * B;
        end;
        if R(4925.)=-4 then PUBLIC(3)=0;
        else do; B=0;
          do J=4913. to 4924.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
          if R(4925.)<0 then PUBLIC(3)=R(4925.);
          else if B<=0 then PUBLIC(3)=-3;                                 else PUBLIC(3)=R(4925.) * B;
        end;
        if R(4930.)=-4 then ED(3)=0;                            else ED(3)=R(4930.);
        if R(4931.)=-4 then EDSS(3)=0;                          else EDSS(3)=R(4931.);
        if R(4933.)=-4 then VET(3)=0;                           else VET(3)=R(4933.);
        if R(4938.)=-4 then GIFT(3)=0;                          else GIFT(3)=R(4938.);
        if R(4940.)=-4 then OTHER(3)=0;                         else OTHER(3)=R(4940.);
        if R(4949.)=-4 then RELWEL(3)=0;                        else RELWEL(3)=R(4949.);
```

```
 if R(4956.10)=-4 then RELREG(3)=0;                    else RELREG(3)=R(4956.10);
 if R(6140.)=-4 then POVERTY(3)=0;                     else POVERTY(3)=R(6140.);
FAMILY(3)=0;                                           do J=6033. to 6117. BY 6;
   if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
  else FAMILY(3)=FAMILY(3)+1;                                     end;
```

```
/*  1982 VARIABLES  */
      if R(8967.)<=0 then INC(4)=-5;                      else INC(4)=R(8304.);
      YOUTH(4)=0; if R(7986.)>=0 then YOUTH(4)=R(7986.);
      if R(7988.)>=0 then YOUTH(4)=Y(4)+R(7988.);
      if R(7992.)>=0 then YOUTH(4)=Y(4)+R(7992.);
      TABLE(4,1,1,1)=4760; TABLE(4,2,1,1)=5990; TABLE(4,3,1,1)=5500;
      TABLE(4,1,2,1)=4080; TABLE(4,2,2,1)=5110; TABLE(4,3,2,1)=4700;
      do J=2 to 15;                TABLE(4,1,1,J)=TABLE(4,1,1,J-1)+1530; TABLE(4,2,1,J)=TABLE(4,2,1,J-
1)+1900;
                             TABLE(4,3,1,J)=TABLE(4,3,1,J-1)+1750; TABLE(4,1,2,J)=TABLE(4,1,2,J-
1)+1290;
                             TABLE(4,2,2,J)=TABLE(4,2,2,J-1)+1610; TABLE(4,3,2,J)=TABLE(4,3,2,J-
1)+1480;
      end;
      FARM(4)=1;                                          if R(8291.)>4 then FARM(4)=2; S(4)=1;
      if R(8169.)=2 then S(4)=2;                          else if R(8169.)=15 then S(4)=3;
      LEVEL(4)=-4; if R(7820.)=-4 then MIL(4)=0;          else MIL(4)=R(7820.);
      if R(7821.)=-4 then WPS(4)=0;                       else WPS(4)=R(7821.);
      if R(7824.)=-4 then SEI(4)=0;                       else SEI(4)=R(7824.);
      if R(7838.)=-4 & R(7839.)=-4 then UI(4)=0;
      else if R(7838.)>0 & R(7839.)>0 then UI(4)=R(7838.) * R(7839.); else UI(4)=R(7839.);
      if R(7842.)=-4 then MILS(4)=0;                      else MILS(4)=R(7842.);
      if R(7843.)=-4 then WPSS(4)=0;                      else WPSS(4)=R(7843.);
      if R(7846.)=-4 then SEIS(4)=0;                      else SEIS(4)=R(7846.);
      if R(7860.)=-4 & R(7861.)=-4 then UIS(4)=0;
      else if R(7860.)>0 & R(7861.)>0 then UIS(4)=R(760.) * R(7861.); else UIS(4)=R(7861.);
      if R(7864.)=-4 then ALIM(4)=0;                      else ALIM(4)=R(7864.);
      if R(7869.)=-4 then CHSP(4)=0;                      else CHSP(4)=R(7869.);
      if R(7885.)=-4 then AFDC(4)=0;                      else do; B=0;
        do J=7873. to 7884.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(7885.)<0 then AFDC(4)=R(7885.);
        else if B<=0 then AFDC(4)=-3;                     else AFDC(4)=R(7885.) * B;
      end;
      if R(7899.)=-4 then FOOD(4)=0;
      else do; B=0; do J=7887. to 7898.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then
B=na; end;
        if R(7899.)<0 then FOOD(4)=R(7899.);
        else if B<=0 then FOOD(4)=-3;                     else FOOD(4)=R(7899.) * B;
      end;
      if R(7913.)=-4 then WELF(4)=0;
      else do; B=0; do J=7901. to 7912.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then
B=na; end;
        if R(7913.)<0 then WELF(4)=R(7913.);
        else if B<=0 then WELF(4)=-3;                     else WELF(4)=R(7913.) * B;
      end;
      if R(7927.)=-4 then PUBLIC(4)=0;
      else do; do J=7915. to 7926.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(7927.)<0 then PUBLIC(4)=R(7927.);
        else if B<=0 then PUBLIC(4)=-3;                                 else PUBLIC(4)=R(7927.) * B;
      end;
      if R(7932.)=-4 then ED(4)=0;                        else ED(4)=R(7932.);
      if R(7933.)=-4 then EDSS(4)=0;                      else EDSS(4)=R(7933.);
      if R(7935.)=-4 then VET(4)=0;                       else VET(4)=R(7935.);
      if R(7939.)=-4 then GIFT(4)=0;                      else GIFT(4)=R(7939.);
      if R(7941.)=-4 then OTHER(4)=0;                     else OTHER(4)=R(7941.);
```

```
 if R(7949.)=-4 then RELWEL(4)=0;          else RELWEL(4)=R(7949.);
 if R(7956.)=-4 then RELREG(4)=0;          else RELREG(4)=R(7956.);
 if R(8306.)=-4 then POVERTY(4)=0;         else POVERTY(4)=R(8306.);
FAMILY(4)=0;                               do J=8177. to 8275. BY 7;
   if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
      else FAMILY(4)=FAMILY(4)+1;
end;
```

```
/* 1983 VARIABLES */
      if R(11444.)<=0 then INC(5)=-5;                      else INC(5)=R(10777.);
      YOUTH(5)=0;              TABLE(5,1,1,1)=4910; TABLE(5,2,1,1)=6160; TABLE(5,3,1,1)=5670;
                               TABLE(5,1,2,1)=4200; TABLE(5,2,2,1)=5290; TABLE(5,3,2,1)=4850;
      do J=2 to 15;            TABLE(5,1,1,J)=TABLE(5,1,1,J-1)+1620; TABLE(5,2,1,J)=TABLE(5,2,1,J-
1)+2020;
                               TABLE(5,3,1,J)=TABLE(5,3,1,J-1)+1860; TABLE(5,1,2,J)=TABLE(5,1,2,J-
1)+1370;
                               TABLE(5,2,2,J)=TABLE(5,2,2,J-1)+1700; TABLE(5,3,2,J)=TABLE(5,3,2,J-
1)+1570;
      end;
      FARM(5)=1;                                           if R(10764.)>4 then FARM(5)=2;
      S(5)=1; if R(10548.)=2 then S(5)=2;                  else if R(10548.)=15 then S(5)=3;
      LEVEL(5)=-4; if R(10239.)=-4 then MIL(5)=0;          else MIL(5)=R(10239.);
      if R(10240.)=-4 then WPS(5)=0;                       else WPS(5)=R(10240.);
      if R(10243.)=-4 then SEI(5)=0;                       else SEI(5)=R(10243.);
      if R(10257.)=-4 & R(10258.)=-4 then UI(5)=0;
      else if R(10257.)>0 & R(10258.)>0 then UI(5)=R(10257.) * R(10258.);
      else if R(10258.)<0 then UI(5)=R(10258.);            else UI(5)=-3;
      if R(10261.)=-4 then MILS(5)=0;                      else MILS(5)=R(10261.);
      if R(10262.)=-4 then WPSS(5)=0;                      else WPSS(5)=R(10262.);
      if R(10265.)=-4 then SEIS(5)=0;                      else SEIS(5)=R(10265.);
      if R(10279.)=-4 & R(10280.)=-4 then UIS(5)=0;
      else if R(10279.)>0 & R(10280.)>0 then UIS(5)=R(10279.) * R(10280.);
      else if R(10280.)<0 then UIS(5)=R(10280.);           else UIS(5)=-3;
      if R(10283.)=-4 then ALIM(5)=0;                      else ALIM(5)=R(10283.);
      if R(10288.)=-4 then CHSP(5)=0;                      else CHSP(5)=R(10288.);
      if R(10304.)=-4 then AFDC(5)=0;
      else do; B=0;
        do J=10292. to 10303.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(10304.)<0 then AFDC(5)=R(10304.);
        else if B<=0 then AFDC(5)=-3;                      else AFDC(5)=R(10304.) * B;
      end;
      if R(10318.)=-4 then FOOD(5)=0;
      else do; B=0;
        do J=10306. to 10317.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(10318.)<0 then FOOD(5)=R(10318.);
        else if B<=0 then FOOD(5)=-3;                      else FOOD(5)=R(10318.) * B;
      end;
      if R(10332.)=-4 then WELF(5)=0;
      else do; B=0;
        do J=10320. to 10331.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(10332.)<0 then WELF(5)=R(10332.);
        else if B<=0 then WELF(5)=-3;                      else WELF(5)=R(10332.) * B;
      end;
      if R(10346.)=-4 then PUBLIC(5)=0;
      else do; B=0;
        do J=10334. to 10345.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(10346.)<0 then PUBLIC(5)=R(10346.);
        else if B<=0 then PUBLIC(5)=-3;                              else PUBLIC(5)=R(10346.) * B;
      end;
      if R(10351.)=-4 then ED(5)=0;                        else ED(5)=R(10351.);
      if R(10352.)=-4 then EDSS(5)=0;                      else EDSS(5)=R(10352.);
      if R(10354.)=-4 then VET(5)=0;                       else VET(5)=R(10354.);
      if R(10358.)=-4 then GIFT(5)=0;                      else GIFT(5)=R(10358.);
```

```
     if R(10360.)=–4 then OTHER(5)=0;                    else OTHER(5)=R(10360.);
     if R(10368.)=–4 then RELWEL(5)=0;                   else RELWEL(5)=R(10368.);
     if R(10375.)=–4 then RELREG(5)=0;                   else RELREG(5)=R(10375.);
     if R(10779.)=–4 then POVERTY(5)=0;                  else POVERTY(5)=R(10779.);
    FAMILY(5)=0;
    do J=10556. to 10654. BY 7;
       if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=54) then A=1;
         else FAMILY(5)=FAMILY(5)+1;
    end;
    do I=1 to 5;
      MN=0; FAMINC(I)=0;
      do K=MIL(I), MILS(I), WPS(I), WPSS(I), SEI(I), SEIS(I), UI(I), UIS(I), ALIM(I), CHSP(I),
        AFDC(I), FOOD(I), WELF(I), PUBLIC(I), ED(I), EDSS(I), VET(I), GIFT(I), OTHER(I),
        RELWEL(I), RELREG(I);
         if K<0 then MN(I,ABS(K))=MN(I,ABS(K))+1;                    else FAMINC(I)=FAMINC(I)
+ K;
      end;
      SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
      INCOME(I)=-3;
      if INC(I)=–5 then do; INCOME(I),CPS(I)=–5; end;
      else INC(I)^=-4 then INCOM E(I)=INC(I);
      else do;
         if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
      end;
      if I>2 then do;
         if FAMILY(I)>0 then do;
           LEVEL(I)=TABLE(I,S(I),FARM(I),FAMILY(I));
           if INC(I)>LEVEL(I) ! FAMINC(I)>LEVEL(I) then CPS(I)=0;
           else if INC(I)>=0 ! (SUMN(I)=0 & INC(I)=–4) then CPS(I)=1;
           else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;                 else CPS(I)=–3;
        end;
        else do;
           if INC(I)>TABLE(I,S(I),FARM(I),15) ! FAMINC(I)>TABLE(I,S(I),FARM(I),15) then CPS(I)=0;
           else if (INC(I)>=0 & INC(I)<=TABLE(I,S(I),FARM(I),1)) ! (SUMN(I)=0 & INC(I)=–4 &
                FAMINC(I)<= TABLE(I,S(I),FARM(I),1)) then CPS(I)=1;
           else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;                 else CPS(I)=–3;
        end;
         if CPS(I)=-3 & FAMILY(I)>0 & YOUTH(I)> TABLE(I,S(I),FARM(I),FAMILY(I)) then
CPS(I)=0;
      end;
    if I=2 then do;
      if INCOME(2)=-5 then CPS(2)=–5;
      else do;
         if FAMILY(2)=1 & INCOME(2)>3778 then CPS(2)=0;
        else if FAMILY(2)=2 & INCOME(2)>4878 then CPS(2)=0;
        else if FAMILY(2)=3 & INCOME(2)>5784 then CPS(2)=0;
        else if FAMILY(2)=4 & INCOME(2)>7412 then CPS(2)=0;
        else if FAMILY(2)=5 & INCOME(2)>8775 then CPS(2)=0;
        else if FAMILY(2)=6 & INCOME(2)>9914 then CPS(2)=0;
        else if FAMILY(2)>6 & INCOME(2)>12280 then CPS(2)=0;
        else if INCOME(2)>12280 then CPS(2)=0;
        else do;
          if INC(2)>=0 & FAMILY(2)>0 then CPS(2)=1;
          else do;
             if SUMN(2)=0 & (R(4027.)=3 ! R(4029.)>1) & FAMILY(2)>0 then CPS(2)=1;
```

```
              else if POVERTY(2)>0 then CPS(2)=POVERTY(2)-1;            else CPS(2)=-3;
             end;
            end;
           end;
          end;

      R(2179.)=INCOME(1);             R(4060.)=INCOME(2);
R(4061.)=POVERTY(2);
      R(6184.10)=INCOME(3);           R(6185.)=POVERTY(3);
R(8986.)=INCOME(4);
      R(8987.)=POVERTY(4);            R(11445.10)=INCOME(5);
R(11448.)=POVERTY(5);
```

```
  /* 1984–86 */

  DCL          1 FAMILY_INCOME (6:8),          5 AFDC,              5 ALIM,
               5 CHSP,                         5 CPS,               5 ED,
               5 EDSS,                         5 FAMILY,            5 FAMINC,
               5 FARM,                         5 FOOD,              5 GIFT,
               5 INC,                          5 INCOME,            5 LEVEL,
               5 MIL,                          5 MILS,              5 MN (3),
               5 OTHER,                        5 POVERTY,           5 PUBLIC,
               5 RELREG,                       5 RELWEL,            5 S,
               5 SEI,                          5 SEIS,                          5
SUMN,
               5 TABLE(3,15),                  5 UI,                5 UIS,
               5 VET,                          5 WELF,              5 WPS,
               5 WPSS;

/* 1984 VARIABLES */

     FAMILY_INCOME=-4;
     if R(15196.)=0 then INC(6)=-5;                  else INC(6)=R(14534.);
     TABLE(6,1,1)=5010; TABLE(6,2,1)=6280; TABLE(6,3,1)=6790;
     do J=2 to 15;         TABLE(6,1,J)=TABLE(6,1,J-1)+1740;
                           TABLE(6,2,J)=TABLE(6,2,J-1)+2170;     TABLE(6,3,J)=TABLE(6,3,J-1)+1990;
     end;
     FARM(6)=1;                                      if R(14521.)>4 then FARM(6)=2;
     S(6)=1; if STATE84=2 then S(6)=2;               else if STATE84=15 then S(6)=3;
     LEVEL(6)=-4; if R(14106.)=-4 then MIL(6)=0;     else MIL(6)=R(14106.);
     if R(14107.)=–4 then WPS(6)=0;                  else WPS(6)=R(14107.);
     if R(14110.)=–4 then SEI(6)=0;                  else SEI(6)=R(14110.);
     if R(14124.)=–4 & R(14125.)=-4 then UI(6)=0;
     else if R(14124.)>0 & R(14125.)>0 then UI(6)=R(14124.) * R(14125.);
     else if R(14125.)<0 then UI(6)=R(14125.);       else UI(6)=–3;
     if R(14128.)=–4 then MILS(6)=0;                 else MILS(6)=R(14128.);
     if R(14129.)=–4 then WPSS(6)=0;                 else WPSS(6)=R(14129.);
     if R(14132.)=–4 then SEIS(6)=0;                 else SEIS(6)=R(14132.);
     if R(14146.)=–4 & R(14147.)=-4 then UIS(6)=0;
     else if R(14146.)>0 & R(14147.)>0 then UIS(6)=R(14146.) * R(14147.);
     else if R(14147.)<0 then UIS(6)=R(14147.);      else UIS(6)=-3;
     if R(14150.)=–4 then ALIM(6)=0;                 else ALIM(6)=R(14150.);
     if R(14155.)=–4 then CHSP(6)=0;                 else CHSP(6)=R(14155.);
     if R(14171.)=–4 then AFDC(6)=0;
     else do; B=0;
       do J=14159. to 14170.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
       if R(14171.)<0 then AFDC(6)=R(14171.);
       else if B<=0 then AFDC(6)=–3;                    else AFDC(6)=R(14171.) * B;
     end;
     if R(14185.)=–4 then FOOD(6)=0;
     else do; B=0;
       do J=14173. to 14184.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
       if R(14185.)<0 then FOOD(6)=R(14185.);
       else if B<=0 then FOOD(6)=-3;                    else FOOD(6)=R(14185.) * B;
     end;
     if R(14199.)=–4 then WELF(6)=0;
     else do; B=0;
       do J=14187. to 14198.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
       if R(14199.)<0 then WELF(6)=R(14199.);
       else if B<=0 then WELF(6)=-3;                    else WELF(6)=R(14199.) * B;
```

```
        end;
        if R(14213.)=–4 then PUBLIC(6)=0;
        else do; B=0;
           do J=14201. to 14212.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
           if R(14213.)<0 then PUBLIC(6)=R(14213.);
           else if B<=0 then PUBLIC(6)=-3;                          else PUBLIC(6)=R(14213.) *
B;
        end;
        if R(14218.)=–4 then ED(6)=0;                 else ED(6)=R(14218.);
        if R(14219.)=–4 then EDSS(6)=0;               else EDSS(6)=R(14219.);
        if R(14221.)=–4 then VET(6)=0;                else VET(6)=R(14221.);
        if R(14225.)=–4 then GIFT(6)=0;               else GIFT(6)=R(14225.);
        if R(14227.)=–4 then OTHER(6)=0;              else OTHER(6)=R(14227.);
        if R(14235.)=–4 then RELWEL(6)=0;             else RELWEL(6)=R(14235.);
        if R(14242.)=–4 then RELREG(6)=0;             else RELREG(6)=R(14242.);
        if R(14536.)=–4 then POVERTY(6)=0;            else POVERTY(6)=R(14536.);
        FAMILY(6)=FAMSIZE;
```

/* To create family size (i.e. FAMSIZE) search thru the household enumeration increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

```
/* 1985 VARIABLES */
      if R(18902.)=0 then INC(7)=-5;                          else INC(7)=R(18006.);
      TABLE(7,1,1)=5180; TABLE(7,2,1)=6500; TABLE(7,3,1)=5970;
      do J=2 to 15;              TABLE(7,1,J)=TABLE(7,1,J-1)+1810;
                                 TABLE(7,2,J)=TABLE(7,2,J-1)+2260; TABLE(7,3,J)=TABLE(7,3,J-
1)+2080;
      end;
      FARM(7)=1;                                               if R(17993.)>4 then FARM(7)=2;
      S(7)=1; if STATE85=2 then S(7)=2;                       else if STATE85=15 then S(7)=3;
      LEVEL(7)=-4; if R(17784.)=-4 then MIL(7)=0;             else MIL(7)=R(17784.);
      if R(17785.)=–4 then WPS(7)=0;                          else WPS(7)=R(17785.);
      if R(17788.)=–4 then SEI(7)=0;                          else SEI(7)=R(17788.);
      if R(17802.)=–4 & R(17803.)=-4 then UI(7)=0;
      else if R(17802.)>0 & R(17803.)>0 then UI(7)=R(17802.) * R(17803.);
      else if R(17803.)<0 then UI(7)=R(17803.);               else UI(7)=–3;
      if R(17806.)=–4 then MILS(7)=0;                         else MILS(7)=R(17806.);
      if R(17807.)=–4 then WPSS(7)=0;                         else WPSS(7)=R(17807.);
      if R(17810.)=–4 then SEIS(7)=0;                         else SEIS(7)=R(17810.);
      if R(17824.)=–4 & R(17825.)=-4 then UIS(7)=0;
      else if R(17824.)>0 & R(17825.)>0 then UIS(7)=R(17824.) * R(17825.);
      else if R(17825.)<0 then UIS(7)=R(17825.);              else UIS(7)=-3;
      if R(17828.)=–4 then ALIM(7)=0;                         else ALIM(7)=R(17828.);
      if R(17833.)=–4 then CHSP(7)=0;                         else CHSP(7)=R(17833.);
      if R(17849.)=–4 then AFDC(7)=0;
      else do; B=0;
        do J=17837. to 17848.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(17849.)<0 then AFDC(7)=R(17849.);
        else if B<=0 then AFDC(7)=–3;                          else AFDC(7)=R(17849.) * B;
      end;
      if R(17863.)=–4 then FOOD(7)=0;
      else do; B=0;
        do J=17851. to 17862.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(17863.)<0 then FOOD(7)=R(17863.);
        else if B<=0 then FOOD(7)=-3;                          else FOOD(7)=R(17863.) * B;
      end;
      if R(17877.)=–4 then PUBLIC(7)=0;
      else do; B=0;
        do J=17865. to 17876.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
        if R(17877.)<0 then PUBLIC(7)=R(17877.);
        else if B<=0 then PUBLIC(7)=-3;                                    else PUBLIC(7)=R(17877.) *
B;
      end;
      if R(17882.)=–4 then ED(7)=0;                           else ED(7)=R(17882.);
      if R(17883.)=–4 then EDSS(7)=0;                         else EDSS(7)=R(17883.);
      if R(17885.)=–4 then VET(7)=0;                          else VET(7)=R(17885.);
      if R(17887.)=–4 then OTHER(7)=0;                        else OTHER(7)=R(17887.);
      if R(17895.)=–4 then RELWEL(7)=0;                       else RELWEL(7)=R(17895.);
      if R(17897.)=–4 then RELREG(7)=0;                       else RELREG(7)=R(17897.);
      if R(18008.)=–4 then POVERTY(7)=0;                      else POVERTY(7)=R(18008.);
      FAMILY(7)=FAMSZ85;

/* To create family size (i.e. FAMSZ85) search thru the household enumeration. Increment family size, if the
relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33 & <=36) or =45
or =46 or (>=50 & <=54) */
```

/* 1986 VARIABLES */

```
     if R(22573.)=0 then INC(8)=-5;                    else INC(8)=R(21622.);
    TABLE(8,1,1)=5430; TABLE(8,2,1)=6790; TABLE(8,3,1)=6250;
    do J=2 to 15;           TABLE(8,1,J)=TABLE(8,1,J-1)+1860;
                            TABLE(8,2,J)=TABLE(8,2,J-1)+2330; TABLE(8,3,J)=TABLE(8,3,J-
1)+2140;
    end;
    FARM(8)=1;                                          if R(21609.)>4 then FARM(8)=2;
     S(8)=1; if STATE86=2 then S(8)=2;                 else if STATE86=15 then S(8)=3;
     LEVEL(8)=-4; if R(21415.)=-4 then MIL(8)=0;        else MIL(8)=R(21415.);
     if R(21416.)=-4 then WPS(8)=0;                     else WPS(8)=R(21416.);
     if R(21419.)=-4 then SEI(8)=0;                     else SEI(8)=R(21419.);
     if R(21433.)=-4 & R(21434.)=-4 then UI(8)=0;
     else if R(21433.)>0 & R(21434.)>0 then UI(8)=R(21433.) * R(21434.);
    else if R(21434.)<0 then UI(8)=R(21434.);           else UI(8)=-3;
     if R(21437.)=-4 then MILS(8)=0;                    else MILS(8)=R(21437.);
     if R(21438.)=-4 then WPSS(8)=0;                    else WPSS(8)=R(21438.);
     if R(21441.)=-4 then SEIS(8)=0;                    else SEIS(8)=R(21441.);
     if R(21455.)=-4 & R(21456.)=-4 then UIS(8)=0;
     else if R(21455.)>0 & R(21456.)>0 then UIS(8)=R(21455.) * R(21456.);
    else if R(21456.)<0 then UIS(8)=R(21456.);          else UIS(8)=-3;
     if R(21459.)=-4 then ALIM(8)=0;                    else ALIM(8)=R(21459.);
     if R(21464.)=-4 then CHSP(8)=0;                    else CHSP(8)=R(21464.);
     if R(21480.)=-4 then AFDC(8)=0;
    else do; B=0;
      do J=21468. to 21479.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
     if R(21480.)<0 then AFDC(8)=R(21480.);
      else if B<=0 then AFDC(8)=-3;                        else AFDC(8)=R(21480.) * B;
    end;
     if R(21494.)=-4 then FOOD(8)=0;
    else do; B=0;
      do J=21482. to 21493.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
     if R(21494.)<0 then FOOD(8)=R(21494.);
      else if B<=0 then FOOD(8)=-3;                        else FOOD(8)=R(21494.) * B;
    end;
     if R(21508.)=-4 then PUBLIC(8)=0;
    else do; B=0;
      do J=21496. to 21507.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
     if R(21508.)<0 then PUBLIC(8)=R(21508.);
      else if B<=0 then PUBLIC(8)=-3;                                  else PUBLIC(8)=R(21508.) *
B;
    end;
     if R(21513.)=-4 then ED(8)=0;                      else ED(8)=R(21513.);
     if R(21514.)=-4 then EDSS(8)=0;                    else EDSS(8)=R(21514.);
     if R(21516.)=-4 then VET(8)=0;                     else VET(8)=R(21516.);
     if R(21518.)=-4 then OTHER(8)=0;                   else OTHER(8)=R(21518.);
     if R(21526.)=-4 then RELWEL(8)=0;                  else RELWEL(8)=R(21526.);
     if R(21528.)=-4 then RELREG(8)=0;                  else RELREG(8)=R(21528.);
     if R(21624.)=-4 then POVERTY(8)=0;                 else POVERTY( 8)=R(21624.);

    FAMILY(8)=FAMSZ86;
    /* To create family size (i.e. FAMSZ86) search thru the household enumeration. Increment family size,
           if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33
           & <=36) or =45 or =46 or (>=50 & <=54) */
   do I=6 to 8;
```

```
      MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0;
      do K=MIL(I), MILS(I), WPS(I), WPSS(I), SEI(I), SEIS(I), UI(I), UIS(I), ALIM(I), CHSP(I),
        AFDC(I), FOOD(I), WELF(I), PUBLIC(I), ED(I), EDSS(I), VET(I), GIFT(I), OTHER(I),
        RELWEL(I), RELREG(I);
         if K>-4 then do;
           if K<0 then MN(I,ABS(K))=MN(I,ABS(K))+1;                    else
FAMINC(I)=FAMINC(I) + K;
         end;
      end;
      SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
      INCOME(I)=-3;
      if INC(I)=-5 then do; INCOME(I)=-5; CPS(I)=-5; end;
      else if INC(I)^=-4 then do;
        INCOME(I)=INC(I);
        if INC(I)<0 then do; MN(I,ABS(INC(I)))=1; SUMN(I)=1; end;
      end;
      else do;
        if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
      end;
      if FAMILY(I)>0 then do;
         LEVEL(I)=TABLE(I,S(I),FAMILY(I));
         if INC(I)>LEVEL(I) ! FAMINC(I)>LEVEL(I) then CPS(I)=0;
         else if INC(I)>=0 ! (SUMN(I)=0 & INC(I)=-4) then CPS(I)=1;
         else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;
         else CPS(I)=-3;
      end;
      else if CPS(I)^=-5 then do;
         if INC(I)>TABLE(I,S(I),15) ! FAMINC(I)>TABLE(I,S(I),15) then CPS(I)=0;
         else if (INC(I)>=0 & INC(I)<=TABLE(I,S(I),1)) ! (SUMN(I)=0 & INC(I)=-4 & FAMINC(I)<=
                 TABLE(I,S(I),1)) then CPS(I)=1;
         else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;
         else CPS(I)=-3;
      end;
    end;

INCOME(6)=R(15197.);
CPS(6)=R(15198.);

INCOME(7)=R(18904.);
CPS(7)=R(18905.);

INCOME(8)=R(22575.);
CPS(8)=R(22576.);
```

/* 1987 VARIABLES */

```
   DCL          1 FAMILY_INCOME (9),               5 AFDC,               5 ALIM,
                5 CHSP,                            5 CPS,                5 ED,
                5 EDSS,                            5 FAMILY,             5 FAMINC,
                5 FAMINC_CENSUS,                   5 FARM,               5 FOOD,
                5 GIFT,                            5 INC,                5 INCOME,
                5 INCOME_CENSUS,                   5 LEVEL,              5 MIL,
                5 MILS,                            5 MN (3),             5 MN_CENSUS
(3),
                5 OTHER,                           5 POVERTY,            5 PUBLIC,
                5 RELREG,                          5 RELWEL,             5 S,
                5 SEI,                             5 SEIS,               5 SUMN,
                5 SUMN_CENSUS,                     5 TABLE(3,15),        5 UI,
                5 UIS,                             5 VET,                5 WELF,
               5 WPS,                              5 WPSS;

      /* 1987 POVERTY INCOME GUIDELINES TABLE */
      TABLE(9,1,1)=5500; TABLE(9,2,1)=6860; TABLE(9,3,1)=6310;
      do J=2 to 15;             TABLE(9,1,J)=TABLE(9,1,J-1)+1900;
                                TABLE(9,2,J)=TABLE(9,2,J-1)+2380; TABLE(9,3,J)=TABLE(9,3,J-
1)+2190;
      end;
      S(9)=1; if STATE87=2 then S(9)=2;                else if STATE87=15 then S(9)=3;
      if R(23502.)=–4 then MIL(9)=0;                   else MIL(9)=R(23502.);
      if R(23503.)=–4 then WPS(9)=0;                   else WPS(9)=R(23503.);
      if R(23506.)=–4 then SEI(9)=0;                   else SEI(9)=R(23506.);
      if R(23520.)=–4 & R(23521.)=-4 then UI(9)=0;
      else if R(23520.)>0 & R(23521.)>0 then UI(9)=R(23520.) * R(23521.);
      else if R(23521.)<0 then UI(9)=R(23521.);        else UI(9)=–3;
      if R(23524.)=–4 then MILS(9)=0;                  else MILS(9)=R(23524.);
      if R(23525.)=–4 then WPSS(9)=0;                  else WPSS(9)=R(23525.);
      if R(23528.)=–4 then SEIS(9)=0;                  else SEIS(9)=R(23528.);
      if R(23542.)=–4 & R(23543.)=-4 then UIS(9)=0;
      else if R(23542.)>0 & R(23543.)>0 then UIS(9)=R(23542.) * R(23543.);
      else if R(23543.)<0 then UIS(9)=R(23543.);       else UIS(9)=-3;
      if R(23546.)=–4 then ALIM(9)=0;                  else ALIM(9)=R(23546.);
      if R(23551.)=-4 then CHSP(9)=0;                  else CHSP(9)=R(23551.);
      if R(23567.)=–4 then AFDC(9)=0;
      else do; B=0;
         if R(23555.)>0 & B^=na then B=B+1;            else if R(23555.)>-4 & R(23555.)<0 then B=na;
         if R(23556.)>0 & B^=na then B=B+1;            else if R(23556.)>-4 & R(23556.)<0 then B=na;
         if R(23557.)>0 & B^=na then B=B+1;            else if R(23557.)>-4 & R(23557.)<0 then B=na;
         if R(23558.)>0 & B^=na then B=B+1;            else if R(23558.)>-4 & R(23558.)<0 then B=na;
         if R(23559.)>0 & B^=na then B=B+1;            else if R(23559.)>-4 & R(23559.)<0 then B=na;
         if R(23560.)>0 & B^=na then B=B+1;            else if R(23560.)>-4 & R(23560.)<0 then B=na;
         if R(23561.)>0 & B^=na then B=B+1;            else if R(23561.)>-4 & R(23561.)<0 then B=na;
         if R(23562.)>0 & B^=na then B=B+1;            else if R(23562.)>-4 & R(23562.)<0 then B=na;
         if R(23563.)>0 & B^=na then B=B+1;            else if R(23563.)>-4 & R(23563.)<0 then B=na;
         if R(23564.)>0 & B^=na then B=B+1;            else if R(23564.)>-4 & R(23564.)<0 then B=na;
         if R(23565.)>0 & B^=na then B=B+1;            else if R(23565.)>-4 & R(23565.)<0 then B=na;
         if R(23566.)>0 & B^=na then B=B+1;            else if R(23566.)>-4 & R(23566.)<0 then B=na;
       if R(23567.)<0 then AFDC(9)=R(23567.); else if B<=0 then AFDC(9)=-3; else AFDC(9)=R(23567.)
* B;
      end;
      if R(23581.)=–4 then FOOD(9)=0;
```

```
        else do; B=0;
              if R(23569.)>0 & B^=na then B=B+1;                    else if R(23569.)>-4 & R(23569.)<0 then B=na;
              if R(23570.)>0 & B^=na then B=B+1;                    else if R(23570.)>-4 & R(23570.)<0 then B=na;
              if R(23571.)>0 & B^=na then B=B+1;                    else if R(23571.)>-4 & R(23571.)<0 then B=na;
              if R(23572.)>0 & B^=na then B=B+1;                    else if R(23572.)>-4 & R(23572.)<0 then B=na;
              if R(23573.)>0 & B^=na then B=B+1;                    else if R(23573.)>-4 & R(23573.)<0 then B=na;
              if R(23574.)>0 & B^=na then B=B+1;                    else if R(23574.)>-4 & R(23574.)<0 then B=na;
              if R(23575.)>0 & B^=na then B=B+1;                    else if R(23575.)>-4 & R(23575.)<0 then B=na;
              if R(23576.)>0 & B^=na then B=B+1;                    else if R(23576.)>-4 & R(23576.)<0 then B=na;
              if R(23577.)>0 & B^=na then B=B+1;                    else if R(23577.)>-4 & R(23577.)<0 then B=na;
              if R(23578.)>0 & B^=na then B=B+1;                    else if R(23578.)>-4 & R(23578.)<0 then B=na;
              if R(23579.)>0 & B^=na then B=B+1;                    else if R(23579.)>-4 & R(23579.)<0 then B=na;
              if R(23580.)>0 & B^=na then B=B+1;                    else if R(23580.)>-4 & R(23580.)<0 then B=na;
           if R(23581.)<0 then FOOD(9)=R(23581.); else if B<=0 then FOOD(9)=–3; else FOOD(9)=R(23581.)
* B;
        end;
         if R(23595.)=–4 then PUBLIC(9)=0;
        else do; B=0;
              if R(23583.)>0 & B^=na then B=B+1;                    else if R(23583.)>-4 & R(23583.)<0 then B=na;
              if R(23584.)>0 & B^=na then B=B+1;                    else if R(23584.)>-4 & R(23584.)<0 then B=na;
              if R(23585.)>0 & B^=na then B=B+1;                    else if R(23585.)>-4 & R(23585.)<0 then B=na;
              if R(23586.)>0 & B^=na then B=B+1;                    else if R(23586.)>-4 & R(23586.)<0 then B=na;
              if R(23587.)>0 & B^=na then B=B+1;                    else if R(23587.)>-4 & R(23587.)<0 then B=na;
              if R(23588.)>0 & B^=na then B=B+1;                    else if R(23588.)>-4 & R(23588.)<0 then B=na;
              if R(23589.)>0 & B^=na then B=B+1;                    else if R(23589.)>-4 & R(23589.)<0 then B=na;
              if R(23590.)>0 & B^=na then B=B+1;                    else if R(23590.)>-4 & R(23590.)<0 then B=na;
              if R(23591.)>0 & B^=na then B=B+1;                    else if R(23591.)>-4 & R(23591.)<0 then B=na;
              if R(23592.)>0 & B^=na then B=B+1;                    else if R(23592.)>-4 & R(23592.)<0 then B=na;
              if R(23593.)>0 & B^=na then B=B+1;                    else if R(23593.)>-4 & R(23593.)<0 then B=na;
              if R(23594.)>0 & B^=na then B=B+1;                    else if R(23594.)>-4 & R(23594.)<0 then B=na;
           if R(23595.)<0 then PUBLIC(9)=R(23595.); else if B<=0 then PUBLIC(9)=–3; else
PUBLIC(9)=R(23595.)*B;
        end;
         if R(23600.)=–4 then ED(9)=0;                              else ED(9)=R(23600.);
         if R(23601.)=–4 then EDSS(9)=0;                            else EDSS(9)=R(23601.);
         if R(23603.)=–4 then VET(9)=0;                             else VET(9)=R(23603.);
         if R(23605.)=–4 then OTHER(9)=0;                           else OTHER(9)=R(23605.);
         if VET(9)>0 & OTHER(9)=VET(9) then OTHER(9)=0;
         if R(23612.)=–4 then RELWEL(9)=0;                          else RELWEL(9)=R(23612.);
         if R(23614.)=–4 then RELREG(9)=0;                          else RELREG(9)=R(23614.);
        FAMILY(9)=FAMSZ87; /* To create family size (i.e. FAMSZ87) search thru the household
               enumeration. Increment family size if the relationship to the youth is a relative. Do not increase
               family size if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
       DCL COMPONENT(19) fixed DEC(9);
       do I=9;
         MN(I,1)=0; MN(I,2)=0; MN(I,3)=0;
         MN_CENSUS(I,1)=0; MN_CENSUS(I,2)=0; MN_CENSUS(I,3)=0;
         FAMINC(I)=0; FAMINC_CENSUS(I)=0;
         INCOME(I)=–3; INCOME_CENSUS(I)=-3;
         COMPONENT(1)=MIL(I);                                         COMPONENT(2)=MILS(I);
         COMPONENT(3)=WPS(I);                                         COMPONENT(4)=WPSS(I);
         COMPONENT(5)=SEI(I);                                         COMPONENT(6)=SEIS(I);
         COMPONENT(7)=UI(I);                                          COMPONENT(8)=UIS(I);
         COMPONENT(9)=ALIM(I);                                        COMPONENT(10)=CHSP(I);
         COMPONENT(11)=AFDC(I);                                       COMPONENT(12)=PUBLIC(I);
         COMPONENT(13)=ED(I);                                         COMPONENT(14)=EDSS(I);
```

```
      COMPONENT(15)=VET(I);                         COMPONENT(16)=OTHER(I);
      COMPONENT(17)=RELWEL(I);                      COMPONENT(18)=RELREG(I);
      COMPONENT(19)=FOOD(I);
      if WEIGHT(I)=0 then do;
        INCOME(I)=-5; INCOME_CENSUS(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
      end;
      else do;
          do K=1 to 19;
            if COMPONENT(K)>-4 then do;
              if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
              else FAMINC(I)=FAMINC(I)+COMPONENT(K);
            end;
          if K<19 then do;
              if COMPONENT(K)<0 then MN_CENSUS(I,ABS(COMPONENT(K)))=
                  MN_CENSUS(I,ABS(COMPONENT(K)))+1;
              else FAMINC_CENSUS(I)=FAMINC_CENSUS(I)+COMPONENT(K);
            end;
          end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
        SUMN_CENSUS(I)=MN_CENSUS(I,1) + MN_CENSUS(I,2) + MN_CENSUS(I,3);
        if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
        if SUMN_CENSUS(I)=0 then INCOME_CENSUS(I)=FAMINC_CENSUS(I);
        else do J=1 to 3; if MN_CENSUS(I,J)>0 then INCOME_CENSUS(I)=-J; end;
      if FAMILY(I)>0 then do;
         LEVEL(I)=TABLE(I,S(I),FAMILY(I));
         if INCOME(I)>LEVEL(I) then CPS(I)=0;
         else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
         else CPS(I)=-3;
      end;
     end;
    end;

  do I=1 to 3;
    if MN(9,I)>0 then MN(9,I)=1;
    if MN_CENSUS(9,I)>0 then MN_CENSUS(9,I)=1;
  end;
  if SUMN(9)>0 then SUMN(9)=1;
  if SUMN_CENSUS(9)>0 then SUMN_CENSUS(9)=1;

 INCOME(9)=R(24447.);
 INCOME_CENSUS(9)=R(24448.);
 CPS(9)=R(24449.);
 LEVEL(9)=R(24450.);
```

```
/* 1988 VARIABLES */

        DCL                              1 FAMILY_INCOME (10),
          5 (AFDC, ALIM, CHSP, CPS, ED, EDSS, FAMILY, FAMINC, FOOD, INCOME, LEVEL, MIL,
MILS,
              MN (3), OTHER, PUBLIC, RELREG, RELWEL, S, SEI, SEIS, SUMN, TABLE(3,15), UI, UIS,
VET,
             WPS, WPSS) fixed DEC(9);
   /*  1988 poverty income guidelines table */ TABLE(10,1,1)=5700; TABLE(10,2,1)=7210;
TABLE(10,3,1)=6650;
        do J=2 to 15;                    TABLE(10,1,J)=TABLE(10,1,J-1)+1960;
                                         TABLE(10,2,J)=TABLE(10,2,J-1)+2450; TABLE(10,3,J)=TABLE(10,3,J-
1)+2250;
    end;
    S(10)=1; if STATE88=2 then S(10)=2;                   else if STATE88=15 then S(10)=3;
     if R(27224.)=-4 then MIL(10)=0;                      else MIL(10)=R(27224.);
     if R(27225.)=-4 then WPS(10)=0;                      else WPS(10)=R(27225.);
     if R(27228.)=-4 then SEI(10)=0;                      else SEI(10)=R(27228.);
     if R(27242.)=-4 & R(27243.)=-4 then UI(10)=0;
    else if R(27242.)>0 & R(27243.)>0 then UI(10)=R(27242.) * R(27243.);
    else if R(27243.)<0 then UI(10)=R(27243.);            else UI(10)=-3;
     if R(27246.)=-4 then MILS(10)=0;                     else MILS(10)=R(27246.);
     if R(27247.)=-4 then WPSS(10)=0;           else WPSS(10)=R(27247.);
     if R(27250.)=-4 then SEIS(10)=0;                     else SEIS(10)=R(27250.);
     if R(27264.)=-4 & R(27265.)=-4 then UIS(10)=0;
    else if R(27264.)>0 & R(27265.)>0 then UIS(10)=R(27264.)*R(27265.);
    else if R(27265.)<0 then UIS(10)=R(27265.);           else UIS(10)=-3;
     if R(27268.)=-4 then ALIM(10)=0;                     else ALIM(10)=R(27268.);
     if R(27273.)=-4 then CHSP(10)=0;                     else CHSP(10)=R(27273.);
     if R(27289.)=-4 then AFDC(10)=0;
    else do; B=0;
         if R(27277.)>0 & B^=na then B=B+1;               else if R(27277.)>-4 & R(27277.)<0 then B=na;
         if R(27278.)>0 & B^=na then B=B+1;               else if R(27278.)>-4 & R(27278.)<0 then B=na;
         if R(27279.)>0 & B^=na then B=B+1;               else if R(27279.)>-4 & R(27279.)<0 then B=na;
         if R(27280.)>0 & B^=na then B=B+1;               else if R(27280.)>-4 & R(27280.)<0 then B=na;
         if R(27281.)>0 & B^=na then B=B+1;               else if R(27281.)>-4 & R(27281.)<0 then B=na;
         if R(27282.)>0 & B^=na then B=B+1;               else if R(27282.)>-4 & R(27282.)<0 then B=na;
         if R(27283.)>0 & B^=na then B=B+1;               else if R(27283.)>-4 & R(27283.)<0 then B=na;
         if R(27284.)>0 & B^=na then B=B+1;               else if R(27284.)>-4 & R(27284.)<0 then B=na;
         if R(27285.)>0 & B^=na then B=B+1;               else if R(27285.)>-4 & R(27285.)<0 then B=na;
         if R(27286.)>0 & B^=na then B=B+1;               else if R(27286.)>-4 & R(27286.)<0 then B=na;
         if R(27287.)>0 & B^=na then B=B+1;               else if R(27287.)>-4 & R(27287.)<0 then B=na;
         if R(27288.)>0 & B^=na then B=B+1;               else if R(27288.)>-4 & R(27288.)<0 then B=na;
      if R(27289.)<0 then AFDC(10)=R(27289.); else if B<=0 then AFDC(10)=-3; else AFDC(10)=R(27289.)
* B;
    end;
     if R(27303.)=-4 then FOOD(10)=0;
    else do; B=0;
         if R(27291.)>0 & B^=na then B=B+1;               else if R(27291.)>-4 & R(27291.)<0 then B=na;
         if R(27292.)>0 & B^=na then B=B+1;               else if R(27292.)>-4 & R(27292.)<0 then B=na;
         if R(27293.)>0 & B^=na then B=B+1;               else if R(27293.)>-4 & R(27293.)<0 then B=na;
         if R(27294.)>0 & B^=na then B=B+1;               else if R(27294.)>-4 & R(27294.)<0 then B=na;
         if R(27295.)>0 & B^=na then B=B+1;               else if R(27295.)>-4 & R(27295.)<0 then B=na;
         if R(27296.)>0 & B^=na then B=B+1;               else if R(27296.)>-4 & R(27296.)<0 then B=na;
         if R(27297.)>0 & B^=na then B=B+1;               else if R(27297.)>-4 & R(27297.)<0 then B=na;
         if R(27298.)>0 & B^=na then B=B+1;               else if R(27298.)>-4 & R(27298.)<0 then B=na;
         if R(27299.)>0 & B^=na then B=B+1;                    else if R(27299.)>-4 & R(27299.)<0 then B=na;
         if R(27300.)>0 & B^=na then B=B+1;                    else if R(27300.)>-4 & R(27300.)<0 then B=na;
         if R(27301.)>0 & B^=na then B=B+1;                    else if R(27301.)>-4 & R(27301.)<0 then B=na;
         if R(27302.)>0 & B^=na then B=B+1;                    else if R(27302.)>-4 & R(27302.)<0 then B=na;
      if R(27303.)<0 then FOOD(10)=R(27303.); else if B<=0 then FOOD(10)=-3; else FOOD(10)=R(27303.)
* B;
    end;
    if R(27317.)=-4 then PUBLIC(10)=0;
    else do; B=0;
         if R(27305.)>0 & B^=na then B=B+1;                    else if R(27305.)>-4 & R(27305.)<0 then B=na;
         if R(27306.)>0 & B^=na then B=B+1;                    else if R(27306.)>-4 & R(27306.)<0 then B=na;
         if R(27307.)>0 & B^=na then B=B+1;                    else if R(27307.)>-4 & R(27307.)<0 then B=na;
         if R(27308.)>0 & B^=na then B=B+1;                    else if R(27308.)>-4 & R(27308.)<0 then B=na;
         if R(27309.)>0 & B^=na then B=B+1;                    else if R(27309.)>-4 & R(27309.)<0 then B=na;
         if R(27310.)>0 & B^=na then B=B+1;                    else if R(27310.)>-4 & R(27310.)<0 then B=na;
         if R(27311.)>0 & B^=na then B=B+1;                    else if R(27311.)>-4 & R(27311.)<0 then B=na;
         if R(27312.)>0 & B^=na then B=B+1;                    else if R(27312.)>-4 & R(27312.)<0 then B=na;
         if R(27313.)>0 & B^=na then B=B+1;                    else if R(27313.)>-4 & R(27313.)<0 then B=na;
         if R(27314.)>0 & B^=na then B=B+1;                    else if R(27314.)>-4 & R(27314.)<0 then B=na;
         if R(27315.)>0 & B^=na then B=B+1;                    else if R(27315.)>-4 & R(27315.)<0 then B=na;
         if R(27316.)>0 & B^=na then B=B+1;                    else if R(27316.)>-4 & R(27316.)<0 then B=na;
      if R(27317.)<0 then PUBLIC(10)=R(27317.); else if B<=0 then PUBLIC(10)=-3; else
            PUBLIC(10)=R(27317.)*B;
    end;
    if R(27322.)=-4 then ED(10)=0;                            else ED(10)=R(27322.);
    if R(27323.)=-4 then EDSS(10)=0;                          else EDSS(10)=R(27323.);
    if R(27325.)=-4 then VET(10)=0;                           else VET(10)=R(27325.);
    if R(27332.)=-4 then OTHER(10)=0;                         else OTHER(10)=R(27332.);
    if VET(10)>0 & OTHER(10)=VET(10) then OTHER(10)=0;
    if R(27339.)=-4 then RELWEL(10)=0;                        else RELWEL(10)=R(27339.);
    if R(27341.)=-4 then RELREG(10)=0;                        else RELREG(10)=R(27341.);
    FAMILY(10)=FAMSZ88; /* To create family size (i.e. FAMSZ88) search thru the household
            enumeration. Increment family size  if the relationship to the youth is a relative. Do not increase
            family size if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
   DCL COMPONENT(19) fixed DEC(9);
   do I=10;
     MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
     COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
     COMPONENT(4)=WPSS(I);         COMPONENT(5)=SEI(I);           COMPONENT(6)=SEIS(I);
     COMPONENT(7)=UI(I);           COMPONENT(8)=UIS(I);
                                   COMPONENT(9)=ALIM(I);
     COMPONENT(10)=CHSP(I);        COMPONENT(11)=AFDC(I);
                                   COMPONENT(12)=PUBLIC(I);
     COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);
                                   COMPONENT(15)=VET(I);
     COMPONENT(16)=OTHER(I);       COMPONENT(17)=RELWEL(I);
     COMPONENT(18)=RELREG(I);      COMPONENT(19)=FOOD(I);
     if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
     else do;
         do K=1 to 19;
           if COMPONENT(K)>-4 then do;
            if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
            else FAMINC(I)=FAMINC(I)+COMPONENT(K);
           end;
         end;
      SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
      if SUMN(I)=0 then INCOME(I)=FAMINC(I);
      else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do;
      LEVEL(I)=TABLE(I,S(I),FAMILY(I));                if INCOME(I)>LEVEL(I) then CPS(I)=0;
      else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
      else CPS(I)=-3;
    end;
   end;
  end;
 do I=1 to 3; if MN(10,I)>0 then MN(10,I)=1; end;
 if SUMN(10)>0 then SUMN(10)=1;

INCOME(10)=R(28702.);            CPS(10)=R(28704.);        LEVEL(10)=R(28705.);
```

/* 1989 VARIABLES */

```
        DCL                               1 FAMILY_INCOME (11),
            5 (AFDC, ALIM, CHSP, CPS, ED, EDSS, FAMILY, FAMINC, FOOD, INCOME, LEVEL, MIL,
MILS,
              MN (3), OTHER, PUBLIC, RELREG, RELWEL, S, SEI, SEIS, SUMN, TABLE(3,15), UI, UIS,
VET,
             WPS, WPSS) fixed DEC(9);
    /* 1989 poverty income guidelines table */ TABLE(11,1,1)=5980; TABLE(11,2,1)=7480;
TABLE(11,3,1)=6870;
        do J=2 to 15;                    TABLE(11,1,J)=TABLE(11,1,J-1)+2040;
                                        TABLE(11,2,J)=TABLE(11,2,J-1)+2550; TABLE(11,3,J)=TABLE(11,3,J-
1)+2350;
    end;
    S(11)=1; if R(29922.)=2 then S(11)=2;                  else if R(29922.)=15 then S(11)=3;
     if R(29713.)=-4 then MIL(11)=0;                       else MIL(11)=R(29713.);
     if R(29714.)=-4 then WPS(11)=0;                       else WPS(11)=R(29714.);
     if R(29717.)=-4 then SEI(11)=0;                       else SEI(11)=R(29717.);
     if R(29731.)=-4 & R(29732.)=-4 then UI(11)=0;
    else if R(29731.)>0 & R(29732.)>0 then UI(11)=R(29731.) * R(29732.);
    else if R(29732.)<0 then UI(11)=R(29732.);             else UI(11)=-3;
     if R(29735.)=-4 then MILS(11)=0;                      else MILS(11)=R(29735.);
     if R(29736.)=-4 then WPSS(11)=0;            else WPSS(11)=R(29736.);
     if R(29739.)=-4 then SEIS(11)=0;                      else SEIS(11)=R(29739.);
     if R(29753.)=-4 & R(29754.)=-4 then UIS(11)=0;
     else if R(29753.)>0 & R(29754.)>0 then UIS(11)=R(29753.) * R(29754.);
    else if R(29754.)<0 then UIS(11)=R(29754.);            else UIS(11)=-3;
     if R(29757.)=-4 then ALIM(11)=0;                      else ALIM(11)=R(29757.);
     if R(29759.)=-4 then CHSP(11)=0;            else CHSP(11)=R(29759.);
     if R(29773.)=-4 then AFDC(11)=0;
    else do; B=0;
          if R(29761.)>0 & B^=na then B=B+1;               else if R(29761.)>-4 & R(29761.)<0 then B=na;
          if R(29762.)>0 & B^=na then B=B+1;               else if R(29762.)>-4 & R(29762.)<0 then B=na;
          if R(29763.)>0 & B^=na then B=B+1;               else if R(29763.)>-4 & R(29763.)<0 then B=na;
          if R(29764.)>0 & B^=na then B=B+1;               else if R(29764.)>-4 & R(29764.)<0 then B=na;
          if R(29765.)>0 & B^=na then B=B+1;               else if R(29765.)>-4 & R(29765.)<0 then B=na;
          if R(29766.)>0 & B^=na then B=B+1;               else if R(29766.)>-4 & R(29766.)<0 then B=na;
          if R(29767.)>0 & B^=na then B=B+1;               else if R(29767.)>-4 & R(29767.)<0 then B=na;
          if R(29768.)>0 & B^=na then B=B+1;               else if R(29768.)>-4 & R(29768.)<0 then B=na;
          if R(29769.)>0 & B^=na then B=B+1;               else if R(29769.)>-4 & R(29769.)<0 then B=na;
          if R(29770.)>0 & B^=na then B=B+1;               else if R(29770.)>-4 & R(29770.)<0 then B=na;
          if R(29771.)>0 & B^=na then B=B+1;               else if R(29771.)>-4 & R(29771.)<0 then B=na;
          if R(29772.)>0 & B^=na then B=B+1;               else if R(29772.)>-4 & R(29772.)<0 then B=na;
       if R(29773.)<0 then AFDC(11)=R(29773.); else if B<=0 then AFDC(11)=-3; else AFDC(11)=R(29773.)
* B;
    end;
     if R(29787.)=-4 then FOOD(11)=0;
    else do; B=0;
          if R(29775.)>0 & B^=na then B=B+1;               else if R(29775.)>-4 & R(29775.)<0 then B=na;
          if R(29776.)>0 & B^=na then B=B+1;               else if R(29776.)>-4 & R(29776.)<0 then B=na;
          if R(29777.)>0 & B^=na then B=B+1;               else if R(29777.)>-4 & R(29777.)<0 then B=na;
          if R(29778.)>0 & B^=na then B=B+1;               else if R(29778.)>-4 & R(29778.)<0 then B=na;
          if R(29779.)>0 & B^=na then B=B+1;               else if R(29779.)>-4 & R(29779.)<0 then B=na;
          if R(29780.)>0 & B^=na then B=B+1;               else if R(29780.)>-4 & R(29780.)<0 then B=na;
          if R(29781.)>0 & B^=na then B=B+1;               else if R(29781.)>-4 & R(29781.)<0 then B=na;
          if R(29782.)>0 & B^=na then B=B+1;               else if R(29782.)>-4 & R(29782.)<0 then B=na;
          if R(29783.)>0 & B^=na then B=B+1;                  else if R(29783.)>-4 & R(29783.)<0 then B=na;
          if R(29784.)>0 & B^=na then B=B+1;                  else if R(29784.)>-4 & R(29784.)<0 then B=na;
          if R(29785.)>0 & B^=na then B=B+1;                  else if R(29785.)>-4 & R(29785.)<0 then B=na;
          if R(29786.)>0 & B^=na then B=B+1;                  else if R(29786.)>-4 & R(29786.)<0 then B=na;
        if R(29787.)<0 then FOOD(11)=R(29787.); else if B<=0 then FOOD(11)=-3; else FOOD(11)=R(29787.)
* B;
     end;
     if R(29801.)=-4 then PUBLIC(11)=0;
     else do; B=0;
          if R(29789.)>0 & B^=na then B=B+1;                  else if R(29789.)>-4 & R(29789.)<0 then B=na;
          if R(29790.)>0 & B^=na then B=B+1;                  else if R(29790.)>-4 & R(29790.)<0 then B=na;
          if R(29791.)>0 & B^=na then B=B+1;                  else if R(29791.)>-4 & R(29791.)<0 then B=na;
          if R(29792.)>0 & B^=na then B=B+1;                  else if R(29792.)>-4 & R(29792.)<0 then B=na;
          if R(29793.)>0 & B^=na then B=B+1;                  else if R(29793.)>-4 & R(29793.)<0 then B=na;
          if R(29794.)>0 & B^=na then B=B+1;                  else if R(29794.)>-4 & R(29794.)<0 then B=na;
          if R(29795.)>0 & B^=na then B=B+1;                  else if R(29795.)>-4 & R(29795.)<0 then B=na;
          if R(29796.)>0 & B^=na then B=B+1;                  else if R(29796.)>-4 & R(29796.)<0 then B=na;
          if R(29797.)>0 & B^=na then B=B+1;                  else if R(29797.)>-4 & R(29797.)<0 then B=na;
          if R(29798.)>0 & B^=na then B=B+1;                  else if R(29798.)>-4 & R(29798.)<0 then B=na;
          if R(29799.)>0 & B^=na then B=B+1;                  else if R(29799.)>-4 & R(29799.)<0 then B=na;
          if R(29800.)>0 & B^=na then B=B+1;                  else if R(29800.)>-4 & R(29800.)<0 then B=na;
        if R(29801.)<0 then PUBLIC(11)=R(29801.); else if B<=0 then PUBLIC(11)=-3; else
PUBLIC(11)=R(29801.)*B;
         end;
         if R(29806.)=-4 then ED(11)=0;                       else ED(11)=R(29806.);
         if R(29807.)=-4 then EDSS(11)=0;                                 else EDSS(11)=R(29807.);
         if R(29809.)=-4 then VET(11)=0;                      else VET(11)=R(29809.);
         if R(29813.)=-4 then OTHER(11)=0;                    else OTHER(11)=R(29813.);
         if VET(11)>0 & OTHER(11)=VET(11) then OTHER(11)=0;
         if R(29820.)=-4 then RELWEL(11)=0;                   else RELWEL(11)=R(29820.);
         if R(29822.)=-4 then RELREG(11)=0;                   else RELREG(11)=R(29822.);
         FAMILY(11)=FAMSZ89; /* To create family size (i.e. FAMSZ89) search thru the household
              enumeration. Increment family size, if the relationship to the youth is a relative. Do not increase
              family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
        DCL COMPONENT(19) fixed DEC(9);
        do I=11;
          MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
          COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);            COMPONENT(3)=WPS(I);
          COMPONENT(4)=WPSS(I);         COMPONENT(5)=SEI(I);             COMPONENT(6)=SEIS(I);
          COMPONENT(7)=UI(I);           COMPONENT(8)=UIS(I);
                                        COMPONENT(9)=ALIM(I);
          COMPONENT(10)=CHSP(I);        COMPONENT(11)=AFDC(I);
                                        COMPONENT(12)=PUBLIC(I);
          COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);
                                        COMPONENT(15)=VET(I);
          COMPONENT(16)=OTHER(I);       COMPONENT(17)=RELWEL(I);
          COMPONENT(18)=RELREG(I);      COMPONENT(19)=FOOD(I);
          if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
          else do;
              do K=1 to 19;
                 if COMPONENT(K)>-4 then do;
                   if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                   else FAMINC(I)=FAMINC(I)+COMPONENT(K);
                 end;
              end;
            SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
            if SUMN(I)=0 then INCOME(I)=FAMINC(I);
           else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
          if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I));
              if INCOME(I)>LEVEL(I) then CPS(I)=0;
              else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
              else CPS(I)=-3;
          end;
         end;
        end;
     do I=1 to 3; if MN(11,I)>0 then MN(11,I)=1; end;
     if SUMN(11)>0 then SUMN(11)=1;

    INCOME(11)=R(30740);             CPS(11)=R(30741);        LEVEL(11)=R(30742);
```

```
/* 1990 VARIABLES */

    do I=12; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
        FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
        do J=1 to 3; MN(I,J)=-4; end;
        OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
        do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
        UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
    end;
   TABLE(12,1,1)=6280; TABLE(12,2,1)=7840; TABLE(12,3,1)=7230;
   do J=2 to 15; TABLE(12,1,J)=TABLE(12,1,J-1)+2140; TABLE(12,2,J)=TABLE(12,2,J-1)+2680;
     TABLE(12,3,J)=TABLE(12,3,J-1)+2460;
   end;
   S(12)=1; if STATE90=2 then S(12)=2;                    else if STATE90=15 then S(12)=3;
    if R(32793.)=-4 then MIL(12)=0;                       else MIL(12)=R(32793.);
    if R(32794.)=-4 then WPS(12)=0;                       else WPS(12)=R(32794.);
    if R(32797.)=-4 then SEI(12)=0;                       else SEI(12)=R(32797.);
    if R(32811.)=-4 & R(32812.)=-4 then UI(12)=0;
   else if R(32811.)>0 & R(32812.)>0 then UI(12)=R(32811.) * R(32812.);
   else if R(32812.)<0 then UI(12)=R(32812.);             else UI(12)=-3;
    if R(32815.)=-4 then MILS(12)=0;                      else MILS(12)=R(32815.);
    if R(32816.)=-4 then WPSS(12)=0;                      else WPSS(12)=R(32816.);
    if R(32819.)=-4 then SEIS(12)=0;                      else SEIS(12)=R(32819.);
    if R(32833.)=-4 & R(32834.)=-4 then UIS(12)=0;
   else if R(32833.)>0 & R(32834.)>0 then UIS(12)=R(32833.) * R(32834.);
   else if R(32834.)<0 then UIS(12)=R(32834.);            else UIS(12)=-3;
    if R(32837.)=-4 then ALIM(12)=0;                      else ALIM(12)=R(32837.);
    if R(32839.)=-4 then CHSP(12)=0;                      else CHSP(12)=R(32839.);
    if R(32853.)=-4 then AFDC(12)=0;
   else do; B=0;
        if R(32841.)>0 & B^=na then B=B+1;                else if R(32841.)>-4 & R(32841.)<0 then B=na;
        if R(32842.)>0 & B^=na then B=B+1;                else if R(32842.)>-4 & R(32842.)<0 then B=na;
        if R(32843.)>0 & B^=na then B=B+1;                else if R(32843.)>-4 & R(32843.)<0 then B=na;
        if R(32844.)>0 & B^=na then B=B+1;                else if R(32844.)>-4 & R(32844.)<0 then B=na;
        if R(32845.)>0 & B^=na then B=B+1;                else if R(32845.)>-4 & R(32845.)<0 then B=na;
        if R(32846.)>0 & B^=na then B=B+1;                else if R(32846.)>-4 & R(32846.)<0 then B=na;
        if R(32847.)>0 & B^=na then B=B+1;                else if R(32847.)>-4 & R(32847.)<0 then B=na;
        if R(32848.)>0 & B^=na then B=B+1;                else if R(32848.)>-4 & R(32848.)<0 then B=na;
        if R(32849.)>0 & B^=na then B=B+1;                else if R(32849.)>-4 & R(32849.)<0 then B=na;
        if R(32850.)>0 & B^=na then B=B+1;                else if R(32850.)>-4 & R(32850.)<0 then B=na;
        if R(32851.)>0 & B^=na then B=B+1;                else if R(32851.)>-4 & R(32851.)<0 then B=na;
        if R(32852.)>0 & B^=na then B=B+1;                else if R(32852.)>-4 & R(32852.)<0 then B=na;
     if R(32853.)<0 then AFDC(12)=R(32853.); else if B<=0 then AFDC(12)=-3; else AFDC(12)=R(32853.)
* B;
   end;
    if R(32867.)=-4 then FOOD(12)=0;
   else do; B=0;
        if R(32855.)>0 & B^=na then B=B+1;                else if R(32855.)>-4 & R(32855.)<0 then B=na;
        if R(32856.)>0 & B^=na then B=B+1;                else if R(32856.)>-4 & R(32856.)<0 then B=na;
        if R(32857.)>0 & B^=na then B=B+1;                else if R(32857.)>-4 & R(32857.)<0 then B=na;
        if R(32858.)>0 & B^=na then B=B+1;                else if R(32858.)>-4 & R(32858.)<0 then B=na;
        if R(32859.)>0 & B^=na then B=B+1;                else if R(32859.)>-4 & R(32859.)<0 then B=na;
        if R(32860.)>0 & B^=na then B=B+1;                else if R(32860.)>-4 & R(32860.)<0 then B=na;
        if R(32861.)>0 & B^=na then B=B+1;                else if R(32861.)>-4 & R(32861.)<0 then B=na;
        if R(32862.)>0 & B^=na then B=B+1;                else if R(32862.)>-4 & R(32862.)<0 then B=na;
```

```
         if R(32863.)>0 & B^=na then B=B+1;                   else if R(32863.)>-4 & R(32863.)<0 then B=na;
         if R(32864.)>0 & B^=na then B=B+1;                   else if R(32864.)>-4 & R(32864.)<0 then B=na;
         if R(32865.)>0 & B^=na then B=B+1;                   else if R(32865.)>-4 & R(32865.)<0 then B=na;
         if R(32866.)>0 & B^=na then B=B+1;                   else if R(32866.)>-4 & R(32866.)<0 then B=na;
       if R(32867.)<0 then FOOD(12)=R(32867.); else if B<=0 then FOOD(12)=-3; else FOOD(12)=R(32867.)
* B;
     end;
     if R(32893.)=-4 then PUBLIC(12)=0;
     else do; B=0;
         if R(32881.)>0 & B^=na then B=B+1;                   else if R(32881.)>-4 & R(32881.)<0 then B=na;
         if R(32882.)>0 & B^=na then B=B+1;                   else if R(32882.)>-4 & R(32882.)<0 then B=na;
         if R(32883.)>0 & B^=na then B=B+1;                   else if R(32883.)>-4 & R(32883.)<0 then B=na;
         if R(32884.)>0 & B^=na then B=B+1;                   else if R(32884.)>-4 & R(32884.)<0 then B=na;
         if R(32885.)>0 & B^=na then B=B+1;                   else if R(32885.)>-4 & R(32885.)<0 then B=na;
         if R(32886.)>0 & B^=na then B=B+1;                   else if R(32886.)>-4 & R(32886.)<0 then B=na;
         if R(32887.)>0 & B^=na then B=B+1;                   else if R(32887.)>-4 & R(32887.)<0 then B=na;
         if R(32888.)>0 & B^=na then B=B+1;                   else if R(32888.)>-4 & R(32888.)<0 then B=na;
         if R(32889.)>0 & B^=na then B=B+1;                   else if R(32889.)>-4 & R(32889.)<0 then B=na;
         if R(32890.)>0 & B^=na then B=B+1;                   else if R(32890.)>-4 & R(32890.)<0 then B=na;
         if R(32891.)>0 & B^=na then B=B+1;                   else if R(32891.)>-4 & R(32891.)<0 then B=na;
         if R(32892.)>0 & B^=na then B=B+1;                   else if R(32892.)>-4 & R(32892.)<0 then B=na;
       if R(32893.)<0 then PUBLIC(12)=R(32893.); else if B<=0 then PUBLIC(12)=-3; else
PUBLIC(12)=R(32893.)*B;
     end;
     if R(32900.)=-4 then ED(12)=0;                           else ED(12)=R(32900.);
     if R(32901.)=-4 then EDSS(12)=0;                         else EDSS(12)=R(32901.);
     if R(32902.)=-4 then VET(12)=0;                          else VET(12)=R(32902.);
     if R(32906.)=-4 then OTHER(12)=0;                        else OTHER(12)=R(32906.);
     if VET(12)>0 & OTHER(12)=VET(12) then OTHER(12)=0;
     if R(32916.)=-4 then RELWEL(12)=0;                       else RELWEL(12)=R(32916.);
     if R(32918.)=-4 then RELREG(12)=0;                       else RELREG(12)=R(32918.);

 /*  HAND EDITS FOR FAMSZ90 */ if ID=2033 then FAMSZ90=1; if ID=3617 then FAMSZ90=1;
     FAMILY(12)=FAMSZ90; /* To create family size (i.e. famsz90) search thru the household enumeration.
              Increment family size if the relationship to the youth is a relative. Do not increase family size if the
              code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
    DCL COMPONENT(19) fixed DEC(9);
    do I=12;
      MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
      COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);           COMPONENT(3)=WPS(I);
      COMPONENT(4)=WPSS(I);         COMPONENT(5)=SEI(I);            COMPONENT(6)=SEIS(I);
      COMPONENT(7)=UI(I);           COMPONENT(8)=UIS(I);
                                    COMPONENT(9)=ALIM(I);
      COMPONENT(10)=CHSP(I);        COMPONENT(11)=AFDC(I);
                                    COMPONENT(12)=PUBLIC(I);
      COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);
                                    COMPONENT(15)=VET(I);
      COMPONENT(16)=OTHER(I);       COMPONENT(17)=RELWEL(I);
      COMPONENT(18)=RELREG(I);      COMPONENT(19)=FOOD(I);
      if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
      else do;
         do K=1 to 19;
           if COMPONENT(K)>-4 then do;
             if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
             else FAMINC(I)=FAMINC(I)+COMPONENT(K);
```

```
          end;
        end;
      SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
      if SUMN(I)=0 then INCOME(I)=FAMINC(I); else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then
CPS(I)=0;
       else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
    end;
   end;
  end;
  do I=1 to 3; if MN(12,I)>0 then MN(12,I)=1; end;       if SUMN(12)>0 then SUMN(12)=1;
INCOME(12)=R(34007.)   CPS(12)=R(34008.)              LEVEL(12)=R(34009.)
```

```
/* 1991 VARIABLES */
    do I=13; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
        FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
        do J=1 to 3; MN(I,J)=-4; end;
        OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
        do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
        UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
    end;
    TABLE(13,1,1)=6620; TABLE(13,2,1)=8290; TABLE(13,3,1)=7610;
    do J=2 to 15; TABLE(13,1,J)=TABLE(13,1,J-1)+2260; TABLE(13,2,J)=TABLE(13,2,J-1)+2820;
      TABLE(13,3,J)=TABLE(13,3,J-1)+2600;
    end;
    S(13)=1; if STATE91=2 then S(13)=2;                 else if STATE91=15 then S(13)=3;
    if R(35589.)=-4 then MIL(13)=0;                     else MIL(13)=R(35589.);
    if R(35590.)=-4 then WPS(13)=0;                     else WPS(13)=R(35590.);
    if R(35593.)=-4 then SEI(13)=0;                     else SEI(13)=R(35593.);
    if R(35607.)=-4 & R(35608.)=-4 then UI(13)=0;
    else if R(35607.)>0 & R(35608.)>0 then UI(13)=R(35607.) * R(35608.);
    else if R(35608.)<0 then UI(13)=R(35608.);          else UI(13)=-3;
    if R(35611.)=-4 then MILS(13)=0;                    else MILS(13)=R(35611.);
    if R(35612.)=-4 then WPSS(13)=0;                    else WPSS(13)=R(35612.);
    if R(35615.)=-4 then SEIS(13)=0;                    else SEIS(13)=R(35615.);
    if R(35629.)=-4 & R(35630.)=-4 then UIS(13)=0;
    else if R(35629.)>0 & R(35630.)>0 then UIS(13)=R(35629.) * R(35630.);
    else if R(35630.)<0 then UIS(13)=R(35630.);         else UIS(13)=-3;
    if R(35633.)=-4 then ALIM(13)=0;                    else ALIM(13)=R(35633.);
    if R(35635.)=-4 then CHSP(13)=0;                    else CHSP(13)=R(35635.);
    if R(35649.)=-4 then AFDC(13)=0;
    else do; B=0;
        if R(35637.)>0 & B^=na then B=B+1;              else if R(35637.)>-4 & R(35637.)<0 then B=na;
        if R(35638.)>0 & B^=na then B=B+1;              else if R(35638.)>-4 & R(35638.)<0 then B=na;
        if R(35639.)>0 & B^=na then B=B+1;              else if R(35639.)>-4 & R(35639.)<0 then B=na;
        if R(35640.)>0 & B^=na then B=B+1;              else if R(35640.)>-4 & R(35640.)<0 then B=na;
        if R(35641.)>0 & B^=na then B=B+1;              else if R(35641.)>-4 & R(35641.)<0 then B=na;
        if R(35642.)>0 & B^=na then B=B+1;              else if R(35642.)>-4 & R(35642.)<0 then B=na;
        if R(35643.)>0 & B^=na then B=B+1;              else if R(35643.)>-4 & R(35643.)<0 then B=na;
        if R(35644.)>0 & B^=na then B=B+1;              else if R(35644.)>-4 & R(35644.)<0 then B=na;
        if R(35645.)>0 & B^=na then B=B+1;              else if R(35645.)>-4 & R(35645.)<0 then B=na;
        if R(35646.)>0 & B^=na then B=B+1;              else if R(35646.)>-4 & R(35646.)<0 then B=na;
        if R(35647.)>0 & B^=na then B=B+1;              else if R(35647.)>-4 & R(35647.)<0 then B=na;
        if R(35648.)>0 & B^=na then B=B+1;              else if R(35648.)>-4 & R(35648.)<0 then B=na;
      if R(35649.)<0 then AFDC(13)=R(35649.); else if B<=0 then AFDC(13)=-3; else AFDC(13)=R(35649.)
* B;
    end;
    if R(35663.)=-4 then FOOD(13)=0;
    else do; B=0;
        if R(35651.)>0 & B^=na then B=B+1;              else if R(35651.)>-4 & R(35651.)<0 then B=na;
        if R(35652.)>0 & B^=na then B=B+1;              else if R(35652.)>-4 & R(35652.)<0 then B=na;
        if R(35653.)>0 & B^=na then B=B+1;              else if R(35653.)>-4 & R(35653.)<0 then B=na;
        if R(35654.)>0 & B^=na then B=B+1;              else if R(35654.)>-4 & R(35654.)<0 then B=na;
        if R(35655.)>0 & B^=na then B=B+1;              else if R(35655.)>-4 & R(35655.)<0 then B=na;
        if R(35656.)>0 & B^=na then B=B+1;              else if R(35656.)>-4 & R(35656.)<0 then B=na;
        if R(35657.)>0 & B^=na then B=B+1;              else if R(35657.)>-4 & R(35657.)<0 then B=na;
        if R(35658.)>0 & B^=na then B=B+1;              else if R(35658.)>-4 & R(35658.)<0 then B=na;
        if R(35659.)>0 & B^=na then B=B+1;            else if R(35659.)>-4 & R(35659.)<0 then B=na;
        if R(35660.)>0 & B^=na then B=B+1;            else if R(35660.)>-4 & R(35660.)<0 then B=na;
        if R(35661.)>0 & B^=na then B=B+1;            else if R(35661.)>-4 & R(35661.)<0 then B=na;
        if R(35662.)>0 & B^=na then B=B+1;            else if R(35662.)>-4 & R(35662.)<0 then B=na;
      if R(35663.)<0 then FOOD(13)=R(35663.); else if B<=0 then FOOD(13)=-3; else FOOD(13)=R(35663.)
* B;
    end;
    if R(35689.)=-4 then PUBLIC(13)=0;
    else do; B=0;
        if R(35677.)>0 & B^=na then B=B+1;            else if R(35677.)>-4 & R(35677.)<0 then B=na;
        if R(35678.)>0 & B^=na then B=B+1;            else if R(35678.)>-4 & R(35678.)<0 then B=na;
        if R(35679.)>0 & B^=na then B=B+1;            else if R(35679.)>-4 & R(35679.)<0 then B=na;
        if R(35680.)>0 & B^=na then B=B+1;            else if R(35680.)>-4 & R(35680.)<0 then B=na;
        if R(35681.)>0 & B^=na then B=B+1;            else if R(35681.)>-4 & R(35681.)<0 then B=na;
        if R(35682.)>0 & B^=na then B=B+1;            else if R(35682.)>-4 & R(35682.)<0 then B=na;
        if R(35683.)>0 & B^=na then B=B+1;            else if R(35683.)>-4 & R(35683.)<0 then B=na;
        if R(35684.)>0 & B^=na then B=B+1;            else if R(35684.)>-4 & R(35684.)<0 then B=na;
        if R(35685.)>0 & B^=na then B=B+1;            else if R(35685.)>-4 & R(35685.)<0 then B=na;
        if R(35686.)>0 & B^=na then B=B+1;            else if R(35686.)>-4 & R(35686.)<0 then B=na;
        if R(35687.)>0 & B^=na then B=B+1;            else if R(35687.)>-4 & R(35687.)<0 then B=na;
        if R(35688.)>0 & B^=na then B=B+1;            else if R(35688.)>-4 & R(35688.)<0 then B=na;
      if R(35689.)<0 then PUBLIC(13)=R(35689.); else if B<=0 then PUBLIC(13)=-3; else
PUBLIC(13)=R(35689.)*B;
    end;
    if R(35695.)=-4 then ED(13)=0;                    else ED(13)=R(35695.);
    if R(35696.)=-4 then EDSS(13)=0;                  else EDSS(13)=R(35696.);
    if R(35698.)=-4 then VET(13)=0;                   else VET(13)=R(35698.);
    if R(35702.)=-4 then OTHER(13)=0;                 else OTHER(13)=R(35702.);
    if VET(13)>0 & OTHER(13)=VET(13) then OTHER(13)=0;
    if R(35712.)=-4 then RELWEL(13)=0;                else RELWEL(13)=R(35712.);
    if R(35714.)=-4 then RELREG(13)=0;                else RELREG(13)=R(35714.);
    FAMILY(13)=FAMSZ91; /* To create family size (i.e. FAMSZ91) search thru the household
            enumeratio.n increment family size if the relationship to the youth is a relative. Do not increase
            family size if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
   DCL  COMPONENT(19) fixed DEC(9);
   do I=13;
     MN(I,1)=0; MN(I,2)=0; MN(I,3)=0;  FAMINC(I)=0; INCOME(I)=-3;
     COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
     COMPONENT(4)=WPSS(I);         COMPONENT(5)=SEI(I);           COMPONENT(6)=SEIS(I);
     COMPONENT(7)=UI(I);           COMPONENT(8)=UIS(I);
                                   COMPONENT(9)=ALIM(I);
     COMPONENT(10)=CHSP(I);        COMPONENT(11)=AFDC(I);
                                   COMPONENT(13)=PUBLIC(I);
     COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);
                                   COMPONENT(15)=VET(I);
     COMPONENT(16)=OTHER(I);       COMPONENT(17)=RELWEL(I);
     COMPONENT(18)=RELREG(I);      COMPONENT(19)=FOOD(I);
     if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
     end;
     else do;
         do K=1 to 19;
           if COMPONENT(K)>-4 then do;
             if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
             else FAMINC(I)=FAMINC(I)+COMPONENT(K);
           end;
```

```
          end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
      if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then
CPS(I)=0;
         else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
     end;
    end;
  end;
  do I=1 to 3; if MN(13,I)>0 then MN(13,I)=1; end;
  if SUMN(13)>0 then SUMN(13)=1;
  INCOME(13)=R(36561.)             CPS(13)=R(36562.)          LEVEL(13)=R(36563.)
```

```
/* 1992 VARIABLES */
    do I=14; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
       FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
       do J=1 to 3; MN(I,J)=-4; end;
       OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
       do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
       UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
   end;
   TABLE(14,1,1)=6810; TABLE(14,2,1)=8500; TABLE(14,3,1)=7830;
   do J=2 to 15; TABLE(14,1,J)=TABLE(14,1,J-1)+2380; TABLE(14,2,J)=TABLE(14,2,J-1)+2980;
     TABLE(14,3,J)=TABLE(14,3,J-1)+2740;
   end;
   S(14)=1; if STATE92=2 then S(14)=2;                 else if STATE92=15 then S(14)=3;
   if R(38970.)=-4 then MIL(14)=0;                     else MIL(14)=R(38970.);
   if R(38971.)=-4 then WPS(14)=0;                     else WPS(14)=R(38971.);
   if R(38974.)=-4 then SEI(14)=0;                     else SEI(14)=R(38974.);
   if R(38988.)=-4 & R(38989.)=-4 then UI(14)=0;
   else if R(38988.)>0 & R(38989.)>0 then UI(14)=R(38988.) * R(38989.);
   else if R(38989.)<0 then UI(14)=R(38989.);          else UI(14)=-3;
   if R(38992.)=-4 then MILS(14)=0;                    else MILS(14)=R(38992.);
   if R(38993.)=-4 then WPSS(14)=0;                    else WPSS(14)=R(38993.);
   if R(38996.)=-4 then SEIS(14)=0;                    else SEIS(14)=R(38996.);
   if R(39010.)=-4 & R(39011.)=-4 then UIS(14)=0;
   else if R(39010.)>0 & R(39011.)>0 then UIS(14)=R(39010.) * R(39011.);
   else if R(39011.)<0 then UIS(14)=R(39011.);         else UIS(14)=-3;
   if R(39014.)=-4 then ALIM(14)=0;                    else ALIM(14)=R(39014.);
   if R(39016.)=-4 then CHSP(14)=0;                    else CHSP(14)=R(39016.);
   if R(39030.)=-4 then AFDC(14)=0;
   else do; B=0;
       if R(39018.)>0 & B^=na then B=B+1;              else if R(39018.)>-4 & R(39018.)<0 then B=na;
       if R(39019.)>0 & B^=na then B=B+1;              else if R(39019.)>-4 & R(39019.)<0 then B=na;
       if R(39020.)>0 & B^=na then B=B+1;              else if R(39020.)>-4 & R(39020.)<0 then B=na;
       if R(39021.)>0 & B^=na then B=B+1;              else if R(39021.)>-4 & R(39021.)<0 then B=na;
       if R(39022.)>0 & B^=na then B=B+1;              else if R(39022.)>-4 & R(39022.)<0 then B=na;
       if R(39023.)>0 & B^=na then B=B+1;              else if R(39023.)>-4 & R(39023.)<0 then B=na;
       if R(39024.)>0 & B^=na then B=B+1;              else if R(39024.)>-4 & R(39024.)<0 then B=na;
       if R(39025.)>0 & B^=na then B=B+1;              else if R(39025.)>-4 & R(39025.)<0 then B=na;
       if R(39026.)>0 & B^=na then B=B+1;              else if R(39026.)>-4 & R(39026.)<0 then B=na;
       if R(39027.)>0 & B^=na then B=B+1;              else if R(39027.)>-4 & R(39027.)<0 then B=na;
       if R(39028.)>0 & B^=na then B=B+1;              else if R(39028.)>-4 & R(39028.)<0 then B=na;
       if R(39029.)>0 & B^=na then B=B+1;              else if R(39029.)>-4 & R(39029.)<0 then B=na;
     if R(39030.)<0 then AFDC(14)=R(39030.); else if B<=0 then AFDC(14)=-3; else AFDC(14)=R(39030.)
* B;
   end;
   if R(39044.)=-4 then FOOD(14)=0;
   else do; B=0;
       if R(39032.)>0 & B^=na then B=B+1;              else if R(39032.)>-4 & R(39032.)<0 then B=na;
       if R(39033.)>0 & B^=na then B=B+1;              else if R(39033.)>-4 & R(39033.)<0 then B=na;
       if R(39034.)>0 & B^=na then B=B+1;              else if R(39034.)>-4 & R(39034.)<0 then B=na;
       if R(39035.)>0 & B^=na then B=B+1;              else if R(39035.)>-4 & R(39035.)<0 then B=na;
       if R(39036.)>0 & B^=na then B=B+1;              else if R(39036.)>-4 & R(39036.)<0 then B=na;
       if R(39037.)>0 & B^=na then B=B+1;              else if R(39037.)>-4 & R(39037.)<0 then B=na;
       if R(39038.)>0 & B^=na then B=B+1;              else if R(39038.)>-4 & R(39038.)<0 then B=na;
       if R(39039.)>0 & B^=na then B=B+1;              else if R(39039.)>-4 & R(39039.)<0 then B=na;
```

```
          if R(39040.)>0 & B^=na then B=B+1;                else if R(39040.)>-4 & R(39040.)<0 then B=na;
          if R(39041.)>0 & B^=na then B=B+1;                else if R(39041.)>-4 & R(39041.)<0 then B=na;
          if R(39042.)>0 & B^=na then B=B+1;                else if R(39042.)>-4 & R(39042.)<0 then B=na;
          if R(39043.)>0 & B^=na then B=B+1;                else if R(39043.)>-4 & R(39043.)<0 then B=na;
        if R(39044.)<0 then FOOD(14)=R(39044.); else if B<=0 then FOOD(14)=-3; else FOOD(14)=R(39044.)
* B;
     end;
     if R(39070.)=-4 then PUBLIC(14)=0;
     else do; B=0;
          if R(39058.)>0 & B^=na then B=B+1;                else if R(39058.)>-4 & R(39058.)<0 then B=na;
          if R(39059.)>0 & B^=na then B=B+1;                else if R(39059.)>-4 & R(39059.)<0 then B=na;
          if R(39060.)>0 & B^=na then B=B+1;                else if R(39060.)>-4 & R(39060.)<0 then B=na;
          if R(39061.)>0 & B^=na then B=B+1;                else if R(39061.)>-4 & R(39061.)<0 then B=na;
          if R(39062.)>0 & B^=na then B=B+1;                else if R(39062.)>-4 & R(39062.)<0 then B=na;
          if R(39063.)>0 & B^=na then B=B+1;                else if R(39063.)>-4 & R(39063.)<0 then B=na;
          if R(39064.)>0 & B^=na then B=B+1;                else if R(39064.)>-4 & R(39064.)<0 then B=na;
          if R(39065.)>0 & B^=na then B=B+1;                else if R(39065.)>-4 & R(39065.)<0 then B=na;
          if R(39066.)>0 & B^=na then B=B+1;                else if R(39066.)>-4 & R(39066.)<0 then B=na;
          if R(39067.)>0 & B^=na then B=B+1;                else if R(39067.)>-4 & R(39067.)<0 then B=na;
          if R(39068.)>0 & B^=na then B=B+1;                else if R(39068.)>-4 & R(39068.)<0 then B=na;
          if R(39069.)>0 & B^=na then B=B+1;                else if R(39069.)>-4 & R(39069.)<0 then B=na;
        if R(39070.)<0 then PUBLIC(14)=R(39070.); else if B<=0 then PUBLIC(14)=-3; else
PUBLIC(14)=R(39070.)*B;
     end;
     if R(39076.)=-4 then ED(14)=0;                         else ED(14)=R(39076.);
     if R(39077.)=-4 then EDSS(14)=0;                       else EDSS(14)=R(39077.);
     if R(39079.)=-4 then VET(14)=0;                        else VET(14)=R(39079.);
     if R(39083.)=-4 then OTHER(14)=0;                      else OTHER(14)=R(39083.);
     if VET(14)>0 & OTHER(14)=VET(14) then OTHER(14)=0;
     if R(39093.)=-4 then RELWEL(14)=0;                     else RELWEL(14)=R(39093.);
     if R(39095.)=-4 then RELREG(14)=0;                     else RELREG(14)=R(39095.);
     FAMILY(14)=FAMSZ92; /* To create family size (i.e. famsz92) search thru the household enumeration.
              Increment family size if the relationship to the youth is a relative. Do not increase family size, if
              the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
    DCL COMPONENT(19) fixed DEC(9);
    do I=13;
       MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
       COMPONENT(1)=MIL(I);            COMPONENT(2)=MILS(I);            COMPONENT(3)=WPS(I);
       COMPONENT(4)=WPSS(I);           COMPONENT(5)=SEI(I);             COMPONENT(6)=SEIS(I);
       COMPONENT(7)=UI(I);             COMPONENT(8)=UIS(I);
                                       COMPONENT(9)=ALIM(I);
       COMPONENT(10)=CHSP(I);          COMPONENT(11)=AFDC(I);
                                       COMPONENT(12)=PUBLIC(I);
       COMPONENT(13)=ED(I);            COMPONENT(14)=EDSS(I);
                                       COMPONENT(15)=VET(I);
       COMPONENT(16)=OTHER(I);         COMPONENT(17)=RELWEL(I);
       COMPONENT(18)=RELREG(I);        COMPONENT(19)=FOOD(I);
       if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
       else do; do K=1 to 19;
             if COMPONENT(K)>-4 then do;
               if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
               else FAMINC(I)=FAMINC(I)+COMPONENT(K);
             end; end;
         SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
         else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
```

```
 /* income trunc */  if INCOME(I) > 100000 then INCOME(I) = 833745;
        if FAMILY(I)>0 then do;
            LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then CPS(I)=0;
            else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
        end;
      end;
    end;
do I=1 to 3; if MN(14,I)>0 then MN(14,I)=1; end;
if SUMN(14)>0 then SUMN(14)=1;

INCOME(14)=R(40066.)                    CPS(14)=R(40067.)             LEVEL(14)=R(40068.)
```

/* 1993 VARIABLES */

```
   do I=15; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
      FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
      do J=1 to 3; MN(I,J)=-4; end;
      OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
      do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
      UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
   end;

/* The 1993 Poverty Income Guidelines are applicable for income received in calendar 1992.  Contact the          */
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,        */
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines.  Talk to Gordan Fisher or Joan            */
/* Turek-Brezina.  Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.               */

/* Several variables pertaining to recipiency from various government income programs were computed from       */
/* a number of different variables in 1993, due to the change to an event history format for data collection.  */
/* these variables are as follows:                          */
/* WKTOT92U = Total number of weeks R received unemployment compensation in calendar year 1992                  */
/* RCTOT92U = Total amount of unemployment compensation R received in calendar year 1992                        */
/* WKTOT92S = Total number of weeks R's spouse received unemployment compensation in calendar year
1992                                                                                                             */
/* RCTOT92U = Total amount of unemployment compensation R's spouse received in calendar year 1992               */
/* MOTOT92A = Total number of months R or spouse received AFDC payments in calendar year 1992                   */
/* RCTOT92A = Total amount of AFDC R or spouse received in calendar year 1992                                   */
/* MOTOT92F = Total number of months R or spouse received government food stamps in calendar year
1992                                                                                                             */
/* RCTOT92F = Total amount of government food stamps R or spouse received in calendar year 1992                 */
/* MOTOT92W = Total number of months R or spouse received other welfare or public assistance payments           */
/*                 in calendar year 1992                                                                        */
/* RCTOT92W = Total amount of other welfare or public assistance payments R or spouse received in               */
/*                 calendar year 1992                                                                           */

      TABLE(15,1,1)=7360; TABLE(15,2,1)=9200; TABLE(15,3,1)=8470;
      do J=2 to 15; TABLE(15,1,J)=TABLE(15,1,J-1)+2480; TABLE(15,2,J)=TABLE(15,2,J-1)+3100;
        TABLE(15,3,J)=TABLE(15,3,J-1)+2850;
      end;
      S(15)=1; if STATE93=2 then S(15)=2;                 else if STATE93=15 then S(15)=3;
       if R(42949.)=-4 then MIL(15)=0;                    else MIL(15)=R(42949.);
       if R(42951.)=-4 then WPS(15)=0;                    else WPS(15)=R(42951.);
       if R(42955.)=-4 then SEI(15)=0;                    else SEI(15)=R(42955.);
      if WKTOT92U=-4 & RCTOT92U=-4 then UI(15)=0;
      else if WKTOT92U>0 & RCTOT92U>0 then UI(15)=WKTOT92U * RCTOT92U;
      else if RCTOT92U<0 then UI(15)=RCTOT92U;            else UI(15)=-3;
       if R(43142.)=-4 then MILS(15)=0;                                 else MILS(15)=R(43142.);
       if R(43144.)=-4 then WPSS(15)=0;                   else WPSS(15)=R(43144.);
       if R(43149.)=-4 then SEIS(15)=0;                   else SEIS(15)=R(43149.);
      if WKTOT92S=-4 & RCTOT92S=-4 then UIS(15)=0;
      else if WKTOT92S>0 & RCTOT92S>0 then UIS(15)=WKTOT92S * RCTOT92S;
      else if RCTOT92S<0 then UIS(15)=RCTOT92S;                         else UIS(15)=-3;
       if R(43325.)=-4 then ALIM(15)=0;                                 else ALIM(15)=R(43325.);
       if R(43341.)=-4 then CHSP(15)=0;                   else CHSP(15)=R(43341.);
       if R(43351.)=-4 then CHSP(15)=0;                   else CHSP(15)=R(43351.);
       if RCTOT92A=-4 then AFDC(15)=0;
      else do;                   if RCTOT92A<0 then AFDC(15)=RCTOT92A;
```

```
                                   else if MOTOT92A<=0 then AFDC(15)=–3;
                                   else AFDC(15)=RCTOT92A * MOTOT92A;
        end;
        if RCTOT92F=-4 then FOOD(15)=0;
        else do;                   if RCTOT92F<0 then FOOD(15)=RCTOT92F;
                                   else if MOTOT92F<=0 then FOOD(15)=-3;
                                   else FOOD(15)=RCTOT92F * MOTOT92F;
        end;
        if RCTOT92W=–4 then PUBLIC(15)=0;
        else do;                   if RCTOT92W<0 then PUBLIC(15)=RCTOT92W;
                                   else if MOTOT92W<=0 then PUBLIC(15)=-3;
                                   else PUBLIC(15)=RCTOT92W * MOTOT92W;
        end;
        if R(43879.)=–4 then ED(15)=0;                  else ED(15)=R(43879.);
        if R(43881.)=–4 then EDSS(15)=0;                          else EDSS(15)=R(43881.);
        if R(43883.)=–4 then VET(15)=0;                 else VET(15)=R(43883.);
        if R(43888.)=–4 then OTHER(15)=0;               else OTHER(15)=R(43888.);
        if VET(15)>0 & OTHER(15)=VET(15) then OTHER(15)=0;
        if R(43900.)=–4 then RELWEL(15)=0;              else RELWEL(15)=R(43900.);
        if R(43903.)=–4 then RELREG(15)=0;              else RELREG(15)=R(43903.);
        FAMILY(15)=R(44176.); /* To create family size (i.e. R(44176.)) Search thru the household
            enumeration. Increment family size if the relationship to the youth is a relative. Do not increase
            family size if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
       DCL COMPONENT(19) fixed DEC(9);
       do I=15;
         MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
         COMPONENT(1)=MIL(I);        COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
         COMPONENT(4)=WPSS(I);       COMPONENT(5)=SEI(I);           COMPONENT(6)=SEIS(I);
         COMPONENT(7)=UI(I);         COMPONENT(8)=UIS(I);
                                     COMPONENT(9)=ALIM(I);
         COMPONENT(10)=CHSP(I);      COMPONENT(11)=AFDC(I);
                                     COMPONENT(12)=PUBLIC(I);
         COMPONENT(13)=ED(I);        COMPONENT(14)=EDSS(I);
                                     COMPONENT(15)=VET(I);
         COMPONENT(16)=OTHER(I);     COMPONENT(17)=RELWEL(I);
         COMPONENT(18)=RELREG(I);    COMPONENT(19)=FOOD(I);
         if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=–5; end;
         else do;
             do K=1 to 19;
                if COMPONENT(K)>-4 then do;
                  if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                  else FAMINC(I)=FAMINC(I)+COMPONENT(K);
                end;
              end;
           SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
           else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;

    /* income trunc  */ if INCOME(I) > 100000 then INCOME(I) = 160097;

         if FAMILY(I)>0 then do;
            LEVEL(I)=TABLE(I,S(I),FAMILY(I));
            if INCOME(I)>LEVEL(I) then CPS(I)=0;
            else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
            else CPS(I)=-3;
         end;
```

```
      end;
     end;
  do I=1 to 3; if MN(15,I)>0 then MN(15,I)=1; end;
  if SUMN(15)>0 then SUMN(15)=1;

INCOME(15)=R(44177.)
CPS(15)=R(44178.)
LEVEL(15)=R(44179.)
```

```
/* 1994 VARIABLES */

   do I=16; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
      FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
      do J=1 to 3; MN(I,J)=-4; end;
      OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
      do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
      UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
  end;

/* The 1994 Poverty Income Guidelines are applicable for income received in calendar 1993.  Contact the       */
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,     */
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines.  Talk to Gordan Fisher or Joan         */
/* Turek-Brezina.  Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.            */

/* Several variables pertaining to recipiency from various government income programs were computed from     */
/* a number of different variables in 1994, due to the change to an event history format for data collection. */
/* these variables are as follows:                                                                            */
/* WKTOT93U = Total number of weeks R received unemployment compensation in calendar year 1993                */
/* RCTOT93U = Total amount of unemployment compensation R received in calendar year 1993                      */
/* WKTOT93S = Total number of weeks R's spouse received unemployment compensation in calendar year
1993                                                                                                          */
/* RCTOT93U = Total amount of unemployment compensation R's spouse received in calendar year 1993             */
/* MOTOT93A = Total number of months R or spouse received AFDC payments in calendar year 1993                 */
/* RCTOT93A = Total amount of AFDC R or spouse received in calendar year 1993                                 */
/* MOTOT93F = Total number of months R or spouse received government food stamps in calendar year
1993                                                                                                          */
/* RCTOT93F = Total amount of government food stamps R or spouse received in calendar year 1993               */
/* MOTOT93W = Total number of months R or spouse received other welfare or public assistance payments         */
/*              in calendar year 1993                                                                         */
/* RCTOT93W = Total amount of other welfare or public assistance payments R or spouse received in             */
/*                calendar year 1993                                                                          */

      TABLE(16,1,1)=7360; TABLE(16,2,1)=9200; TABLE(16,3,1)=8470;
      do J=2 to 15; TABLE(16,1,J)=TABLE(16,1,J-1)+2480; TABLE(16,2,J)=TABLE(16,2,J-1)+3100;
        TABLE(16,3,J)=TABLE(16,3,J-1)+2850;
      end;
      S(16)=1; if STATE94=2 then S(16)=2;               else if STATE94=15 then S(16)=3;
       if R(49826)=-4 then MIL(16)=0;                   else MIL(16)=R(49826.);
       if R(49828.)=-4 then WPS(16)=0;                  else WPS(16)=R(49828.);
       if R(49832.)=-4 then SEI(16)=0;                  else SEI(16)=R(49832.);
      if WKTOT93U=-4 & RCTOT93U=-4 then UI(16)=0;
      else if WKTOT93U>0 & RCTOT93U>0 then UI(16)=WKTOT93U * RCTOT93U;
      else if RCTOT93U<0 then UI(16)=RCTOT93U;          else UI(16)=-3;
       if R(49958.)=-4 then MILS(16)=0;                                 else MILS(16)=R(49958.);
       if R(49960.)=-4 then WPSS(16)=0;                 else WPSS(16)=R(49960.);
       if R(49966.)=-4 then SEIS(16)=0;                 else SEIS(16)=R(49966.);
      if WKTOT93S=-4 & RCTOT93S=-4 then UIS(16)=0;
       else if WKTOT93S>0 & RCTOT93S>0 then UIS(16)=WKTOT93S * RCTOT93S;
      else if RCTOT93S<0 then UIS(16)=RCTOT93S;                         else UIS(16)=-3;
       if R(50096.)=-4 then ALIM(16)=0;                                 else ALIM(16)=R(50096.);
      if R(50119.)=-4 then CHSP(16)=0;                  else CHSP(16)=R(50119.);
       if R(50130.)=-4 then CHSP(16)=0;                 else CHSP(16)=R(50130.);
       if RCTOT93A=-4 then AFDC(16)=0;
      else do;                     if RCTOT93A<0 then AFDC(16)=RCTOT93A;
```

```
                                    else if MOTOT93A<=0 then AFDC(16)=–3;
                                    else AFDC(16)=RCTOT93A * MOTOT93A;
      end;
       if RCTOT93F=-4 then FOOD(16)=0;
      else do;                      if RCTOT93F<0 then FOOD(16)=RCTOT93F;
                                    else if MOTOT93F<=0 then FOOD(16)=-3;
                                    else FOOD(16)=RCTOT93F * MOTOT93F;
      end;
       if RCTOT93W=–4 then PUBLIC(16)=0;
      else do;                      if RCTOT93W<0 then PUBLIC(16)=RCTOT93W;
                                    else if MOTOT93W<=0 then PUBLIC(16)=-3;
                                    else PUBLIC(16)=RCTOT93W * MOTOT93W;
      end;
       if R(50441.)=–4 then ED(16)=0;                  else ED(16)=R(50441.);
       if R(50443.)=–4 then EDSS(16)=0;                          else EDSS(16)=R(50443.);
       if R(50445.)=–4 then VET(16)=0;                  else VET(16)=R(50445.);
       if R(50450.)=–4 then OTHER(16)=0;                else OTHER(16)=R(50450.);
       if VET(16)>0 & OTHER(16)=VET(16) then OTHER(16)=0;
       if R(50462.)=–4 then RELWEL(16)=0;               else RELWEL( 16)=R(50462.);
       if R(50465.)=–4 then RELREG(16)=0;               else RELREG(16)=R(50465.);
      FAMILY(16)=FAMSZ94; /* To create family size (i.e. FAMSZ94) search thru the household
          enumeration. Increment family size, if the relationship to the youth is a relative. Do not increase
          family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
     DCL COMPONENT(19) fixed DEC(9);
     do I=16;
       MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
       COMPONENT(1)=MIL(I);         COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
       COMPONENT(4)=WPSS(I);        COMPONENT(5)=SEI(I);           COMPONENT(6)=SEIS(I);
       COMPONENT(7)=UI(I);          COMPONENT(8)=UIS(I);
                                    COMPONENT(9)=ALIM(I);
       COMPONENT(10)=CHSP(I);       COMPONENT(11)=AFDC(I);
                                    COMPONENT(12)=PUBLIC(I);
       COMPONENT(13)=ED(I);         COMPONENT(14)=EDSS(I);
                                    COMPONENT(15)=VET(I);
        COMPONENT(16)=OTHER(I);     COMPONENT(17)=RELWEL(I);
       COMPONENT(18)=RELREG(I);     COMPONENT(19)=FOOD(I);
       if WEIGHT(I)=0 then do;
         INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
       end;
       else do;
           do K=1 to 19;
              if COMPONENT(K)>-4 then do;
                if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                else FAMINC(I)=FAMINC(I)+COMPONENT(K);
              end;
            end;
         SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
         if SUMN(I)=0 then INCOME(I)=FAMINC(I);
         else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;

   /* income trunc  */     /*   if INCOME(I) > 100000 then INCOME(I) = 500000; */

       if FAMILY(I)>0 then do;
          LEVEL(I)=TABLE(I,S(I),FAMILY(I));
          if INCOME(I)>LEVEL(I) then CPS(I)=0;
```

```
              else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
              else CPS(I)=-3;
           end;
         end;
       end;
  do I=1 to 3; if MN(16,I)>0 then MN(16,I)=1; end;
  if SUMN(16)>0 then SUMN(16)=1;
```

INCOME(16)=R(50807.) CPS(16)=R(50808.)   LEVEL(16)=R(50809.)

```
/* 1996 VARIABLES */

   do I=17; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
      FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
      do J=1 to 3; MN(I,J)=-4; end;
      OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4;
SUMN(I)=-4;
      do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
      UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
  end;

/* The 1996 Poverty Income Guidelines are applicable for income received in calendar 1995.  Contact the        */
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,       */
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines.  Talk to Gordan Fisher or Joan           */
/* Turek-Brezina.  Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.              */

/* Several variables pertaining to recipiency from various government income programs were computed from      */
/* a number of different variables in 1994, due to the change to an event history format for data collection. */
/* these variables are as follows:                                                                              */
/* WKTOT95U = Total number of weeks R received unemployment compensation in calendar year 1995                  */
/* RCTOT95U = Total amount of unemployment compensation R received in calendar year 1995                        */
/* WKTOT95S = Total number of weeks R's spouse received unemployment compensation in calendar year
1995                                                                                                            */
/* RCTOT95U = Total amount of unemployment compensation R's spouse received in calendar year 1995               */
/* MOTOT95A = Total number of months R or spouse received AFDC payments in calendar year 1995                   */
/* RCTOT95A = Total amount of AFDC R or spouse received in calendar year 1995                                   */
/* MOTOT95F = Total number of months R or spouse received government food stamps in calendar year
1995                                                                                                            */
/* RCTOT95F = Total amount of government food stamps R or spouse received in calendar year 1995                 */
/* MOTOT95W = Total number of months R or spouse received other welfare or public assistance payments           */
/*                    in calendar year 1995                                                                     */
/* RCTOT95W = Total amount of other welfare or public assistance payments R or spouse received in               */
/*                  calendar year 1995                                                                          */

      TABLE(17,1,1)=7470; TABLE(17,2,1)=9340; TABLE(17,3,1)=8610;
      do J=2 to 15; TABLE(17,1,J)=TABLE(17,1,J-1)+2560; TABLE(17,2,J)=TABLE(17,2,J-1)+3200;
        TABLE(17,3,J)=TABLE(17,3,J-1)+2940;
      end;
      S(17)=1; if STATE96=2 then S(17)=2;              else if STATE96=15 then S(17)=3;
       if R(56260.)=-4 then MIL(17)=0;                  else MIL(17)=R(56260.);
       if R(56262.)=-4 then WPS(17)=0;                  else WPS(17)=R(56262.);
       if R(56266.)=-4 then SEI(17)=0;                  else SEI(17)=R(56266.);
      if WKTOT95U=-4 & RCTOT95U=-4 then UI(17)=0;
      else if WKTOT95U>0 & RCTOT95U>0 then UI(17)=WKTOT95U * RCTOT95U;
      else if RCTOT95U<0 then UI(17)=RCTOT95U;         else UI(17)=-3;
       if Q13_16=-4 then MILS(17)=0;                    else MILS(17)= Q13_16;
       if R(56508.)=-4 then WPSS(17)=0;                 else WPSS(17)=R(56508.);
       if R(56514.)=-4 then SEIS(17)=0;                 else SEIS(17)=R(56514.);
      if WKTOT95S=-4 & RCTOT95S=-4 then UIS(17)=0;
      else if WKTOT95S>0 & RCTOT95S>0 then UIS(17)=WKTOT95S * RCTOT95S;
      else if RCTOT95S<0 then UIS(17)=RCTOT95S;                   else UIS(17)=-3;
       if Q13_32=-4 then ALIM(17)=0;                    else ALIM(17)= Q13_32;
       if Q13_33I=-4 then CHSP(17)=0;                   else CHSP(17)= Q13_33I;
       if Q13_33M=-4 then CHSP(17)=0;                   else CHSP(17)= Q13_33M;
       if RCTOT95A=-4 then AFDC(17)=0;
      else do;                    if RCTOT95A<0 then AFDC(17)=RCTOT95A;
```

```
                              else if MOTOT95A<=0 then AFDC(17)=-3;
                              else AFDC(17)=RCTOT95A * MOTOT95A;
     end;
      if RCTOT95F=-4 then FOOD(17)=0;
     else do;                 if RCTOT95F<0 then FOOD(17)=RCTOT95F;
                              else if MOTOT95F<=0 then FOOD(17)=-3;
                              else FOOD(17)=RCTOT95F * MOTOT95F;
     end;
      if RCTOT95W=-4 then PUBLIC(17)=0;
     else do;                 if RCTOT95W<0 then PUBLIC(17)=RCTOT95W;
                              else if MOTOT95W<=0 then PUBLIC(17)=-3;
                              else PUBLIC(17)=RCTOT95W * MOTOT95W;
     end;
      if R(57255.)=-4 then ED(17)=0;                 else ED(17)=R(57255.);
      if R(57257.)=-4 then EDSS(17)=0;                        else EDSS(17)=R(57257.);
      if Q13_70=-4 then VET(17)=0;                   else VET(17)= Q13_70;
      if R(57264.)=-4 then OTHER(17)=0;              else OTHER(17)=R(57264.);
      if VET(17)>0 & OTHER(17)=VET(17) then OTHER(17)=0;
      if Q13_92=-4 then RELWEL(17)=0;                else RELWEL(17)= Q13_92;
      if Q13_92C=-4 then RELREG(17)=0;               else RELREG(17)= Q13_92C;
      FAMILY(17)=FAMSZ96; /* To create family size (i.e. FAMSZ96) search thru the household
         enumeration. Increment family size, if the relationship to the youth is a relative. Do not increase
         family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
     DCL COMPONENT(19) fixed DEC(9);
     do I=17;
       MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
       COMPONENT(1)=MIL(I);         COMPONENT(2)=MILS(I);         COMPONENT(3)=WPS(I);
       COMPONENT(4)=WPSS(I);        COMPONENT(5)=SEI(I);          COMPONENT(6)=SEIS(I);
       COMPONENT(7)=UI(I);          COMPONENT(8)=UIS(I);
                                    COMPONENT(9)=ALIM(I);
       COMPONENT(10)=CHSP(I);       COMPONENT(11)=AFDC(I);
                                    COMPONENT(12)=PUBLIC(I);
       COMPONENT(13)=ED(I);         COMPONENT(14)=EDSS(I);
                                    COMPONENT(15)=VET(I);
       COMPONENT(16)=OTHER(I);      COMPONENT(17)=RELWEL(I);
       COMPONENT(18)=RELREG(I);     COMPONENT(19)=FOOD(I);
       if WEIGHT(I)=0 then do;
         INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
       end;
       else do;
           do K=1 to 19;
             if COMPONENT(K)>-4 then do;
               if COMPONENT(K)<0 then
MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
               else FAMINC(I)=FAMINC(I)+COMPONENT(K);
             end;
           end;
          SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
          if SUMN(I)=0 then INCOME(I)=FAMINC(I);
          else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;

  /* income trunc  */    /*   if INCOME(I) > 100000 then INCOME(I) = 500000; */

       if FAMILY(I)>0 then do;
          LEVEL(I)=TABLE(I,S(I),FAMILY(I));
          if INCOME(I)>LEVEL(I) then CPS(I)=0;
          else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
          else CPS(I)=-3;
        end;
       end;
     end;
  do I=1 to 3; if MN(17,I)>0 then MN(17,I)=1; end;
  if SUMN(17)>0 then SUMN(17)=1;

INCOME(17)=R(51660.)CPS(17)=R(51661.)          LEVEL(17)=R(51662.)
```

/* 1998 VARIABLES */

/*programming for the 1998 variable TNFI is done in SPSS.   Only question numbers are listed.*/

```
compute AFDC_CUR=-4  compute CHSP_CUR=-4  compute CHSPS_C=-4
compute CPS_CUR=-4   compute ED_CUR=-4    compute EDSS_CUR=-4
compute FAMILY_C=-4  compute FAMINC_C=-4  compute FOOD_CUR=-4
compute INCOME_C=-4  compute LEVEL_C=-4   compute MIL_CUR=-4
compute MILS_CUR=-4  compute MN1701=-4    compute MN1702=-4
compute MN1703=-4    compute OTHER_C=-4   compute SSI_CUR=-4
compute RELREG_C=-4  compute RELWEL_C=-4  compute S_CUR=-4
compute SEI_CUR=-4   compute SEIS_CUR=-4  compute SUMN_CUR=-4
compute UI_CUR=-4              compute UIS_CUR=-4      compute  VET_CUR=-4
compute WPS_CUR=-4   compute WPSS_CUR=-4

/* The 1997 Poverty Income Guidelines are applicable for income received in calendar 1997.  Contact the        */
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,      */
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines.  Talk to Gordan Fisher or Joan          */
/* Turek-Brezina.  Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.            */

/* Several variables pertaining to recipiency from various government income programs were computed from     */
/* a number of different variables in 1997, due to the change to an event history format for data collection.  */
/* these variables are as follows:                                                                             */
/* WKTOT97U = Total number of weeks R received unemployment compensation in calendar year 1997                */
/* RCTOT97U = Total amount of unemployment compensation R received in calendar year 1997                      */
/* WKTOT97S = Total number of weeks R's spouse received unemployment compensation in calendar year
1997                                                                                                           */
/* RCTOT97U = Total amount of unemployment compensation R's spouse received in calendar year 1997             */
/* MOTOT97A = Total number of months R or spouse received AFDC payments in calendar year 1997                 */
/* RCTOT97A = Total amount of AFDC R or spouse received in calendar year 1997                                 */
/* MOTOT97F = Total number of months R or spouse received government food stamps in calendar year
1997                                                                                                           */
/* RCTOT97F = Total amount of government food stamps R or spouse received in calendar year 1997               */
/* MOTOT97W = Total number of months R or spouse received other welfare or public assistance payments         */
/*                    in calendar year 1997                                                                    */
/* RCTOT97W = Total amount of other welfare or public assistance payments R or spouse received in             */
/*                  calendar year 1997                                                                         */

compute TABLE1=7890             compute TBL1_INC=2720
compute TABLE2=9870             compute TBL2_INC=3400
compute TABLE3=9070             compute TBL3_INC=3130

compute S_CUR=1                 do if (STATE98N = ‘ 2’)
compute S_CUR=2                 else if (STATE98N = ‘15’)
compute S_CUR=3                 end if

do if (Q13_3 = -4)              compute MIL_CUR=0
else                            compute MIL_CUR=Q13_3
end if

do if (Q13_5 = -4)              compute WPS_CUR=0
else                            compute WPS_CUR=Q13_5
end if

do if (Q13_9 = -4)              compute SEI_CUR=0
else                            compute SEI_CUR=Q13_9
```

```
end if

do if (WKTOT97U = -4 and RCTOT97U = -4)          compute UI_CUR=0
else if (WKTOT97U = -4 and RCTOT97U = 0)         compute UI_CUR=0
else if (WKTOT97U > 0 and RCTOT97U > 0)          compute UI_CUR=(WKTOT97U * RCTOT97U)
else if (WKTOT97U < 0)                           compute UI_CUR=WKTOT97U
else if (RCTOT97U < 0)                           compute UI_CUR=RCTOT97U
else                                             compute UI_CUR=-3
end if

do if (PAFLAGA EQ 1 or Q13_16 = -4)              compute MILS_CUR=0
else                                             compute MILS_CUR=Q13_16
end if

do if (PAFLAGA EQ 1 or Q13_18 = -4)              compute WPSS_CUR=0
else                                             compute WPSS_CUR=Q13_18
end if

do if (PAFLAGA EQ 1 or Q13_24 = -4)              compute SEIS_CUR=0
else                                             compute SEIS_CUR=Q13_24
end if

do if (PAFLAGA EQ 1 or (WKTOT97S = -4 and RCTOT97S = -4))  compute UIS_CUR=0
else if (PAFLAGA EQ 1 or (WKTOT97S = -4 and RCTOT97S = 0)) compute UIS_CUR=0
else if (WKTOT97S > 0 and RCTOT97S > 0)          compute UIS_CUR=(WKTOT97S * RCTOT97S)
else if (WKTOT97S < 0)                           compute UIS_CUR=WKTOT97S
else if (RCTOT97S < 0)                           compute UIS_CUR=RCTOT97S
else                                             compute UIS_CUR=-3
end if

do if (Q13_33I = -4)                             compute CHSP_CUR=0
else                                             compute CHSP_CUR=Q13_33I
end if

do if (PAFLAGA EQ 1 or Q13_33M = -4)             compute CHSPS_C=0
else                                             compute CHSPS_C=Q13_33M
end if

do if (RCTOT97A EQ -4 or RCTOT97A EQ 0)          compute AFDC_CUR=0
else if (MOTOT97A EQ -4 or MOTOT97A EQ 0)        compute AFDC_CUR=0
else if (RCTOT97A < 0)                           compute AFDC_CUR=RCTOT97A
else if (MOTOT97A < 0)                           compute AFDC_CUR=MOTOT97A
else if (MOTOT97A <= 0)                          compute AFDC_CUR=-3
else                                             compute AFDC_CUR=(RCTOT97A * MOTOT97A)
end if

do if (RCTOT97F EQ -4 or RCTOT97F EQ 0)          compute FOOD_CUR=0
else if (MOTOT97F EQ -4 or MOTOT97F EQ 0)        compute FOOD_CUR=0
else if (RCTOT97F < 0)                           compute FOOD_CUR=RCTOT97F
else if (MOTOT97F < 0)                           compute FOOD_CUR=MOTOT97F
else if (MOTOT97F <= 0)                          compute FOOD_CUR=-3
else                                             compute FOOD_CUR=(RCTOT97F * MOTOT97F)
end if

do if (RCTOT97W EQ -4 or RCTOT97W EQ 0)          compute SSI_CUR=0
else if (MOTOT97W EQ -4 or MOTOT97W EQ 0)        compute SSI_CUR=0
else if (RCTOT97W < 0)                         compute SSI_CUR=RCTOT97W
else if (MOTOT97W < 0)                         compute SSI_CUR=MOTOT97W
else if (MOTOT97W <= 0)                        compute SSI_CUR=-3
else                                           compute SSI_CUR=(RCTOT97W * MOTOT97W)
end if

do if (Q13_66 = -4)                   compute ED_CUR=0
else                                  compute ED_CUR=Q13_66
end if

do if (PAFLAGA EQ 1 or Q13_68 = -4)compute EDSS_CUR=0
else                                  compute EDSS_CUR=Q13_68
end if

do if (Q13_70A = -4)                  compute VET_CUR=0
else                                  compute VET_CUR=Q13_70A
end if

do if (Q13_75 = -4)                   compute OTHER_C=0
else                                  compute OTHER_C=Q13_75
end if

if (VET_CUR > 0 and OTHER_C = VET_CUR) OTHER_C=0

do if (Q13_92 = -4)                   compute RELWEL_C=0
else                                  compute RELWEL_C=Q13_92
end if

do if (Q13_92C = -4)                  compute RELREG_C=0
else                                  compute RELREG_C=Q13_92C
end if

compute FAMILY_C=FAMSZ98 /* To create family size (i.e. famsz98) search thru the household
enumeration. Increment family size, if the relationship to the youth is a relative. Do not increase family size, if
the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

compute MN1701=0                      compute MN1702=0
compute MN1703=0                      compute FAMINC_C=0
compute INCOME_C=-3                   compute COMPO1=MIL_CUR
compute COMPO2=MILS_CUR               compute COMPO3=WPS_CUR
compute COMPO4=WPSS_CUR               compute COMPO5=SEI_CUR
compute COMPO6=SEIS_CUR               compute COMPO7=UI_CUR
compute COMPO8=UIS_CUR                compute COMPO9=CHSP_CUR
compute COMPO10=AFDC_CUR              compute COMPO11=SSI_CUR
compute COMPO12=ED_CUR                compute COMPO13=EDSS_CUR
compute COMPO14=VET_CUR               compute COMPO15=OTHER_C
compute COMPO16=RELWEL_C              compute COMPO17=RELREG_C
compute COMPO18=FOOD_CUR              compute COMPO19=CHSPS_C

do if (WKTOT97U EQ -4 and RCTOT97U GT 0)       compute UI_CUR=-3          compute COMPO7=-3
else if (WKTOT97U LT 0 and WKTOT97U GT -4)     compute UI_CUR=-3          compute COMPO7=-3
end if

do if (WKTOT97S EQ -4 and RCTOT97S GT 0)       compute UIS_CUR=-3         compute COMPO8=-3
else if (WKTOT97S LT 0 and WKTOT97S GT -4)     compute UIS_CUR=-3         compute COMPO8=-3
end if
```

```
do if (MOTOT97A EQ -4 and RCTOT97A GT 0)        compute AFDC_CUR=-3     compute COMPO10=-3
else if (MOTOT97A LT 0 and MOTOT97A GT -4)      compute AFDC_CUR=-3     compute COMPO10=-3
end if

do if (MOTOT97F EQ -4 and RCTOT97F GT 0)        compute FOOD_CUR=-3     compute COMPO18=-3
else if (MOTOT97F LT 0 and MOTOT97F GT -4)      compute FOOD_CUR=-3     compute COMPO18=-3
end if

do if (MOTOT97W EQ -4 and RCTOT97W GT 0)        compute SSI_CUR=-3      compute COMPO11=-3
else if (MOTOT97W LT 0 and MOTOT97W GT -4)      compute SSI_CUR=-3      compute COMPO11=-3
end if

do repeat COMPO=COMPO1 to COMPO19
.  do if (COMPO > -4)
.    do if (COMPO < 0 and COMPO EQ -1)           .   compute MN1701=(MN1701+1)
.    else if (COMPO < 0 and COMPO EQ -2)         .   compute MN1702=(MN1702+1)
.    else if (COMPO < 0 and COMPO EQ -3)         .   compute MN1703=(MN1703+1)
.    else                                        .   compute FAMINC_C=(FAMINC_C + COMPO)
.    end if
.  end if
end repeat

do if (SUMN_CUR = 0)
compute INCOME_C=FAMINC_C
else
.  do if (MN1703 > 0)       .  compute INCOME_C=-3
.  else if (MN1702 > 0)     .  compute INCOME_C=-2
.  else if (MN1701 > 0)     .  compute INCOME_C=-1
.  end if
end if

if (PAFLAGA EQ -5) INCOME_C=-5

   /* INCOME TRUNC  */    /* if (INCOME_C >= 161400) INCOME_C=1022003 */

  /* POVERTY STATUS and LEVEL */
do if (FAMILY_C > 0 and S_CUR EQ 1)
compute LEVEL_C=(TABLE1 + (TBL1_INC * (FAMILY_C-1)))
else if (FAMILY_C > 0 and S_CUR EQ 2)
compute LEVEL_C=(TABLE2 + (TBL2_INC * (FAMILY_C-1)))
else if (FAMILY_C > 0 and S_CUR EQ 3)
compute LEVEL_C=(TABLE3 + (TBL3_INC * (FAMILY_C-1)))
end if

do if (INCOME_C > LEVEL_C)                                  compute CPS_CUR=0
else if (INCOME_C >= 0 and INCOME_C <= LEVEL_C)             compute CPS_CUR=1
else                                                        compute CPS_CUR=-3
end if

do repeat MN=MN1701 to MN1703
if (MN > 0) MN=1
end repeat PRINT

if (SUMN_CUR > 0) SUMN_CUR=1
if (PAFLAGA EQ -5) LEVEL_C=-5
if (PAFLAGA EQ -5) CPS_CUR=-5
```

/* 2000 VARIABLES */

```
/*programming for the 2000 variable TNFI is done in SPSS.   Only question numbers are listed.*/

compute afdc_cur=-4     compute chsp_cur=-4       compute chsps_c=-4        compute cps_cur=-4
compute ed_cur=-4       compute edss_cur=-4       compute family_c=-4       compute faminc_c=-4
compute food_cur=-4     compute income_c=-4       compute level_c=-4        compute mil_cur=-4
compute mils_cur=-4     compute mn1701=-4         compute mn1702=-4         compute mn1703=-4
compute other_c=-4      compute ssi_cur=-4        compute relreg_c=-4       compute relwel_c=-4
compute s_cur=-4        compute sei_cur=-4        compute seis_cur=-4       compute sumn_cur=-4
compute ui_cur=-4       compute uis_cur=-4        compute vet_cur=-4        compute wps_cur=-4
compute wpss_cur=-4     compute wkcmp_c=-4

/* The 1999 Poverty Income Guidelines are applicable for income received in calendar 1999.  Contact the      */
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,     */
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines.  Talk to Gordan Fisher or Joan         */
/* Turek-Brezina.  Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.            */

/* Several variables pertaining to recipiency from various government income programs were computed from     */
/* a number of different variables in 1999, due to the change to an event history format for data collection. */
/* these variables are as follows:                                                                             */
/* WKTOT99U = Total number of weeks R received unemployment compensation in calendar year 1999                 */
/* RCTOT99U = Total amount of unemployment compensation R received in calendar year 1999                       */
/* WKTOT99S = Total number of weeks R's spouse received unemployment compensation in calendar year
1999                                                                                                           */
/* RCTOT99U = Total amount of unemployment compensation R's spouse received in calendar year 1999              */
/* MOTOT99A = Total number of months R or spouse received AFDC payments in calendar year 1999                  */
/* RCTOT99A = Total amount of AFDC R or spouse received in calendar year 1999                                  */
/* MOTOT99F = Total number of months R or spouse received government food stamps in calendar year
1999                                                                                                           */
/* RCTOT99F = Total amount of government food stamps R or spouse received in calendar year 1999                */
/* MOTOT99W = Total number of months R or spouse received other welfare or public assistance payments          */
/*                   in calendar year 1999                                                                     */
/* RCTOT99W = Total amount of other welfare or public assistance payments R or spouse received in              */
/*                  calendar year 1999                                                                         */

compute table1=8240     compute tbl1_inc=2820
compute table2=10320    compute tbl2_inc=3520
compute table3=9490     compute tbl3_inc=3240

/* set state of residence by hand for HA and Alaska for movers */
/* ste_cd is the original code and norcids are listed for all who reported */
/* a move to HA or Alaska */
compute s_cur=1
do if ((ste_cd=2) or ((caseid=5671) or (caseid=11881)))               compute s_cur=2
else if ((ste_cd=15) or (caseid=1266))            compute s_cur=3             end if

do if (Q13_3 = -4)                                compute mil_cur=0
else                                              compute mil_cur=Q13_3       end if

do if (Q13_5 = -4)                                compute wps_cur=0
else                                              compute wps_cur=Q13_5       end if

/* new series to incorporate new farm & bus income qs */ /* norcids are listed for cases with 2 farms or
businesses */
```

```
compute fjt_inc=fjt_141
if (caseid eq 3187) fjt_inc=13000

do if (fjt_inc eq -4)                             compute farm_cur=0
else                                              compute farm_cur=fjt_inc                end if

compute bpjt_inc=bp_141                           if (caseid eq 2410) bpjt_inc=15000
                                                  if (caseid eq 6068) bpjt_inc=40000

do if (bpjt_inc eq -4)                            compute bus_cur=0
else                                              compute bus_cur=bpjt_inc                end if

compute fbr_inc=q13_132d
do if (fbr_inc eq -4)                             compute fbr_cur=0
else                                              compute fbr_cur=fbr_inc                 end if

do if (Q13_9 = -4)                                compute sei_cur=0
else                                              compute sei_cur=Q13_9                   end if
if (farm_cur gt 0) sei_cur=(sei_cur+farm_cur)
if (bus_cur gt 0) sei_cur=(sei_cur+bus_cur)
if (fbr_cur gt 0) sei_cur=(sei_cur+fbr_cur)

do if (wktot99u = -4 and rctot99u = -4)           compute ui_cur=0
else if (wktot99u = -4 and rctot99u = 0)          compute ui_cur=0
else if (wktot99u > 0 and rctot99u > 0)           compute ui_cur=rctot99u
else if (wktot99u < 0)                            compute ui_cur=wktot99u
else if (rctot99u < 0)                            compute ui_cur=rctot99u
else                                              compute ui_cur=-3                       end if

do if (paflaga eq 1 or Q13_16 = -4)               compute mils_cur=0
else                                              compute mils_cur=Q13_16                 end if

do if (paflaga eq 1 or Q13_18 = -4)               compute wpss_cur=0
else                                              compute wpss_cur=Q13_18                 end if


do if (paflaga eq 1 or Q13_24 = -4)               compute seis_cur=0
else                                              compute seis_cur=Q13_24                 end if

do if (paflaga eq 1 or (wktot99s = -4 and rctot99s = -4))          compute uis_cur=0
else if (paflaga eq 1 or (wktot99s = -4 and rctot99s = 0))         compute uis_cur=0
else if (wktot99s > 0 and rctot99s > 0)           compute uis_cur=rctot99s
else if (wktot99s < 0)                            compute uis_cur=wktot99s
else if (rctot99s < 0)                            compute uis_cur=rctot99s
else                                              compute uis_cur=-3          end if

do if (q13_31b = -4)                              compute wkcmp_c=0
else                                              compute wkcmp_c=q13_31b                 end if

do if (Q13_33I = -4)                              compute chsp_cur=0
else                                              compute chsp_cur=Q13_33I                end if

do if (paflaga eq 1 or Q13_33M = -4)              compute chsps_c=0
else                                              compute chsps_c=Q13_33M                 end if

do if (rctot99a eq -4 or rctot99a eq 0)           compute afdc_cur=0
```

```
else if (motot99a eq -4 or motot99a eq 0)                      compute afdc_cur=0
else if (rctot99a < 0)                            compute afdc_cur=rctot99a
else if (motot99a < 0)                            compute afdc_cur=motot99a
else if (motot99a <= 0)                           compute afdc_cur=-3
else                                              compute afdc_cur=rctot99a                end if

do if (rctot99f eq -4 or rctot99f eq 0)           compute food_cur=0
else if (motot99f eq -4 or motot99f eq 0)         compute food_cur=0
else if (rctot99f < 0)                            compute food_cur=rctot99f
else if (motot99f < 0)                            compute food_cur=motot99f
else if (motot99f <= 0)                           compute food_cur=-3
else                                              compute food_cur=rctot99f                end if

do if (rctot99w eq -4 or rctot99w eq 0)           compute ssi_cur=0
else if (motot99w eq -4 or motot99w eq 0)         compute ssi_cur=0
else if (rctot99w < 0)                            compute ssi_cur=rctot99w
else if (motot99w < 0)                            compute ssi_cur=motot99w
else if (motot99w <= 0)                           compute ssi_cur=-3
else                                              compute ssi_cur=rctot99w                 end if

do if (Q13_66 = -4)                               compute ed_cur=0
else                                              compute ed_cur=Q13_66                    end if

do if (paflaga eq 1 or Q13_68 = -4)               compute edss_cur=0
else                                              compute edss_cur=Q13_68                  end if

do if (q13_70 = -4)                               compute vet_cur=0
else                                              compute vet_cur=q13_70                   end if

do if (Q13_75 = -4)                               compute other_c=0
else                                              compute other_c=Q13_75                   end if

if (vet_cur > 0 and other_c = vet_cur) other_c=0

do if (Q13_92 = -4)                               compute relwel_c=0
else                                              compute relwel_c=Q13_92                  end if

do if (Q13_92C = -4)                              compute relreg_c=0
else                                              compute relreg_c=Q13_92C                 end if

compute family_c=famsz00 /* To create family size (i.e. FAMSZ00) search thru the household enumeration.
          Increment family size, if the relationship to the youth is a relative.  Do not increase family size, if
          the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

compute mn1701=0                    compute mn1702=0
compute mn1703=0                    compute faminc_c=0
compute income_c=-3                 compute compo1=mil_cur
compute compo2=mils_cur             compute compo3=wps_cur
compute compo4=wpss_cur             compute compo5=sei_cur
compute compo6=seis_cur             compute compo7=ui_cur
compute compo8=uis_cur              compute compo9=chsp_cur
compute compo10=afdc_cur            compute compo11=ssi_cur
compute compo12=ed_cur              compute compo13=edss_cur
compute compo14=vet_cur             compute compo15=other_c
compute compo16=relwel_c            compute compo17=relreg_c
compute compo18=food_cur            compute compo19=chsps_c
compute compo20=wkcmp_c

do if (wktot99u eq -4 and rctot99u gt 0)          compute ui_cur=-3          compute compo7=-3
else if (wktot99u lt 0 and wktot99u gt -4)                   compute ui_cur=-3          compute
compo7=-3
end if

do if (wktot99s eq -4 and rctot99s gt 0)          compute uis_cur=-3         compute compo8=-3
else if (wktot99s lt 0 and wktot99s gt -4)                   compute uis_cur=-3         compute
compo8=-3
end if

do if (motot99a eq -4 and rctot99a gt 0)          compute afdc_cur=-3        compute compo10=-3
else if (motot99a lt 0 and motot99a  gt -4)                  compute afdc_cur=-3        compute
compo10=-3
end if

do if (motot99f eq -4 and rctot99f gt 0)          compute food_cur=-3        compute compo18=-3
else if (motot99f lt 0 and motot99f gt -4)                   compute food_cur=-3        compute
compo18=-3
end if

do if (motot99w eq -4 and rctot99w gt 0)                     compute ssi_cur=-3         compute
compo11=-3
else if (motot99w lt 0 and motot99w gt -4)        compute ssi_cur=-3         compute compo11=-3
end if

do repeat compo=compo1 to compo20
.  do if (compo > -4)
.    do if (compo < 0 and compo eq -1)            .  compute mn1701=(mn1701+1)
.    else if (compo < 0 and compo eq -2)          .  compute mn1702=(mn1702+1)
.    else if (compo < 0 and compo eq -3)          .  compute mn1703=(mn1703+1)
.    else                                         .  compute faminc_c=(faminc_c + compo) .   end if
.  end if
end repeat print

compute sumn_cur=mn1701 + mn1702 + mn1703

do if (sumn_cur = 0)                              compute income_c=faminc_c
else
.   do if (mn1703 > 0)                            .  compute income_c=-3
.   else if (mn1702 > 0)                          .  compute income_c=-2
.   else if (mn1701 > 0)                          .  compute income_c=-1                .   end if
end if

if (paflaga eq -5) income_c=-5
if (caseid eq 9373) income_c=-3

  /* poverty status and level */
do if (family_c > 0 and s_cur eq 1)               compute level_c=(table1 + (tbl1_inc * (family_c-1)))
else if (family_c > 0 and s_cur eq 2)             compute level_c=(table2 + (tbl2_inc * (family_c-1)))
else if (family_c > 0 and s_cur eq 3)             compute level_c=(table3 + (tbl3_inc * (family_c-1)))
end if

do if (income_c > level_c)                                   compute cps_cur=0
else if (income_c >= 0 and income_c <= level_c)   compute cps_cur=1
else                                     compute cps_cur=-3                         end if

do repeat mn=mn1701 to mn1703
if (mn > 0) mn=1
end repeat print

if (sumn_cur > 0) sumn_cur=1
if (paflaga eq -5) level_c=-5
if (paflaga eq -5) cps_cur=-5
```

FAMILY POVERTY STATUS AND FAMILY POVERTY LEVEL VARIABLES

Family Poverty Status and Family Poverty Level variables are calculated using the Poverty Income Guidelines, updated yearly by the U.S. Department of Health and Human Services. The Poverty Income Guidelines use one person as a base and an increment is added to that figure for each single person increase in family size.

An NLSY79 respondent is determined to be in poverty if the family income for the last calendar year for the family size is below the Poverty Income Guidelines. The family poverty level, available for survey years 1979, 1987 and later survey years, is the actual federal Poverty Income Guideline amount for each NLSY79 respondent based on the family size for that survey year.

Three sets of guidelines exist, one each for 1) the contiguous 48 states, 2) Alaska, and 3) Hawaii. The figures for Alaska and Hawaii apply scaling factors of 1.25 and 1.15 (respectively) of the figure for the contiguous 48 states. These figures mean that the guidelines for Alaska are 1.25 times those for the contiguous 48 states (or 25% greater), and the guidelines for Hawaii are 1.15 times those of the contiguous 48 states (or 15% greater). In addition, a distinction was made between farm and non-farm families for the contiguous 48 states, Alaska, and Hawaii. Poverty Income Guidelines for farm families were defined as 85% of those for non-farm families. This farm and non-farm distinction was officially eliminated after 1982, and that change was reflected in the NLSY79 variable creation.

For survey years 1980 through 1986, CHRR staff projected poverty guidelines used in the NLSY79 household interview schedule and in the creation of poverty status. This was done because the guidelines were issued in February of the survey year, more than a month after all materials had to be printed for the field period.

For survey years 1979 and 1987 and after, changes in the household interview eliminated the need for projected poverty guidelines, and the actual Poverty Income Guidelines were used in the creation of the NLSY79 poverty status. Because of these changes, there is a higher missing rate in the later survey years on the family poverty status variable for these years.

1979 FAMILY POVERTY STATUS CREATION

The 1979 NLSY79 household interview schedules did not use the Poverty Income Guidelines. As a result, when the family poverty status and the family poverty level variables were created in 1990, the actual federal Poverty Income Guidelines were used.

Since the 1979 family poverty status variable was not created until 1990, a modified version of the family poverty status variable creation was used. First, if the created total net family income variable R(2179.) was greater than zero then income was compared to the Poverty Income Guideline amount for the respondent's family size, farm/non-farm distinction, and state of residence. Second, if the total net family income was less than zero, an abbreviated version of income was calculated using selected income variables (primarily earned income variables and other income variables that did not require computation of the number of months that the income was received). This abbreviated income amount was only used to determine if the family was not in poverty. Third, if the family poverty status was not computed based on either of the previous scenarios, then the screener income R(1916.10), if it was greater than zero, was compared to the Poverty Income Guideline amount for the respondent's screener family size, farm/non-farm distinction, and 1979 state of residence. A special flag variable (R2179.30) was created so that users could determine which income value was used to create the poverty status and the poverty level.

1980-2000 FAMILY POVERTY STATUS AND 1988-2000 FAMILY POVERTY LEVEL CREATION

1980-2000 NLSY79 Family Poverty Status and 1988-2000 Family Poverty Level variable creation is contained in the algorithm for Total Net Family Income, which is part of this appendix.

1980-1986 POVERTY INCOME GUIDELINE PROJECTIONS

The following is a brief description of the procedure used by NLSY79 staff to project the Poverty Income Guidelines that were used in the NLSY79 Family Poverty Status and Family Poverty Level variables. All procedures are applied to the various sets of guidelines and the income increments for those guidelines.

From 1980-1986, NLSY79 staff used the Consumer Price Index (CPI) from the two years preceding the actual NLSY79 survey year to project the Poverty Income Guidelines for the NLSY79 survey year. The following steps were taken:

1) The average CPI was calculated for each of the two years preceding the NLSY79 survey year (i.e. for the 1982 NLSY79, the average CPI for 1981 and 1980).

2) The average CPI for the most recent year was divided by the average CPI for the next most recent year (i.e. for the 1982 NLSY79, the 1981 average CPI was divided by the 1980 average CPI). The result is the most recent year's CPI as a percentage of the next most recent year's CPI (i.e. for the 1982 NLSY79, the 1981 CPI as a percentage of the 1980 CPI). This is a scaling factor for the amount of change in the average CPI between these two years. This number has been greater than 1 since 1979, because the CPI has always risen between years, not declined. The amount that the scaling figure is greater than 1 is the actual percentage of change between years (i.e. a scaling factor of 1.03561 between 1980 and 1981 indicates that the CPI for 1981 was 1.03561 times the CPI for 1980, or 3.561% greater.)

3) The Poverty Income Guidelines for the year preceding the actual NLSY79 survey year are multiplied by the scaling factor resulting from the operation in Step 2 (i.e. for the 1982 NLSY79, the 1981 Poverty Income Guidelines are multiplied by the scaling factor).

   This step requires an assumption that the degree of change in the CPI between the two years preceding the NLSY79 survey year is a reasonable estimate of the change that will occur between the year immediately previous and the actual NLSY79 survey year. The accuracy of these projections varies. All may be considered reasonable. However, some figures are more precise than others.

POVERTY INCOME GUIDELINE TABLES

Copies of the computed Poverty Income Guidelines 1980-1986 and the federal Poverty Income Guidelines for 1979, 1987 and subsequent years are part of this documentation.

1978 Poverty Income Guidelines (applicable for survey year 1979)

Non-Farm				Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 3,400	$ 4,270	$ 3,930	$ 2,910	$ 3,650	$ 3,350
2	4,500	5,640	5,910	3,840	4,810	4,420
3	5,600	7,010	6,450	4,770	5,970	5,490
4	6,700	8,380	7,710	5,700	7,130	6,560
5	7,800	9,750	8,970	6,630	8,290	7,630
6	8,900	11,120	10,230	7,560	9,450	8,700
Each add'l person	$ 1,100	$ 1,370	$ 1,260	$ 930	$ 1,160	$ 1,070

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

1979 Poverty Income Guidelines (applicable for survey year 1980)

Non-Farm				Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$3,770	$4,740	$4,350	$3,220	$4,030	$3,690
2	5,000	6,270	5,760	4,260	5,330	4,890
3	6,230	7,800	7,170	5,300	6,630	6,090
4	7,460	9,330	8,580	6,340	7,930	7,290
5	8,690	10,860	9,990	7,380	9,230	8,490
6	9,920	12,390	11,400	8,420	10,530	9,690
7	11,150	13,920	12,810	9,460	11,830	10,890
8	12,380	15,450	14,220	10,500	13,130	12,090
9	13,610	16,980	15,630	11,540	14,430	13,290
10	14,840	18,510	17,040	12,580	15,730	14,490
11	16,070	20,040	18,450	13,620	17,030	15,690
12	17,300	21,570	19,860	14,660	18,330	16,890
13	18,530	23,100	21,270	15,700	19,630	18,090
14	19,760	24,630	22,680	16,740	20,930	19,290
15	20,990	26,160	24,090	17,780	22,230	20,490
16	22,220	27,690	25,500	18,820	23,530	21,690
17	23,450	29,220	26,910	19,860	24,830	22,890
18	24,680	30,750	28,320	20,900	26,130	24,090
19	25,910	32,280	29,730	21,940	27,430	25,290
20	27,140	33,810	31,140	22,980	28,730	26,490
Each add'l person	$ 1230	$ 1530	$ 1410	$ 1040	$ 1300	$ 1200

1980 Poverty Income Guidelines (applicable for survey year 1981)

	Non-Farm			Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 4,320	$ 5,400	$ 4,970	$ 3,690	$ 4,660	$ 4,270
2	5,700	7,130	6,560	4,860	6,110	5,610
3	7,080	8,860	8,150	6,030	7,560	6,950
4	8,460	10,590	9,740	7,200	9,010	8,290
5	9,840	12,320	11,330	8,370	10,460	9,630
6	11,220	14,050	12,900	9,540	11,910	10,970
7	12,600	15,780	14,510	10,710	13,360	12,310
8	13,980	17,510	16,100	11,880	14,810	13,650
9	15,360	19,240	17,690	13,050	16,260	14,990
10	16,740	20,970	19,280	14,220	17,710	16,330
11	18,120	22,700	20,870	15,390	19,160	17,670
12	19,500	24,430	22,460	16,560	20,610	19,010
13	20,880	26,160	24,050	17,730	22,060	20,350
14	22,260	27,890	25,640	18,900	23,510	21,690
15	23,640	29,620	27,230	20,070	24,960	23,030
16	25,020	31,350	28,820	21,240	26,410	24,370
17	26,400	33,080	30,410	22,410	27,860	25,710
18	27,780	34,810	32,000	23,580	29,310	27,050
19	29,160	36,540	33,590	24,750	30,760	28,390
20	30,540	38,270	35,180	25,920	32,210	29,730
Each add’l person	$ 1,380	$ 1,730	$ 1,590	$ 1,170	$ 1,450	$ 1,340

1981 Poverty Income Guidelines (applicable for survey year 1982)

	Non-Farm			Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 4,320	$ 5,400	$ 4,970	$ 3,690	$ 4,660	$ 4,270
2	5,700	7,130	6,560	4,860	6,110	5,610
3	7,080	8,860	8,150	6,030	7,560	6,950
4	8,460	10,590	9,740	7,200	9,010	8,290
5	9,840	12,320	11,330	8,370	10,460	9,630
6	11,220	14,050	12,900	9,540	11,910	10,970
7	12,600	15,780	14,510	10,710	13,360	12,310
8	13,980	17,510	16,100	11,880	14,810	13,650
9	15,360	19,240	17,690	13,050	16,260	14,990
10	16,740	20,970	19,280	14,220	17,710	16,330
11	18,120	22,700	20,870	15,390	19,160	17,670
12	19,500	24,430	22,460	16,560	20,610	19,010
13	20,880	26,160	24,050	17,730	22,060	20,350
14	22,260	27,890	25,640	18,900	23,510	21,690
15	23,640	29,620	27,230	20,070	24,960	23,030
16	25,020	31,350	28,820	21,240	26,410	24,370
17	26,400	33,080	30,410	22,410	27,860	25,710
18	27,780	34,810	32,000	23,580	29,310	27,050
19	29,160	36,540	33,590	24,750	30,760	28,390
20	30,540	38,270	35,180	25,920	32,210	29,730
Each add’l person	$ 1,380	$ 1,730	$ 1,590	$ 1,170	$ 1,450	$ 1,340

1982 Poverty Income Guidelines (applicable for survey ye ar 1983)

	Non-Farm			Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 4,910	$ 6,160	$ 5,670	$ 4,200	$ 5,290	$ 4,850
2	6,530	8,180	7,530	5,570	6,990	6,420
3	8,150	10,200	9,390	6,940	8,690	7,990
4	9,770	12,220	11,250	8,310	10,390	9,560
5	11,390	14,240	13,110	9,680	12,090	11,130
6	13,010	16,260	14,970	11,050	13,790	12,700
7	14,630	18,280	16,830	12,420	15,490	14,270
8	16,250	20,300	18,690	13,790	17,190	15,840
9	17,870	22,320	20,550	15,160	18,890	17,410
10	19,490	24,340	22,410	16,530	20,590	18,980
11	21,110	26,360	24,270	17,900	22,290	20,550
12	22,730	28,380	26,130	19,270	23,990	22,120
13	24,350	30,400	27,990	20,640	25,690	23,690
14	25,970	32,420	29,850	22,010	27,390	25,260
15	27,590	34,440	31,710	23,380	29,090	26,830
Each add’l person	$ 1,620	$ 2,020	$ 1,860	$ 1,370	$ 1,700	$ 1,570

1983 Poverty Income Guidelines (applicable for survey year 1984)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,010	$ 6,280	$ 6,790
2	6,750	8,450	7,780
3	8,490	10,620	9,770
4	10,230	12,790	11,760
5	11,970	14,960	13,750
6	13,710	17,130	15,740
7	15,450	19,300	17,730
8	17,190	21,470	19,720
9	18,930	23,640	21,710
10	20,670	25,810	23,700
11	22,410	27,980	25,690
12	24,150	30,150	27,680
13	25,890	32,320	29,670
14	27,630	34,490	31,660
15	29,370	36,660	33,650
Each add’l person	$ 1,740	$ 2,170	$ 1,990

1984 Poverty Income Guidelines (applicable for survey year 1985)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,180	$ 6,500	$ 5,970
2	6,990	8,760	8,050
3	8,800	11,020	10,130
4	10,610	13,280	12,210
5	12,420	15,540	14,290
6	14,230	17,800	16,370
7	16,040	20,060	18,450
8	17,850	22,320	20,530
9	19,660	24,580	22,610
10	21,470	26,840	24,690
11	23,280	29,100	26,770
12	25,090	31,360	28,850
13	26,900	33,620	30,930
14	28,710	35,880	33,010
15	30,520	38,140	35,090
16	32,330	40,400	37,170
17	34,140	42,660	39,250
18	35,950	44,920	41,330
19	37,760	47,180	43,410
20	39,570	49,440	45,490
Each add’l person	$ 1,810	$ 2,260	$ 2,080

1985 Poverty Income Guidelines (applicable for survey year 1986)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,430	$ 6,790	$ 6,250
2	7,290	9,120	8,390
3	9,150	11,450	10,530
4	11,010	13,780	12,670
5	12,870	16,110	14,810
6	14,730	18,440	16,950
7	16,590	20,770	19,090
8	18,450	23,100	21,230
9	20,310	25,430	23,370
10	22,170	27,760	25,510
11	24,030	30,090	27,650
12	25,890	32,420	29,790
13	27,750	34,750	31,930
14	29,610	37,080	34,070
15	31,470	39,410	36,210
16	33,330	41,740	38,350
17	35,190	44,070	40,490
18	37,050	46,400	42,630
19	38,910	48,730	44,770
20	40,770	51,060	46,910
Each add’l person	$ 1,860	$ 2,330	$ 2,140

1986 Poverty Guidelines (applicable for survey year 1987)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,550	$ 6,680	$ 6,310
2	7,400	9,240	8,500
3	9,300	11,620	10,690
4	11,200	14,000	12,880
5	13,100	16,380	15,070
6	15,000	18,760	17,260
7	16,900	21,140	19,450
8	18,800	23,520	21,640
Each add’l person	$ 1,900	$ 2,380	$ 2,190

1987 Poverty Guidelines (applicable for survey year 1988)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,770	$ 7,210	$ 6,650
2	7,730	9,660	8,900
3	9,690	12,110	11,150
4	11,650	14,560	13,400
5	13,610	17,010	15,650
6	15,570	19,460	17,900
7	17,530	21,910	20,150
8	19,490	24,360	22,400
Each add’l person	$ 1,960	$ 2,450	$ 2,250

1988 Poverty Income Guidelines (applicable for survey year 1989)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 5,980	$ 7,480	$ 6,870
2	8,020	10,030	9,220
3	10,060	12,580	11,570
4	12,100	15,130	13,920
5	14,140	17,680	16,270
6	16,180	20,230	18,620
7	18,220	22,780	20,970
8	20,260	25,330	23,320
Each add’l person	$ 2,040	$ 2,550	$ 2,350

1989 Poverty Income Guidelines (applicable for survey year 1990)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 6,280	$ 7,840	$ 7,230
2	8,420	10,520	9,690
3	10,560	13,200	12,150
4	12,700	15,880	14,610
5	14,840	18,560	17,070
6	16,980	21,240	19,530
7	19,120	23,920	21,990
8	21,260	26,600	24,450
Each add’l person	$ 2,140	$ 2,680	$ 2,460

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any

of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

1990 Poverty Income Guidelines (applicable for survey year 1991)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 6,620	$ 8,290	$ 7,610
2	8,880	11,110	10,210
3	11,140	13,930	12,810
4	13,400	16,750	15,410
5	15,660	19,570	18,010
6	17,920	22,390	20,610
7	20,180	25,210	23,210
8	22,440	28,030	25,810
Each add’l person	$ 2,260	$ 2,820	$ 2,600

1991 Poverty Income Guidelines (applicable for survey year 1992)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 6,810	$ 8,500	$ 7,830
2	9,190	11,480	10,570
3	11,570	14,460	13,310
4	13,950	17,440	16,050
5	16,330	20,420	18,790
6	18,710	23,400	21,530
7	21,090	26,380	24,270
8	23,470	29,360	27,010
Each add’l person	$ 2,380	$ 2,980	$ 2,740

1992 Poverty Income Guidelines (applicable for survey year 1993)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 7,360	$ 9,200	$ 8,470
2	9,840	12,300	11,320
3	12,320	15,400	14,170
4	14,800	18,500	17,020
5	17,280	21,600	19,870
6	19,760	24,700	22,720
7	22,240	27,800	25,570
8	24,720	30,900	28,420
Each add’l person	$ 2,480	$ 3,100	$ 2,850

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any

of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

1993 Poverty Income Guidelines (applicable for survey year 1994)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 7,360	$ 9,200	$ 8,470
2	9,840	12,300	11,320
3	12,320	15,400	14,170
4	14,800	18,500	17,020
5	17,280	21,600	19,870
6	19,760	24,700	22,720
7	22,240	27,800	25,570
8	24,720	30,900	28,420
Each add’l person	$ 2,480	$ 3,100	$ 2,850

1995 Poverty Income Guidelines (applicable for survey year 1996)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 7,470	$ 9,340	$ 8,610
2	10,030	12,540	11,550
3	12,590	15,740	14,490
4	15,150	18,940	17,430
5	17,710	22,140	20,370
6	20,270	25,340	23,310
7	22,830	28,540	26,250
8	25,390	31,740	29,190
Each add’l person	$ 2,560	$ 3,200	$ 2,940

1997 Poverty Income Guidelines (applicable for survey year 1998)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 7,890	$ 9,870	$ 9,070
2	10,610	13,270	12,200
3	13,330	16,670	15,330
4	16,050	20,070	18,460
5	18,770	23,470	21,590
6	21,490	26,870	24,720
7	24,210	30,270	27,850
8	26,930	33,670	30,980
Each add’l person	$ 2,720	$ 3,400	$ 3,130

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any

of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

1999 Poverty Income Guidelines (applicable for survey year 2000)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	$ 8,240	$10,320	$ 9490
2	11,060	13,840	12,730
3	13,880	17,360	15,970
4	16,700	20,880	19,210
5	19,520	24,400	22,450
6	22,340	27,920	25,690
7	25,160	31,440	28,930
8	27,980	34,960	32,170
Each add'l person	$ 2,820	$ 3,520	$ 3,240

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.