NLSY97 CODEBOOK SUPPLEMENT

MAIN FILE ROUND 3

Prepared for the
U.S. Department of Labor by

Center for Human Resource Research
The Ohio State University

Under contract with
National Opinion Research Center
University of Chicago

2001

NLSY97 Appendix 2:

Employment Variable Creation

Introduction

A number of the created employment variables use the same program as input. The program in this section is referred to throughout the employment variables. For example, to create the "Weeks Worked at Employee Job #x during 19xx" variables, survey staff first run the program below titled "emp_begin.sas" and then run the program included in the weeks worked section of this appendix.

EMP_BEGIN.SAS

This program calculates total weeks worked at each job for each respondent. It converts start and stop dates for jobs and within-job gaps to continuous week numbers, subtracts within-job gaps, and finally counts the total weeks worked. The variables listed here are those employed in any program that uses emp_begin.sas as input; they may not be used in this initial program.

Name in Program	Question Name on CD	Name in Program	Question Name on CD
pubid	PUBID	mgap2_2-mgap2_7	YEMP-102100.02.02-.07
r1int_d, _m, _y	YINF-900_D, _M, _Y (round 1)	mgap3_2-mgap3_4	YEMP-102100.03.02-.04
birthdy, birthmo, birthyr	KEY!BDATE_D, _M, _Y (round 1)	mgap4_2, mgap4_3	YEMP-102100.04.02, .03
r1uid1- r1uid7	YEMP_UID.01-.07 (round 1)	mgap5_2, mgap5_3	YEMP-102100.05.02, .03
r2int_d, _m, _y	YINTDATE~D, ~M, ~Y (round 2)	bgdy1_1-bgdy5_1	YEMP-102200.01.01~D-.05.01~D
r2uid1-r2uid9	YEMP_UID.01-.09 (round 2)	bgmo1_1-bgmo5_1	YEMP-102200.01.01~M-.05.01~M
r3int_d, _m, _y	YINTDATE~D, ~M, ~Y	bgyr1_1-bgyr5_1	YEMP-102200.01.01~Y-.05.01~Y
e200	YEMP-200A	bgdy1_2-bgdy1_5	YEMP-102400.01.02~D-.05~D
e303_1- e303_7	YEMP-303.01-.07	bgmo1_2-bgmo1_5	YEMP-102400.01.02~M-.05~M
e239011-e239019	YEMP-23901.01-.09	bgyr1_2-bgyr1_5	YEMP-102400.01.02~Y-.05~Y
e245011-e245019	YEMP-24501.01-.09	bgdy2_2-bgdy2_6	YEMP-102400.02.02~D-.06~D
e344021-e344025	YEMP-34402.01-.05	bgmo2_2-bgmo2_6	YEMP-102400.02.02~M-.06~M
e344031-e344033	YEMP-34403.01-.03	bgyr2_2-bgyr2_6	YEMP-102400.02.02~Y-.06~Y
e344281-e344285	YEMP-34428.01-.05	bgdy3_2, bgmo3_2, bgyr3_2	YEMP-102400.03.02~D, ~M, ~Y
e226101	YEMP-22610.01.02	bgdy3_3, bgmo3_3, bgyr3_3	YEMP-102400.03.03~D, ~M, ~Y
e226102	YEMP-22610.02.02	bgdy4_2, bgmo4_2, bgyr4_2	YEMP-102400.04.02~D, ~M, ~Y
e226112	YEMP-22611.02.03	bgdy5_2, bgmo5_2, bgyr5_2	YEMP-102400.05.02~D, ~M, ~Y
e226113	YEMP-22611.03.03	egdy1_1-egdy1_5	YEMP-102700.01.01~D-.05~D
e226114	YEMP-22611.04.03	egmo1_1-egmo1_5	YEMP-102700.01.01~M-.05~M
e379021-e379028	YEMP-37902.01-.08	egyr1_1-egyr1_5	YEMP-102700.01.01~Y-.05~Y
e379041-e379045	YEMP-37904.01-.05	egdy2_1-egdy2_6	YEMP-102700.02.01~D-.06~D
e380001-e380009	YEMP-38000.01-.09	egmo2_1-egmo2_6	YEMP-102700.02.01~M-.06~M
e38000f1-e38000f7	YEMP-38000F.01-.07	egyr2_1-egyr2_6	YEMP-102700.02.01~Y-.06~Y
e381031-e381035	YEMP-38103.01-.05	egdy3_1-egdy3_3	YEMP-102700.03.01~D-.03~D
e381051-e381053	YEMP-38105.01-.03	egmo3_1-egmo3_3	YEMP-102700.03.01~M-.03~M
e599011-e599017	YEMP-59901.01-.07	egyr3_1-egyr3_3	YEMP-102700.03.01~Y-.03~Y
e880001-e880007	YEMP-88000.01-.07	egdy4_1, egmo4_1, egyr4_1	YEMP-102700.04.01~D, ~M, ~Y
e885011-e885015	YEMP-88501.01-.05	egdy4_2, egmo4_2, egyr4_2	YEMP-102700.04.02~D, ~M, ~Y
e984021-e984026	YEMP-98402.01-.06	egdy5_1, egmo5_1, egyr5_1	YEMP-102700.05.01~D, ~M, ~Y
e984031-e984033	YEMP-98403.01-.03	egdy5_2, egmo5_2, egyr5_2	YEMP-102700.05.02~D, ~M, ~Y
e984291-e984293	YEMP-98429.01-.03	stardy1-stardy9	YEMP_STARTDATE.01~D-.09~D
e102311	YEMP-100231.01.02	starmo1-starmo9	YEMP_STARTDATE.01~M-.09~M
e102314	YEMP-100231.04.02	staryr1-staryr9	YEMP_STARTDATE.01~Y-.09~Y
e102321	YEMP-100232.01.03	stopdy1-stopdy9	YEMP_STOPDATE.01~D-.09~D
e1002561, e1002562	YEMP-100256.01, .02	stopmo1-stopmo9	YEMP_STOPDATE.01~M-.09~M
gap1-gap9	YEMP-101500.01-.09	stopyr1-stopyr9	YEMP_STOPDATE.01~Y-.09~Y
mgap1_2-mgap1_6	YEMP-102100.01.02-.06	r3uid1-r3uid9	YEMP_UID.01-.09

```
/***************** SECTION 1:  Convert birth week, week turned age 14, and week of ****************
****************** interview date into continuous weeks
**********************************************/

/** Calulate Age 14 year  **/          AGE14YR=birthyr+14;
/** Convert Age 14 Birthdate to week number  **//*Convert age14 month and day to total days (bbdays)*/
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;                           if birthmo=2 then bbdays=birthdy+31;
  if birthmo=3 then bbdays=birthdy+59;             if birthmo=4 then bbdays=birthdy+90;
  if birthmo=5 then bbdays=birthdy+120;            if birthmo=6 then bbdays=birthdy+151;
  if birthmo=7 then bbdays=birthdy+181;            if birthmo=8 then bbdays=birthdy+212;
  if birthmo=9 then bbdays=birthdy+243;            if birthmo=10 then bbdays=birthdy+273;
  if birthmo=11 then bbdays=birthdy+304;           if birthmo=12 then bbdays=birthdy+334;
end;

***Account for leap years;
if age14yr=1980 or age14yr=1984 or age14yr=1988 or age14yr=1992 or age14yr=1996 then do;
  if birthmo>0 and birthdy>0 then do;
    if birthmo=1 then bbdays=birthdy;              if birthmo=2 then bbdays=birthdy+31;
    if birthmo=3 then bbdays=birthdy+60;           if birthmo=4 then bbdays=birthdy+91;
    if birthmo=5 then bbdays=birthdy+121;          if birthmo=6 then bbdays=birthdy+152;
    if birthmo=7 then bbdays=birthdy+182;          if birthmo=8 then bbdays=birthdy+213;
    if birthmo=9 then bbdays=birthdy+244;          if birthmo=10 then bbdays=birthdy+274;
    if birthmo=11 then bbdays=birthdy+305;         if birthmo=12 then bbdays=birthdy+335;
  end;
end;

/***Convert days into week numbers.  Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of
days remaining in DEC})/7].  This formula applies throughout program.    ***/

/* Default age 14 week = 9999 */                age14wk=9999;
  if age14yr>0 and bbdays>0 then do;
    if age14yr=1980 then age14wk=ceil((bbdays+2)/7);        if age14yr=1981 then
age14wk=52+ceil((bbdays+4)/7);
    if age14yr=1982 then age14wk=104+ceil((bbdays+5)/7);   if age14yr=1983 then
age14wk=156+ceil((bbdays+6)/7);
    if age14yr=1984 then age14wk=209+ceil((bbdays)/7);     if age14yr=1985 then
age14wk=261+ceil((bbdays+2)/7);
    if age14yr=1986 then age14wk=313+ceil((bbdays+3)/7);   if age14yr=1987 then
age14wk=365+ceil((bbdays+4)/7);
    if age14yr=1988 then age14wk=417+ceil((bbdays+5)/7);   if age14yr=1989 then
age14wk=470+ceil((bbdays)/7);
    if age14yr=1990 then age14wk=522+ceil((bbdays+1)/7);   if age14yr=1991 then
age14wk=574+ceil((bbdays+2)/7);
    if age14yr=1992 then age14wk=626+ceil((bbdays+3)/7);   if age14yr=1993 then
age14wk=678+ceil((bbdays+5)/7);
    if age14yr=1994 then age14wk=730+ceil((bbdays+6)/7);   if age14yr=1995 then
age14wk=783+ceil((bbdays)/7);
    if age14yr=1996 then age14wk=835+ceil((bbdays+1)/7);   if age14yr=1997 then
age14wk=887+ceil((bbdays+3)/7);
    if age14yr=1998 then age14wk=939+ceil((bbdays+4)/7);   if age14yr=1999 then
age14wk=991+ceil((bbdays+5)/7);
    if age14yr=2000 then age14wk=1043+ceil((bbdays+6)/7);
  end;

/** Convert Birthdate to week number  **//* Default birthdate week=0 if birthdate < 12/30/79 */
birthwk=0;
  if birthyr>0 and bbdays>0 then do;
    if birthyr=1980 then birthwk=ceil((bbdays+2)/7);          if birthyr=1981 then birthwk=52+ceil((bbdays+4)/7);
    if birthyr=1982 then birthwk=104+ceil((bbdays+5)/7);    if birthyr=1983 then birthwk=156+ceil((bbdays+6)/7);
    if birthyr=1984 then birthwk=209+ceil((bbdays)/7);       if birthyr=1985 then birthwk=261+ceil((bbdays+2)/7);
    if birthyr=1986 then birthwk=313+ceil((bbdays+3)/7);    if birthyr=1987 then birthwk=365+ceil((bbdays+4)/7);
    if birthyr=1988 then birthwk=417+ceil((bbdays+5)/7);    if birthyr=1989 then birthwk=470+ceil((bbdays)/7);
    if birthyr=1990 then birthwk=522+ceil((bbdays+1)/7);    if birthyr=1991 then birthwk=574+ceil((bbdays+2)/7);
    if birthyr=1992 then birthwk=626+ceil((bbdays+3)/7);    if birthyr=1993 then birthwk=678+ceil((bbdays+5)/7);
    if birthyr=1994 then birthwk=730+ceil((bbdays+6)/7);    if birthyr=1995 then birthwk=783+ceil((bbdays)/7);
    if birthyr=1996 then birthwk=835+ceil((bbdays+1)/7);    if birthyr=1997 then birthwk=887+ceil((bbdays+3)/7);
    if birthyr=1998 then birthwk=939+ceil((bbdays+4)/7);    if birthyr=1999 then birthwk=991+ceil((bbdays+5)/7);
    if birthyr=2000 then birthyr=1043+ceil((bbdays+6)/7);
  end;

if e200=-5 then do; age14wk=-5; birthwk=-5; end;

/** Section 2:  Convert Round 3 Interview Date to week number  **/
***Convert interview month and day to total days (intdays);
  if r3int_m>0 and r3int_d>0 then do;
    if r3int_m=1 then intdays=r3int_d;              if r3int_m=2 then intdays=r3int_d+31;
    if r3int_m=3 then intdays=r3int_d+59;           if r3int_m=4 then intdays=r3int_d+90;
    if r3int_m=5 then intdays=r3int_d+120;          if r3int_m=6 then intdays=r3int_d+151;
    if r3int_m=7 then intdays=r3int_d+181;          if r3int_m=8 then intdays=r3int_d+212;
    if r3int_m=9 then intdays=r3int_d+243;          if r3int_m=10 then intdays=r3int_d+273;
    if r3int_m=11 then intdays=r3int_d+304;         if r3int_m=12 then intdays=r3int_d+334;
  end;
***Account for leap year 2000;
  if r3int_m>0 and r3int_d>0 and r3int_y=2000 then do;
    if r3int_m=1 then intdays=r3int_d;              if r3int_m=2 then intdays=r3int_d+31;
    if r3int_m=3 then intdays=r3int_d+60;           if r3int_m=4 then intdays=r3int_d+91;
    if r3int_m=5 then intdays=r3int_d+121;          if r3int_m=6 then intdays=r3int_d+152;
    if r3int_m=7 then intdays=r3int_d+182;          if r3int_m=8 then intdays=r3int_d+213;
    if r3int_m=9 then intdays=r3int_d+244;          if r3int_m=10 then intdays=r3int_d+274;
    if r3int_m=11 then intdays=r3int_d+305;         if r3int_m=12 then intdays=r3int_d+335;
  end;

/**Convert days into week numbers.  **/
/* Default interview week = 9999 */            intwk=9999;
  if r3int_y>0 and intdays>0 then do;
    if r3int_y=1999 then do; intwk=991+ceil((intdays+5)/7); end;
    if r3int_y=2000 then do; intwk=1043+ceil((intdays+6)/7); end;
  end;

/** Convert Date of Last Interview (Round 2) to week number  **/
***Convert interview month and day to total days (rd2days);
  if r2int_m>0 and r2int_d>0 then do;
    if r2int_m=1 then rd2days=r2int_d;              if r2int_m=2 then rd2days=r2int_d+31;
    if r2int_m=3 then rd2days=r2int_d+59;           if r2int_m=4 then rd2days=r2int_d+90;
    if r2int_m=5 then rd2days=r2int_d+120;          if r2int_m=6 then rd2days=r2int_d+151;
    if r2int_m=7 then rd2days=r2int_d+181;          if r2int_m=8 then rd2days=r2int_d+212;
    if r2int_m=9 then rd2days=r2int_d+243;          if r2int_m=10 then rd2days=r2int_d+273;
    if r2int_m=11 then rd2days=r2int_d+304;                  if r2int_m=12 then rd2days=r2int_d+334;
  end;
/* Default interview week = 9999 */            r2int=9999;
  if r2int_y>0 and rd2days>0 then do; if r2int_y=1998 then do; r2int=939+ceil((rd2days+4)/7); end; end;
  if r2int_y>0 and rd2days>0 then do; if r2int_y=1999 then do; r2int=991+ceil((rd2days+5)/7); end; end;

/**Convert Date of First Interview (Round 1) to week number **/
***Convert interview month and day to total days (rd2days);
  if r1int_m>0 and r1int_d>0 then do;
```

```
    if r1int_m=1 then rd1days=r1int_d;                  if r1int_m=2 then rd1days=r1int_d+31;
    if r1int_m=3 then rd1days=r1int_d+59;               if r1int_m=4 then rd1days=r1int_d+90;
    if r1int_m=5 then rd1days=r1int_d+120;              if r1int_m=6 then rd1days=r1int_d+151;
    if r1int_m=7 then rd1days=r1int_d+181;              if r1int_m=8 then rd1days=r1int_d+212;
    if r1int_m=9 then rd1days=r1int_d+243;              if r1int_m=10 then rd1days=r1int_d+273;
    if r1int_m=11 then rd1days=r1int_d+304;                     if r1int_m=12 then rd1days=r1int_d+334;
  end;

/* Hand edits carried over from Round 2 */
if (pubid=2 or  pubid=5615 or  pubid=5902) then do; r1int_y=1997; end;

/* Default interview week = 9999 */             r1int=9999;
  if r1int_y>0 and rd1days>0 then do; if r1int_y=1997 then do; r1int=887+ceil((rd1days+3)/7); end; end;
  if r1int_y>0 and rd1days>0 then do; if r1int_y=1998 then do; r1int=939+ceil((rd1days+4)/7); end; end;

/* Hand edits carried over from Round 2 */
if pubid in (471, 476) then r2int=990; if pubid=7315 then r1int=919;

/* Done to create a seamless flow between rounds */  intwk=intwk-1;
/* Round 2 non-interview case */                     if r2int_d=-5 then r2int=-5;
/* Round 3 non-interview case */                     if e200=-5 then intwk=-5;


/**************** Section 3:  Convert Start/Stop dates into NLSY97 week numbers ****************/
/*  This program reads in raw start and stop dates for each job (max=9) and converts them into NLSY97 week
numbers. Some start/stop DAYS and MONTHS have been imputed if missing. **/
/* ostartm represents the "old" start dates, used when start dates are updated to interview dates. */
array ostartm (i) ostarmo1-ostarmo9;            array ostartd (i) ostardy1-ostardy9;
array ostarty (i) ostaryr1-ostaryr9;            array ostopm (i) ostopmo1-ostopmo9;
array ostopd (i) ostopdy1-ostopdy9;             array ostopy (i) ostopyr1-ostopyr9;
array startm (i) starmo1-starmo9;               array startd (i) stardy1-stardy9;
array starty (i) staryr1-staryr9;               array stopm  (i) stopmo1-stopmo9;
array stopd  (i) stopdy1-stopdy9;               array stopy  (i) stopyr1-stopyr9;
array UID    (i) UID1-UID9;

array sttdays (i) sttday1-sttday9;              /* total days in that year from Jan 1 to startdate */
array stpdays (i) stpday1-stpday9;              /* total days in that year from Jan 1 to stopdate */
array startwk (i) starw1-starw9;
array stopwk  (i) stopw1-stopw9;
array srflag  (i) srflg1-srflg10;
array spflag  (i) spflg1-spflg10;
array uflag   (i) uflag1-uflag9;                /* uflag=1 when job startdate is updated */
array smofl   (i) smofl1-smofl9;                /* dummy equals 1 when a start month is imputed */
array emofl   (i) emofl1-emofl9;                /* dummy equals 1 when a stop month is imputed */

/* Hand edits.  If raw data does not match the data in this program then invalid skip. */
if pubid=231 then do; stopdy1=-3; stopmo1=-3; stopyr1=-3; end;
if pubid=657 then do; stopdy2=-3; stopmo2=-3; stopyr2=-3; end;
if pubid=2896 then do; stopdy2=-3; stopmo2=-3; stopyr2=-3; stardy1=-3; starmo1=-3; staryr1=-3; stardy2=-3;
          starmo2=-3; staryr2=-3; end;
if pubid=3102 then do; stopdy4=-3; stopmo4=-3; stopyr4=-3; end;
if pubid=3296 then do; stopdy2=-2; stopmo2=-2; stopyr2=-2; stardy2=-2; starmo2=-2; staryr2=-2; end;
if pubid=5055 then do; stopdy1=-2; stopmo1=-2; stopyr1=-2; end;
/* First job (UID=9701).dates are switched*/  if pubid=1960 then do; stardy1=27; starmo1=1; end;

/* Initialize smofl and emofl */
```

```
do i=1 to 9; smofl=0; emofl=0; end;

/* Define old start and stop dates */
do i=1 to 9; ostartm=startm; ostartd=startd; ostarty=starty; ostopm=stopm; ostopd=stopd; ostopy=stopy; end;

***Fill-in start/stop day for those missing;
do i=1 to 9; flag1=-4; flag2=-4; flag3=-4; end;

/* Impute missing start days to 1, missing stop days to 28, missing start months to 1 (Jan.) and missing stop months
to 12 (Dec.). */ /* Reset flag(1-3) to zero when the start year is valid */
do over starty;
  if starty>0 then do;
    flag1=0; flag2=0; flag3=0;
    if startm>0 and startd<=0 then do; startd=1; flag1=1; srflag=1; end;
    if startm<=0 and startd>0 then do; startm=1; flag2=1; srflag=1; smofl=1; end;
    if startm<=0 and startd<=0 then do; startm=1; startd=1; flag3=1; srflag=1; smofl=1; end;
  end;
end;

do i=1 to 9; flag4=-4; flag5=-4; flag6=-4; end;

do over stopy;
  if stopy>0 then do;
    flag4=0; flag5=0; flag6=0;
    if stopm>0 and stopd<=0 then do; stopd=28; flag4=1; spflag=1; end;
    if stopm<=0 and stopd>0 then do; stopm=12; flag5=1; spflag=1; emofl=1; end;
    if stopm<=0 and stopd<=0 then do; stopm=12; stopd=28; flag6=1; spflag=1; emofl=1; end;

/* Prevent imputed values for stop months and days from exceeding the Round3 interview date. */
    if stopy=r3int_y and stopm=r3int_m and stopd>r3int_d and spflag=1 then do; stopd=r3int_d; end;
    if stopy=r3int_y and stopm>r3int_m then do; stopm=r3int_m; stopd=r3int_d; end;
  end;
end;

/* Account for the cases where the respondent has a start date for a job reported in Round 1 or Round 2 that comes
before the Round 2 interview date. The idea is to only count the weeks employed from dli to today.  Then, the
activity from Rounds 1 & 2 will be added to the Round 3 activity to get the full event history.  To achieve this, all
jobs reported in Round 1 or 2 (UID's begin with 97 or 98) will have their startdates updated to the Round 2
interview date. The start week and stop week for these jobs will be counted the same. */
/* Job tenure information reported about the period prior to the last interview is not used (these data should have
bee reported by the respondent in the previous round but were overlooked.  These jobs will be included in job
specific created variables, but not year specific created variables. */

/* Initialize uflag */
do over uflag; uflag=0; end;
do i=1 to 9;
  if r2int>0 and uid>0 and startd>-4 and pubid not in (202, 1222, 1419, 1590, 2888, 3468, 3526, 4658, 5995, 7297,
         7465, 8827) then do;
    if starty=r2int_y and startm=r2int_m and startd<r2int_d then do; startd=r2int_d; uflag=1; end;
    if starty=r2int_y and startm<r2int_m then do; startm=r2int_m; startd=r2int_d; uflag=1; end;
    if starty<r2int_y then do; starty=r2int_y; startm=r2int_m; startd=r2int_d; uflag=1; end;
  end;
end;

/* For respondents who were not interviewed in Round 2 but were interviewed in Rounds 1 & 3, the start date is set
to the Round 1 interview date, instead of the Round 2 interview date. */
```

```
do i=1 to 9;
   if r2int=-5 and uid>0 and startd>-4 then do;
      if starty=r1int_y and startm=r1int_m and startd<r1int_d then do; startd=r1int_d; uflag=1; end;
      if starty=r1int_y and startm<r1int_m then do; startm=r1int_m; startd=r1int_d; uflag=1; end;
      if starty<r1int_y then do; starty=r1int_y; startm=r1int_m; startd=r1int_d; uflag=1; end;
   end;
end;

***Convert START month and day to total days;
do over startm;
   if startm>0 and startd>0 then do;
      if startm=1 then sttdays=startd;          if startm=2 then sttdays=startd+31;
      if startm=3 then sttdays=startd+59;       if startm=4 then sttdays=startd+90;
      if startm=5 then sttdays=startd+120;      if startm=6 then sttdays=startd+151;
      if startm=7 then sttdays=startd+181;      if startm=8 then sttdays=startd+212;
      if startm=9 then sttdays=startd+243;      if startm=10 then sttdays=startd+273;
      if startm=11 then sttdays=startd+304;     if startm=12 then sttdays=startd+334;
   end;
end;

***Account for leap years;
do over starty;
   if starty=1980 or starty=1984 or starty=1988 or starty=1992 or starty=1996 or starty=2000 then do;
      if startm>0 and startd>0 then do;
         if startm=1 then sttdays=startd;                     if startm=2 then sttdays=startd+31;
         if startm=3 then sttdays=startd+60;      if startm=4 then sttdays=startd+91;
         if startm=5 then sttdays=startd+121;     if startm=6 then sttdays=startd+152;
         if startm=7 then sttdays=startd+182;     if startm=8 then sttdays=startd+213;
         if startm=9 then sttdays=startd+244;     if startm=10 then sttdays=startd+274;
         if startm=11 then sttdays=startd+305;    if startm=12 then sttdays=startd+335;
      end;
   end;
end;

***Convert STOP month and day to total days;
do over stopm;
   if stopm>0 and stopd>0 then do;
      if stopm=1 then stpdays=stopd;            if stopm=2 then stpdays=stopd+31;
      if stopm=3 then stpdays=stopd+59;         if stopm=4 then stpdays=stopd+90;
      if stopm=5 then stpdays=stopd+120;        if stopm=6 then stpdays=stopd+151;
      if stopm=7 then stpdays=stopd+181;        if stopm=8 then stpdays=stopd+212;
      if stopm=9 then stpdays=stopd+243;        if stopm=10 then stpdays=stopd+273;
      if stopm=11 then stpdays=stopd+304;       if stopm=12 then stpdays=stopd+334;
   end;
end;

***Account for leap years;
do over stopy;
   if stopy=1980 or stopy=1984 or stopy=1988 or stopy=1992 or stopy=1996 or stopy=2000 then do;
      if stopm>0 and stopd>0 then do;
         if stopm=1 then stpdays=stopd;                       if stopm=2 then stpdays=stopd+31;
         if stopm=3 then stpdays=stopd+60;        if stopm=4 then stpdays=stopd+91;
         if stopm=5 then stpdays=stopd+121;       if stopm=6 then stpdays=stopd+152;
         if stopm=7 then stpdays=stopd+182;       if stopm=8 then stpdays=stopd+213;
         if stopm=9 then stpdays=stopd+244;       if stopm=10 then stpdays=stopd+274;
         if stopm=11 then stpdays=stopd+305;      if stopm=12 then stpdays=stopd+335;
```

```
    end;
  end;
end;

***Convert days into week numbers;
/* Create year flag variable */
array byear (i) byear1-byear9;        array eyear (i) eyear1-eyear9;
do over starty;
  if starty>0 and sttdays>0 then do;
    if starty=1980 then startwk=ceil((sttdays+2)/7);           if starty=1981 then startwk=52+ceil((sttdays+4)/7);
    if starty=1982 then startwk=104+ceil((sttdays+5)/7);       if starty=1983 then startwk=156+ceil((sttdays+6)/7);
    if starty=1984 then startwk=209+ceil((sttdays)/7);         if starty=1985 then startwk=261+ceil((sttdays+2)/7);
    if starty=1986 then startwk=313+ceil((sttdays+3)/7);       if starty=1987 then startwk=365+ceil((sttdays+4)/7);
    if starty=1988 then startwk=417+ceil((sttdays+5)/7);       if starty=1989 then startwk=470+ceil((sttdays)/7);
    if starty=1990 then startwk=522+ceil((sttdays+1)/7);       if starty=1991 then startwk=574+ceil((sttdays+2)/7);
    if starty=1992 then startwk=626+ceil((sttdays+3)/7);       if starty=1993 then startwk=678+ceil((sttdays+5)/7);
    if starty=1994 then startwk=730+ceil((sttdays+6)/7);       if starty=1995 then startwk=783+ceil((sttdays)/7);
    if starty=1996 then startwk=835+ceil((sttdays+1)/7);       if starty=1997 then startwk=887+ceil((sttdays+3)/7);
    if starty=1998 then startwk=939+ceil((sttdays+4)/7);       if starty=1999 then startwk=991+ceil((sttdays+5)/7);
    if starty=2000 then startwk=1043+ceil((sttdays+6)/7);
  end;
  if starty<0 and starty>-4 then do; startwk=-3; end;
  if -4<ostarty<0 then byear=1;
end;

do over stopy;
  if stopy>0 and stpdays>0 then do;
    if stopy=1980 then stopwk=ceil((stpdays+2)/7);             if stopy=1981 then stopwk=52+ceil((stpdays+4)/7);
    if stopy=1982 then stopwk=104+ceil((stpdays+5)/7);         if stopy=1983 then stopwk=156+ceil((stpdays+6)/7);
    if stopy=1984 then stopwk=209+ceil((stpdays)/7);           if stopy=1985 then stopwk=261+ceil((stpdays+2)/7);
    if stopy=1986 then stopwk=313+ceil((stpdays+3)/7);         if stopy=1987 then stopwk=365+ceil((stpdays+4)/7);
    if stopy=1988 then stopwk=417+ceil((stpdays+5)/7);         if stopy=1989 then stopwk=470+ceil((stpdays)/7);
    if stopy=1990 then stopwk=522+ceil((stpdays+1)/7);         if stopy=1991 then stopwk=574+ceil((stpdays+2)/7);
    if stopy=1992 then stopwk=626+ceil((stpdays+3)/7);         if stopy=1993 then; stopwk=678+ceil((stpdays+5)/7);
    if stopy=1994 then stopwk=730+ceil((stpdays+6)/7);         if stopy=1995 then stopwk=783+ceil((stpdays)/7);
    if stopy=1996 then stopwk=835+ceil((stpdays+1)/7);         if stopy=1997 then stopwk=887+ceil((stpdays+3)/7);
    if stopy=1998 then stopwk=939+ceil((stpdays+4)/7);         if stopy=1999 then stopwk=991+ceil((stpdays+5)/7);
    if stopy=2000 then stopwk=1043+ceil((stpdays+6)/7);
  end;
  if stopy<0 and stopy>-4 then do; stopwk=-3; end;
  if -4<ostarty<0 then eyear=1;
end;

/* Hand edits to fix problems in the raw data that are not yet fixed.  The dummy "ignore" is used later in this
program to identify these cases. */
if pubid=157 then UID4=199901; if pubid=175 then UID1=199901;
if pubid=3196 then UID1=199901;

/* To create a seamless list of information from Round2 to Round3, the following lines of code are included. Just as
the Round2 interview date was decreased by one, jobs that are worked up to the interview date are also decreased
by one. This is done so that the Round3 interview week will not be counted twice for respondents with a job during
that time. The same procedure was used for the Round1/Round2 seam. */
do over stopwk;
  if stopwk>0 and UID>0 then do; if stopwk>intwk then do; stopwk=intwk; end; end;
end;
```

```
/* The following lines considers jobs that begin the same week as the Round2 interview date. Since we are updating
the Round2 interview week by -1, we need to account for jobs that start in the same week or tenures of -1 will
result. */
do over startwk;
  if startwk>0 and UID>0 then do; if startwk>intwk then do; startwk=intwk; end; end;
end;

/* To account for respondents interviewed in Round1 but not Round2 */
if starmo1=-5 then do;
  do over startwk; startwk=-5; stopwk=-5; end;
end;

/* Tests for missing start/stop data */
miss=0;
do i=1 to 9;
  if (startm=-4 and stopm>-4) or (startm>-4 and stopm=-4) or (startd=-4 and startd>-4) or (startd>-4 and stopd=-4)
          or (starty=-4 and starty>-4) or (starty>-4 and stopy=-4) then do; miss=1; end;
end;

/* Correcting for imputed values that resulted in the start date being later than the stop date. In these cases, the
imputed date will be updated to the good date. */
do i=1 to 9;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) then do; startwk=stopwk; end;
  if startwk>stopwk and (flag4=1 or flag5=1 or flag6=1) then do; stopwk=startwk; end;
end;

/* Check that previously reported jobs do not predate the date of last interview */
do over UID;
  if UID<=9900 and startwk<r2int and (flag1=1 or flag2=1 or flag3=1)  and startwk ne . and r2int>0 then do;
     check=1; end;
  if UID<=9900 and startwk<r1int and (flag1=1 or flag2=1 or flag3=1)  and startwk ne . and r2int=-5 then do;
     check=1; end;
end;

/* Check for imputed start/stop date in incorrect chronological order */
do i=1 to 9;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) and (flag4=1 or flag5=1 or flag6=1) then do; checky=1;
     end;
end;

/* Check for back-reporters, i.e. people that report a new job (UID>199900) that begins before the last interview
date (back1 or back2) */
do over UID;
  if startwk<r2int and startwk>0 and r2int>0 then do; back2=1; end;
  if startwk<r1int and startwk>0 and r2int=-5 then do; back1=1; end;
end;

/* Check for imputed months that predate the respondent's 14th birthday */
do i=1 to 9; if (flag2=1 or flag3=1) and startwk<age14wk and startwk ne . then do; yowzer=1; end; end;

/* Check for non-imputed values where start week is greater than stop date */
do i=1 to 9;
  if startwk>stopwk and startwk>0 and stopwk>0 and (flag1=0 or flag2=0 or flag3=0) and (flag4=0 or flag5=0 or
     flag6=0) then do; yippe=1; end;
end;
```

```
/* Check for start dates later than stop dates, imputed dates included. */
do over startwk; if startwk>0 and stopwk>0 and startwk>stopwk then do; oops=1; end; end;

/* Check for valid start date and missing stop date and vice versa. */
do i=1 to 9; if (startwk>-6 and stopwk=.) or (startwk=. and stopwk>-6) then do; egads=1; end; end;

/* Check to make sure each non-missing start/stop date has a week number */
do i=1 to 9; if ostartd>0 and ostartm>0 and ostarty>0 and startwk=. then do; golly=1; end; end;

/* Check for the year 1900 in any of the start/stop years */
do i=1 to 9; if ostarty=1900 or ostopy=1900 then woohoo=1; end;

/* Check that non-interview cases are updating correctly */
do i=1 to 9;
  if r2int=-5 and intwk>0 and age14wk>r1int then do; if startwk<r1int and startwk>0 then eeek=1; end;
end;

/* Check that UID's are different at each loop, when greater than -4. */
do i=2 to 9; if UID1=UID and UID1>-4 then wham=1; end;
do i=3 to 9; if UID2=UID and UID2>-4 then wham=1; end;
do i=4 to 9; if UID3=UID and UID3>-4 then wham=1; end;
do i=5 to 9; if UID4=UID and UID4>-4 then wham=1; end;
do i=6 to 9; if UID5=UID and UID5>-4 then wham=1; end;
do i=7 to 9; if UID6=UID and UID6>-4 then wham=1; end;
do i=8 to 9; if UID7=UID and UID7>-4 then wham=1; end;
if UID8=UID9 and UID8>-4 then wham=1;

/* Check that start/stop dates have positive UID values. */
do i=1 to 9; if startwk>0 and UID=-4 then umiss=1; if uid>0 and ostarty=-4 then smiss=1; end;
```

/************ Section 4: Convert dates of within-job gaps into NLSY97 week numbers. *************/**

```
/**  JOB 1 GAPS  **/
/*  These variables are read as follows:
          BGDY1_1 = Begin day of within-job gap 1 on job 1
          EGMO1_5 = End month of within-job gap 5 on job 1
          BGAP1_3 = Begin week of within-job gap 3 on job 1    [CREATED]     */

array bgdy (i) BGDY1_1-BGDY1_5;          array egdy (i) EGDY1_1-EGDY1_5;
array bgmo (i) BGMO1_1-BGMO1_5;          array egmo (i) EGMO1_1-EGMO1_5;
array bgyr (i) BGYR1_1-BGYR1_5;          array egyr (i) EGYR1_1-EGYR1_5;

array bdays (i) bday1_1-bday1_5;      /* begin day of job1_gap#  (internal calculation)*/
array edays (i) eday1_1-eday1_5;      /* end day of job1_gap#    (internal calculation)*/
array bweek (i) bgap1_1-bgap1_5;      /* begin week of job1_gap# (created) */
array eweek (i) egap1_1-egap1_5;      /* end week of job1_gap#   (created) */
array bflag (i) bflg1_1-bflg1_5;
array eflag (i) eflg1_1-eflg1_5;
array bgfl (i) bgfl1_1-bgfl1_5;
array egfl (i) egfl1_1-egfl1_5;

/* ONLY IMPUTE START/STOP DATES IF DAY IS MISSING */
/* Fill-in start day for those missing; impute all missing gap days (start and stop) to 1. */
do over bgyr;
  if bgyr>0 then do;
```

```
    if bgmo>0 and bgdy<=0 then do; bgdy=1; bflag=1; end;
     /* This accounts for beginning gap dates before job start date */
    if bgyr=staryr1 and bgmo=starmo1 and bgdy<stardy1 and bflag=1 then do; bgdy=stardy1; end;
  end;
end;

/* Fill-in stop day for those missing */
do over egyr;
  if egyr>0 then do;
    if egmo>0 and egdy<=0 then do; egdy=1; eflag=1; end;
     /* This accounts for end gap dates after job start date */
    if egyr=stopyr1 and egmo=stopmo1 and egdy>stopdy1 and eflag=1 then do; egdy=stopdy1; end;
  end;
end;

/*Set flag for gap exists but invalid data*/
do over bgyr;
  bgfl=-4;
  if (-4 < bgmo < 0)  or (-4 < bgyr <0) then do; bgfl=1; end;
end;
do over egyr;
  egfl=-4;
  if (-4 < egyr < 0) or (-4 < egmo < 0) then do; egfl=1; end;
end;

***Identify within-job gaps on JOB 1 ***;
***Convert gap dates to week numbers ***;

  ***Convert START month and day to total days (BDAYS);
  do over bgmo;
    if bgmo>0 and bgdy>0 then do;
      if bgmo=1 then bdays=bgdy;                  if bgmo=2 then bdays=bgdy+31;
      if bgmo=3 then bdays=bgdy+59;               if bgmo=4 then bdays=bgdy+90;
      if bgmo=5 then bdays=bgdy+120;              if bgmo=6 then bdays=bgdy+151;
      if bgmo=7 then bdays=bgdy+181;              if bgmo=8 then bdays=bgdy+212;
      if bgmo=9 then bdays=bgdy+243;              if bgmo=10 then bdays=bgdy+273;
      if bgmo=11 then bdays=bgdy+304;             if bgmo=12 then bdays=bgdy+334;
    end;
  end;
  ***Account for leap years;
  do over bgyr;
    if bgyr=1980 or bgyr=1984 or bgyr=1988 or bgyr=1992 or bgyr=1996 or bgyr=2000 then do;
      if bgmo>0 and bgdy>0 then do;
        if bgmo=1 then bdays=bgdy;                          if bgmo=2 then bdays=bgdy+31;
        if bgmo=3 then bdays=bgdy+60;             if bgmo=4 then bdays=bgdy+91;
        if bgmo=5 then bdays=bgdy+121;            if bgmo=6 then bdays=bgdy+152;
        if bgmo=7 then bdays=bgdy+182;            if bgmo=8 then bdays=bgdy+213;
        if bgmo=9 then bdays=bgdy+244;            if bgmo=10 then bdays=bgdy+274;
        if bgmo=11 then bdays=bgdy+305;           if bgmo=12 then bdays=bgdy+335;
      end;
    end;
  end;

  ***Convert STOP month and day to total days (EDAYS);
  do over egmo;
    if egmo>0 and egdy>0 then do;
```

```
      if egmo=1 then edays=egdy;                 if egmo=2 then edays=egdy+31;
      if egmo=3 then edays=egdy+59;              if egmo=4 then edays=egdy+90;
      if egmo=5 then edays=egdy+120;             if egmo=6 then edays=egdy+151;
      if egmo=7 then edays=egdy+181;             if egmo=8 then edays=egdy+212;
      if egmo=9 then edays=egdy+243;             if egmo=10 then edays=egdy+273;
      if egmo=11 then edays=egdy+304;            if egmo=12 then edays=egdy+334;
    end;
  end;
  ***Account for leap years;
  do over egyr;
    if egyr=1980 or egyr=1984 or egyr=1988 or egyr=1992 or egyr=1996 or egyr=2000 then do;
      if egmo>0 and egdy>0 then do;
        if egmo=1 then edays=egdy;               if egmo=2 then edays=egdy+31;
        if egmo=3 then edays=egdy+60;            if egmo=4 then edays=egdy+91;
        if egmo=5 then edays=egdy+121;           if egmo=6 then edays=egdy+152;
        if egmo=7 then edays=egdy+182;           if egmo=8 then edays=egdy+213;
        if egmo=9 then edays=egdy+244;           if egmo=10 then edays=egdy+274;
        if egmo=11 then edays=egdy+305;          if egmo=12 then edays=egdy+335;
      end;
    end;
  end;

  ***Convert days into week numbers.  Note that this program takes the week following the actual start of the gap
as the measure of when the non-working period begins;

  do over bgyr;
    if bgyr>0 and bdays>0 then do;
      if bgyr=1980 then bweek=ceil((bdays+2)/7);          if bgyr=1981 then bweek=52+ceil((bdays+4)/7);
      if bgyr=1982 then bweek=104+ceil((bdays+5)/7);      if bgyr=1983 then bweek=156+ceil((bdays+6)/7);
      if bgyr=1984 then bweek=209+ceil((bdays)/7);        if bgyr=1985 then bweek=261+ceil((bdays+2)/7);
      if bgyr=1986 then bweek=313+ceil((bdays+3)/7);      if bgyr=1987 then bweek=365+ceil((bdays+4)/7);
      if bgyr=1988 then bweek=417+ceil((bdays+5)/7);      if bgyr=1989 then bweek=470+ceil((bdays)/7);
      if bgyr=1990 then bweek=522+ceil((bdays+1)/7);      if bgyr=1991 then bweek=574+ceil((bdays+2)/7);
      if bgyr=1992 then bweek=626+ceil((bdays+3)/7);      if bgyr=1993 then bweek=678+ceil((bdays+5)/7);
      if bgyr=1994 then bweek=730+ceil((bdays+6)/7);      if bgyr=1995 then bweek=783+ceil((bdays)/7);
      if bgyr=1996 then bweek=835+ceil((bdays+1)/7);      if bgyr=1997 then bweek=887+ceil((bdays+3)/7);
      if bgyr=1998 then bweek=939+ceil((bdays+4)/7);      if bgyr=1999 then bweek=991+ceil((bdays+5)/7);
      if bgyr=2000 then bweek=1043+ceil((bdays+6)/7);     if bweek>0 then bweek=bweek+1;
    end;
  end;

  do over egyr;
    if egyr>0 and edays>0 then do;
      if egyr=1980 then eweek=ceil((edays+2)/7);          if egyr=1981 then eweek=52+ceil((edays+4)/7);
      if egyr=1982 then eweek=104+ceil((edays+5)/7);      if egyr=1983 then eweek=156+ceil((edays+6)/7);
      if egyr=1984 then eweek=209+ceil((edays)/7);        if egyr=1985 then eweek=261+ceil((edays+2)/7);
      if egyr=1986 then eweek=313+ceil((edays+3)/7);      if egyr=1987 then eweek=365+ceil((edays+4)/7);
      if egyr=1988 then eweek=417+ceil((edays+5)/7);      if egyr=1989 then eweek=470+ceil((edays)/7);
      if egyr=1990 then eweek=522+ceil((edays+1)/7);      if egyr=1991 then eweek=574+ceil((edays+2)/7);
      if egyr=1992 then eweek=626+ceil((edays+3)/7);      if egyr=1993 then eweek=678+ceil((edays+5)/7);
      if egyr=1994 then eweek=730+ceil((edays+6)/7);      if egyr=1995 then eweek=783+ceil((edays)/7);
      if egyr=1996 then eweek=835+ceil((edays+1)/7);      if egyr=1997 then eweek=887+ceil((edays+3)/7);
      if egyr=1998 then eweek=939+ceil((edays+4)/7);      if egyr=1999 then eweek=991+ceil((edays+5)/7);
      if egyr=2000 then eweek=1043+ceil((edays+6)/7);
if eweek>0 then eweek=eweek-1;
    end;
  end;

array update (i) update1-update5;

/* The following lines omit gap start and stop dates for gaps less than one work week (5 days) */
do over bdays; if edays-bdays<5 and bweek>eweek and bdays ne . and edays ne . then do; bweek=.; eweek=.; end;
end;

/* The following lines are for end gap dates that exceed the Round3 int. date due to rounding methods. */
do over eweek; if 0<eweek<1400 and eweek>intwk then do; eweek=intwk; end; end;

/* The following omits cases where bweek>eweek, which are caused when missing values are substituted in. This
situation will be fixed by making eweek and bweek the same. This is repeated for each job. */
do over bweek; if 0<eweek<1200 then do; if bweek>eweek then bweek=eweek; update=1; end; end;

/* If a repsondent reports a gap on a Round 1 or Round 2 job that occured before the Round 2 interview date, then
we will not use this information.  However, a newly reported job (i.e., 1999 UID) containing within job gaps that
occured before the Round2 interview date will be counted. This is repeated for each job.  Note that we have
respondents who were not interviewed in Round2. The same process is done for these people, except we use the
Round1 int date. */

/* If a gap begins before the Round2 interview date and ends after the Round 3 interview date, the beginning gap
date is updated to the Round 2 interview date. This is repeated at each job. */
do over bweek;
  if 9700<=UID1<9900 and r2int>0 then do;
    /* Entire gap is before Round 2 interview date */
    if r2int>bweek and r2int>eweek then do; bweek=.; eweek=.; end;
    /* B.gap is before R2 interview date, e.gap is after R2 int. date. */
    if r2int<eweek and r2int>bweek then do; bweek=r2int; end;
  end;
end;
do over bweek;
  if 9700<=UID1<9900 and r1int=-5 then do;
    /* Entire gap is before Round 1 interview date */
    if r1int>bweek and r1int>eweek then do; bweek=.; eweek=.; end;
    /* B.gap is before R1 interview date, e.gap is after R2 int. date. */
    if r1int<eweek and r1int>bweek then do; bweek=r1int; end;
  end;
end;

/*The following erases gaps that occur after the job has ended, only in cases where there are no imputed job start or
stop months or years (imputed job start/stop days acceptable) */
do over bweek;
  if bweek>0 and eweek>0 and starw1>0 and stopw1>0 and smofl1=0 and emofl1=0 then do;
    if bweek>stopw1 and eweek>stopw1 then do; bweek=.; eweek=.; end;
  end;
end;

/* To correct for bad gap information.*/
do over bdays;
  if eweek>stopw1 and eweek ne . and stopw1>0 then eweek=stopw1;
  if bweek<starw1 and bweek ne . and bweek>0 then bweek=starw1;
  if bweek>stopw1 and bweek ne . and stopw1>0 then huh=1;
  if eweek<starw1 and eweek ne . and starw1>0 then huh=1;
end;
do over bweek;
```

```
   if 0<bweek<1200 and 0<eweek<1200 then do;
      if bweek>eweek then do; gapmiss=1; eweek=bweek; end;
   end;
end;

/** At this point the program repeats the same gaps code for jobs 2-5.  This code is not reproduced here due to
space constraints.  Researchers needing more information should contact NLS User Services.  Only hand edits in
the jobs 2-5 loops are shown below. **/

/* Job 2 hand edit:  To correct for bad gap information in the raw data. The end gap date is -2/-2/-2, so set the
beginning gap date to -2/-2/-2. */
if pubid=7227 then do; bgdy2_1=-2; bgmo2_1=-2; bgyr2_1=-2; bgap2_1=.; bgfl2_1=1; end;

/* merge information about jobs 1-9 from all of the above code.*/
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

do over starw;
   if starw>0 and stopw>0 then do;
      if r2int>0 then do; if starw<r2int then error1=1; end;
      if r2int=-5 then do; if starw<r1int then error2=1; end;
   end;
end;

/*********************** Section 5:  Weeks worked by respondent on each job
*************************/
/* This part counts the weeks worked by the respondent and removes the within job gaps. It will place a "1" into
weeks where the respondent was employed. Weeks range from the first week of 1980 to the last week of 2000, for a
total of 1096 weeks. */

array job1wks (i) wk1_1-wk1_1096;              array job2wks (i) wk2_1-wk2_1096;
array job3wks (i) wk3_1-wk3_1096;              array job4wks (i) wk4_1-wk4_1096;
array job5wks (i) wk5_1-wk5_1096;              array job6wks (i) wk6_1-wk6_1096;
array job7wks (i) wk7_1-wk7_1096;              array job8wks (i) wk8_1-wk8_1096;
array job9wks (i) wk9_1-wk9_1096;
array starw (i) starw1-starw9;            array stopw (i) stopw1-stopw9;

* Default Settings;
do i=1 to 1096;
   job1wks=0;      job2wks=0;      job3wks=0;      job4wks=0;      job5wks=0;
   job6wks=0;      job7wks=0;      job8wks=0;      job9wks=0;
end;

/* Define a dummy that indicates some type of data error (backreporting, missing stop date, etc.). */
if error1=1 or error2=1 or miss=1 then ignore=1;

/* Define rd3wk as the maximum of dliwk and age14wk. This is used for bad start/stop weeks. */
/* For the re-interview cases, rd3wk will be set to the maximum of the Round 1 interview date and the respondent's
14th birthday. */
if r2int>0 then do;
   if r2int>age14wk then do; rd3wk=r2int; end;
   if age14wk=>r2int then do; rd3wk=age14wk; end;
end;
if r2int=-5 then do;
   if r1int>age14wk then do; rd3wk=r1int; end;
   if age14wk=>r1int then do; rd3wk=age14wk; end;
```

```
end;

/*** TOTAL WEEKS WORKED ON JOB 1 ***/
starfl_1=0;          stopfl_1=0;

if starw1=-3 and uid1 ne -5 then do; starw1=rd3wk; starfl_1=1; end;
if stopw1=-3 and uid1 ne -5 then do; stopw1=intwk; stopfl_1=1; end;
if smofl1=1 then do; starfl_1=1; end;
if emofl1=1 then do; stopfl_1=1; end;

/* [1] */ if starw1>0 and stopw1>0 then do;
  do i=(starw1) to (stopw1); job1wks=1; end;

  *** Remove gap 1 on job 1;
  if bgap1_1>0 & egap1_1>0 then do; do i=(bgap1_1) to (egap1_1); job1wks=0; end; end;
  *** Remove gap 2 on job 1;
  if bgap1_2>0 & egap1_2>0 then do; do i=(bgap1_2) to (egap1_2); job1wks=0; end; end;
  *** Remove gap 3 on job 1;
  if bgap1_3>0 & egap1_3>0 then do; do i=(bgap1_3) to (egap1_3); job1wks=0; end; end;
  *** Remove gap 4 on job 1;
  if bgap1_4>0 & egap1_4>0 then do; do i=(bgap1_4) to (egap1_4); job1wks=0; end; end;
  *** Remove gap 5 on job 1;
  if bgap1_5>0 & egap1_5>0 then do; do i=(bgap1_5) to (egap1_5); job1wks=0; end;
  end;

/* include the end gap dates in the min operators and the beginning gap dates in the max operators. */
  *** Remove gap 1 on job 1 - beginning gap date bad;
  if bgfl1_1=1 & egap1_1>0 then do; do i=(starw1) to (egap1_1); job1wks=-3; gpfl1_1=1; end; end;
  *** Remove gap 1 on job 1 - end gap date bad;
  if bgap1_1>0 & egfl1_1=1 then do;
    do i=(bgap1_1) to min(stopw1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_2+1, egap1_3+1,
         egap1_4+1, egap1_5+1); job1wks=-3; gpfl1_1=1; end;
  end;
  *** Remove gap 1 on job 1 - both gap dates bad;
  if bgfl1_1=1 & egfl1_1=1 then do;
    do i=(starw1) to min(stopw1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_2+1, egap1_3+1,
         egap1_4+1, egap1_5+1); job1wks=-3; gpfl1_1=1; end;
  end;

  *** Remove gap 2 on job 1 - beginning gap date bad;
  if bgfl1_2=1 & egap1_2>0 then do;
    do i=max(starw1, bgap1_1-1, egap1_1+1) to (egap1_2); job1wks=-3; gpfl1_2=1; end;
  end;
  *** Remove gap 2 on job 1 - end gap date bad;
  if bgap1_2>0 & egfl1_2=1 then do;
    do i=(bgap1_2) to min(stopw1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_3+1, egap1_4+1, egap1_5+1);
         job1wks=-3; gpfl1_2=1; end;
  end;
  *** Remove gap 2 on job 1 - both gap dates bad;
  if bgfl1_2=1 & egfl1_2=1 then do;
    do i=max(starw1, bgap1_1-1, egap1_1+1) to min(stopw1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_3+1,
         egap1_4+1, egap1_5+1); job1wks=-3; gpfl1_2=1; end;
  end;

  *** Remove gap 3 on job 1 - beginning gap date bad;
  if bgfl1_3=1 & egap1_3>0 then do;
```

```
    do i=max(starw1,bgap1_1-1,bgap1_2-1,egap1_1+1,egap1_2+1) to (egap1_3); job1wks=-3; gpfl1_3=1; end;
  end;
  *** Remove gap 3 on job 1 - end gap date bad;
  if bgap1_3>0 & egfl1_3=1 then do;
    do i=(bgap1_3) to min(stopw1,bgap1_4-1,bgap1_5-1,egap1_4+1,egap1_5+1); job1wks=-3; gpfl1_3=1; end;
  end;
  *** Remove gap 3 on job 1 - both gap dates bad;
  if bgfl1_3=1 & egfl1_3=1 then do;
    do i=max(starw1,bgap1_1-1,bgap1_2-1,egap1_1+1,egap1_2+1) to min(stopw1,bgap1_4-1,bgap1_5-
          1,egap1_4+1,egap1_5+1); job1wks=-3; gpfl1_3=1; end;
  end;

  *** Remove gap 4 on job 1 - beginning gap date bad;
  if bgfl1_4=1 & egap1_4>0 then do;
    do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,egap1_1+1,egap1_2+1,egap1_3+1) to (egap1_4);
          job1wks=-3; gpfl1_4=1; end;
  end;
  *** Remove gap 4 on job 1 - end gap date bad;
  if bgap1_4>0 & egfl1_4=1 then do;
    do i=(bgap1_4) to min(stopw1,bgap1_5-1,egap1_5+1); job1wks=-3; gpfl1_4=1; end;
  end;
  *** Remove gap 4 on job 1 - both gap dates bad;
  if bgfl1_4=1 & egfl1_4=1 then do;
    do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,egap1_1+1,egap1_2+1,egap1_3+1) to min(stopw1,bgap1_5-
          1,egap1_5-1); job1wks=-3; gpfl1_4=1; end;
  end;

  *** Remove gap 5 on job 1 - beginning gap date bad;
  if bgfl1_5=1 & egap1_5>0 then do;
    do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, egap1_1+1, egap1_2+1, egap1_3+1,
          egap1_4+1) to (egap1_5); job1wks=-3; gpfl1_5=1; end;
  end;
  *** Remove gap 5 on job 1 - end gap date bad;
  if bgap1_5>0 & egfl1_5=1 then do;
    do i=(bgap1_5) to (stopw1); job1wks=-3; gpfl1_5=1; end;
  end;
  *** Remove gap 5 on job 1 - both gap dates bad;
  if bgfl1_5=1 & egfl1_5=1 then do;
    do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, egap1_1+1, egap1_2+1, egap1_3+1,
          egap1_4+1) to (stopw1); job1wks=-3; gpfl1_5=1; end;
  end;
end; /* [1] */

if (starfl_1=1 and ignore ne 1) then do;
  do i=(starw1) to min(stopw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_1+1, egap1_2+1,
          egap1_3+1, egap1_4+1, egap1_5+1); job1wks=-3; end;
end;

if (stopfl_1=1 and ignore ne 1) then do;
  do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, egap1_1+1, egap1_2+1,
          egap1_3+1, egap1_4+1, egap1_5+1) to (stopw1); job1wks=-3; end;
end;

/** At this point the program loops through the same code as above for jobs 2-9.  This code is not shown here due
to space considerations.  Researchers should contact NLS User Services for details if necessary. **/
```

```
endsas;
```

NUMBER OF WEEKS WORKED DURING 19XX

Variables Created: CV_WKSWK_YR.80 – CV_WKSWK_YR.99

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program counts the number of weeks each respondent worked at any employee-type job for each year of potential work activity (1980-99). Respondents not working in a given year are given a default value of zero (0) weeks worked. Otherwise, the variable indicates the actual cumulative weeks worked on all jobs in that year.

```
/* Section 1: Create variables for # of weeks worked at
any job during a given year based on round 3 data. */

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
array job3wks (i) wk3_1-wk3_1096;
array job4wks (i) wk4_1-wk4_1096;
array job5wks (i) wk5_1-wk5_1096;
array job6wks (i) wk6_1-wk6_1096;
array job7wks (i) wk7_1-wk7_1096;
array job8wks (i) wk8_1-wk8_1096;
array job9wks (i) wk9_1-wk9_1096;
array alljobs (i) wks1-wks1096;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array uid (i) uid1-uid9;

do over alljobs; alljobs=0; end;

/** Overlay multiple jobs over JOB 1 work weeks **/
do over job1wks; alljobs=job1wks; end;

do over alljobs;
  if job2wks=1 then do; alljobs=job2wks; end;
  if job2wks=-3 and alljobs=0 then do;
alljobs=job2wks;
  end;
end;

/* The program repeats the “do over all jobs” command
for jobs 3–9.  The loops are not shown here; contact
NLS User Services for more information. */

/* Calculate cumulative weeks on all jobs for each year
*/
/* Need separate do loops for valid data ("1s") and for
invalid data ("-3s") or it will just add them up */

/* 1980 */
wksw80=0;
do i=1 to 52;
  if alljobs=1 then do; wksw80=wksw80+1; end;
end;
do i=1 to 52;
  if alljobs=-3 then do; wksw80=-3; end;
end;

/*********At this point the program loops through the
same code used above for 1980 for each year 1981-
2000, creating the variables wks81, wks82, wks83, and
so on through wks99.  These loops are deleted due to
space considerations; users who need to see the entire
code should contact NLS User Services.  The week
numbers for the “do i” statement for each year are as
follows:
   1981   53-104           1991   575-626
   1982   105-156          1992   627-678
   1983   157-209          1993   679-730
   1984   210-261          1994   731-783
   1985   262-313          1995   784-835
   1986   314-365          1996   836-887
   1987   366-417          1997   888-939
   1988   418-470          1998   940-991
   1989   471-522          1999   992-1044
   1990   523-574          2000   1045-1096 ***/

do i=1 to 9;
/* start date invalid */
if starw<0 and starw>-4 then do;
  if stopw>1 then do; wksw80=-3; end;
  if stopw>52 then do; wksw81=-3; end;
  if stopw>104 then do; wksw82=-3; end;
  if stopw>156 then do; wksw83=-3; end;
  if stopw>209 then do; wksw84=-3; end;
  if stopw>261 then do; wksw85=-3; end;
  if stopw>313 then do; wksw86=-3; end;
  if stopw>365 then do; wksw87=-3; end;
  if stopw>417 then do; wksw88=-3; end;
  if stopw>470 then do; wksw89=-3; end;
  if stopw>522 then do; wksw90=-3; end;
  if stopw>574 then do; wksw91=-3; end;
  if stopw>626 then do; wksw92=-3; end;
  if stopw>678 then do; wksw93=-3; end;
  if stopw>730 then do; wksw94=-3; end;
  if stopw>783 then do; wksw95=-3; end;
  if stopw>834 then do; wksw96=-3; end;
  if stopw>887 then do; wksw97=-3; end;
  if stopw>939 then do; wksw98=-3; end;
  if stopw>991 then do; wksw99=-3; end;
  if stopw>1044 then do; wksw00=-3; end;

/* stop date invalid */
if stopw<0 and stopw>-4 then do;
  if starw<53 then do; wksw80=-3; end;
  if starw<105 then do; wksw81=-3; end;
  if starw<157 then do; wksw82=-3; end;
  if starw<210 then do; wksw83=-3; end;
  if starw<262 then do; wksw84=-3; end;
  if starw<314 then do; wksw85=-3; end;
  if starw<366 then do; wksw86=-3; end;
  if starw<418 then do; wksw87=-3; end;
  if starw<471 then do; wksw88=-3; end;
  if starw<523 then do; wksw89=-3; end;
  if starw<575 then do; wksw90=-3; end;
  if starw<627 then do; wksw91=-3; end;
  if starw<679 then do; wksw92=-3; end;
  if starw<731 then do; wksw93=-3; end;
  if starw<784 then do; wksw94=-3; end;
  if starw<836 then do; wksw95=-3; end;
  if starw<888 then do; wksw96=-3; end;
  if starw<940 then do; wksw97=-3; end;
  if starw<992 then do; wksw98=-3; end;
  if starw<1044 then do; wksw99=-3; end;
  if starw<1096 then do; wksw00=-3; end;
end;

*** Include valid skips;
if age14wk>intwk then do;
  wksw80=-4; wksw81=-4; wksw82=-4; wksw83=-4;
    wksw84=-4; wksw85=-4; wksw86=-4; wksw87=-4;
    wksw88=-4; wksw89=-4; wksw90=-4; wksw91=-4;
    wksw92=-4; wksw93=-4; wksw94=-4; wksw95=-4;
    wksw96=-4; wksw97=-4; wksw98=-4; wksw99=-4;
    wksw00=-4;
end;

if e200=-5 then do;
  wksw80=-5; wksw81=-5; wksw82=-5; wksw83=-5;
    wksw84=-5; wksw85=-5; wksw86=-5; wksw87=-5;
    wksw88=-5; wksw89=-5; wksw90=-5; wksw91=-5;
    wksw92=-5; wksw93=-5; wksw94=-5; wksw95=-5;
    wksw96=-5; wksw97=-5; wksw98=-5; wksw99=-5;
    wksw00=-5;
end;

endsas;
```

NUMBER OF WEEKS WORKED SINCE LAST INTERVIEW

Variables Created: CV_WKSWK_DLI

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

Created for each individual, this program counts the number of weeks the respondent worked at any employee-type job since the last interview. Respondents not working in a given year are given a default value of zero.

```
do over alljobs; alljobs=0; end;

/* Jobs that begin and end before the last interview date
will not be counted in this program. */
array switch (i) switch1-switch9;
do over switch; switch=0; end;
whoa=0;
do over starw;
  if starw<r2int and stopw<r2int and starw>0 and
   stopw>0 and r2int>0 then do; switch=1; whoa=1;
   end;
  if starw<r1int and stopw<r1int and starw>0 and
   stopw>0 and r2int=-5 then do; switch=1; whoa=1;
   end;
end;

do i=1 to 1096;
  if switch1=1 then do; job1wks=0; end;
  if switch2=1 then do; job2wks=0; end;
  /*and so on through*/
  if switch9=1 then do; job9wks=0; end;
end;

/** Overlay multiple jobs over JOB 1 work weeks **/
do over job1wks; alljobs=job1wks; end;
do over alljobs;
  if job2wks=1 then alljobs=job2wks;
  if job2wks=-3 and alljobs=0 then alljobs=job2wks;
end;
/* The program repeats the “do over all jobs” command
for jobs 3–9.  The loops are not shown here; contact
NLS User Services for more information. */

/** Calculate cumulative weeks on all jobs since dli **/
allwks=0;
if r2int>0 then do;
  if r2int=>age14wk then do;
    do i=r2int to 1096; if alljobs=1 then do;
          allwks=allwks+1; end; end;
    do i=r2int to 1096; if alljobs=-3 then do; allwks=-3;
          end; end;
  end;
  if age14wk>r2int then do;
    do i=age14wk to 1096; if alljobs=1 then do;
          allwks=allwks+1; end; end;
    do i=age14wk to 1096; if alljobs=-3 then do;
          allwks=-3; end; end;
  end;
end;
if r2int=-5 then do; /* Round 2 non-interviews */
  if r1int=>age14wk then do;
    do i=r1int to 1096; if alljobs=1 then do;
          allwks=allwks+1; end; end;
    do i=r1int to 1096; if alljobs=-3 then do; allwks=-3;
          end; end;
  end;
  if age14wk>r1int then do;
    do i=age14wk to 1096; if alljobs=1 then do;
          allwks=allwks+1; end; end;
    do i=age14wk to 1096; if alljobs=-3 then do;
          allwks=-3; end; end;
  end;
end;

if starw1<0 and starw1>-4 then do; allwks=-3; end;
if starw2<0 and starw2>-4 then do; allwks=-3; end;
if starw3<0 and starw3>-4 then do; allwks=-3; end;
if starw4<0 and starw4>-4 then do; allwks=-3; end;
if starw5<0 and starw5>-4 then do; allwks=-3; end;
if starw6<0 and starw6>-4 then do; allwks=-3; end;
if starw7<0 and starw7>-4 then do; allwks=-3; end;
if starw8<0 and starw8>-4 then do; allwks=-3; end;
if starw9<0 and starw9>-4 then do; allwks=-3; end;
if stopw1<0 and stopw1>-4 then do; allwks=-3; end;
if stopw2<0 and stopw2>-4 then do; allwks=-3; end;
if stopw3<0 and stopw3>-4 then do; allwks=-3; end;
if stopw4<0 and stopw4>-4 then do; allwks=-3; end;
if stopw5<0 and stopw5>-4 then do; allwks=-3; end;
if stopw6<0 and stopw6>-4 then do; allwks=-3; end;
if stopw7<0 and stopw7>-4 then do; allwks=-3; end;
if stopw8<0 and stopw8>-4 then do; allwks=-3; end;
if stopw9<0 and stopw9>-4 then do; allwks=-3; end;

if age14wk>intwk then do; allwks=-4; end;
if e200=-3 then do; allwks=-3; end;
if e200=-5 then do; allwks=-5; end;
if r2int>0 then do; maxcv=intwk-r2int+1; end;
if r2int=-5 then do; maxcv=intwk-r1int+1; end;
endsas;
```

NUMBER OF WEEKS WORKED SINCE AGE 14

Variables Created: CV_WKSWK_EVER

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

For each respondent, this program creates a variable counting the number of weeks worked at any employee-type job since the age of 14. Respondents not working are given a default value of zero.

```
/* Section 1:  Creates a variable for number of weeks
worked at any job since age 14 using R3 data. */

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
  /*and so on through*/
array job9wks (i) wk9_1-wk9_1096;
array alljobs (i) wks1-wks1096;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

/** Overlay multiple jobs over JOB 1 work weeks **/
do over job1wks; alljobs=job1wks; end;

do over alljobs;
  if job2wks=1 then alljobs=job2wks;
  if job2wks=-3 and alljobs=0 then alljobs=job2wks;
end;

/* The program repeats the "do over all jobs" command
for jobs 3–9.  The loops are not shown here; contact
NLS User Services for more information. */

/* Calculate cumulative weeks on all jobs since age 14
*/
/* wks14w is the variable that counts the number of
weeks worked since respondent's 14th birthday. */

wks14w=0;
if age14wk>0 then do;
  do i=age14wk to 1096;
    if alljobs=1 then do; wks14w=wks14w+1; end;
  end;
  do i=age14wk to 1096;
    if alljobs=-3 then do; wks14w=-3; end;
  end;
end;

do i=1 to 9;
if starw<0 and starw>-4 then do;
  if age14wk<53 and stopw>1 then wks14w=-3;
  if age14wk<105 and stopw>52 then wks14w=-3;
  if age14wk<157 and stopw>104 then wks14w=-3;
  if age14wk<210 and stopw>156 then wks14w=-3;
  if age14wk<262 and stopw>209 then wks14w=-3;
  if age14wk<314 and stopw>261 then wks14w=-3;
  if age14wk<366 and stopw>313 then wks14w=-3;
  if age14wk<418 and stopw>365 then wks14w=-3;
  if age14wk<471 and stopw>417 then wks14w=-3;
  if age14wk<523 and stopw>470 then wks14w=-3;
  if age14wk<575 and stopw>522 then wks14w=-3;
  if age14wk<627 and stopw>574 then wks14w=-3;
  if age14wk<679 and stopw>626 then wks14w=-3;
  if age14wk<731 and stopw>678 then wks14w=-3;
  if age14wk<784 and stopw>730 then wks14w=-3;
  if age14wk<836 and stopw>783 then wks14w=-3;
  if age14wk<888 and stopw>835 then wks14w=-3;
  if age14wk<940 and stopw>887 then wks14w=-3;
  if age14wk<991 and stopw>939 then wks14w=-3;
  if age14wk<1044 and stopw>991 then wks14w=-3;
  if age14wk<1096 and stopw>1044 then wks14w=-3;
end;

if stopw<0 and stopw>-4 then do;
  if age14wk<53 and starw<53 then wks14w=-3;
  if age14wk<105 and starw<105 then wks14w=-3;
  if age14wk<157 and starw<157 then wks14w=-3;
  if age14wk<210 and starw<210 then wks14w=-3;
  if age14wk<262 and starw<262 then wks14w=-3;
  if age14wk<314 and starw<314 then wks14w=-3;
  if age14wk<366 and starw<366 then wks14w=-3;
  if age14wk<418 and starw<418 then wks14w=-3;
  if age14wk<471 and starw<471 then wks14w=-3;
  if age14wk<523 and starw<523 then wks14w=-3;
  if age14wk<575 and starw<575 then wks14w=-3;
  if age14wk<627 and starw<627 then wks14w=-3;
  if age14wk<679 and starw<679 then wks14w=-3;
  if age14wk<731 and starw<731 then wks14w=-3;
  if age14wk<784 and starw<784 then wks14w=-3;
  if age14wk<836 and starw<836 then wks14w=-3;
  if age14wk<888 and starw<888 then wks14w=-3;
  if age14wk<940 and starw<940 then wks14w=-3;
  if age14wk<991 and starw<991 then wks14w=-3;
  if age14wk<1044 and starw<1044 then wks14w=-3;
  if age14wk<1096 and starw<1096 then wks14w=-3;
end;
end;

*** Include valid skips;
```

```
if e200=-3 then do; wks14w=-3; end;
if e200=-5 then do; wks14w=-5; end;

/* Initialize created variable for each round to zero. */
totwks14=0;

/* For the Round 3 created variable, we can simply add
the two created variables from Round 2 and Round 3 if
they are both postive. If one is positive and one is zero,
then the positive value will be the total created variable
for both rounds. If neither is positive, then the total
created variable for both rounds will be zero. */

if r2wks14>0 then do; totwks14=r2wks14; end;
/* non-interview case */
if r1wks14>0 and r2int=-5 and intwk>0 then do;
  totwks14=r1wks14;
end;

if r3wks14>0 then do; totwks14=r3wks14; end;

/* Both Round 2 and Round 3 positive */
if r2wks14>0 and r3wks14>0 then do;
  totwks14=r2wks14+r3wks14;
end;

/* Both R1 and R3 positive, R2 non-interview */
if r1wks14>0 and r3wks14>0 and r2int=-5 and intwk>0
then do;
  totwks14=r1wks14+r3wks14;
end;

/* Define negative values for the total created variable.
*/
if -4<r2wks14<0 then totwks14=-3;
if -4<r1wks14<0 and r2int=-5 and intwk>0
   then totwks14=-3;
if -4<r3wks14<0 then totwks14=-3;
if r3wks14=-5 then totwks14=-5;

/* Some problems in the raw data were not fixed before
the data release. The dummy "ignore" is used later in
this program to identify these cases for hand-editing. */
if pubid in (175,3196) then ignore=.;
if ignore=1 then totwks14=-3;

endsas;
```

WEEKS WORKED AT EMPLOYEE JOB #X DURING 19XX

Variables Created: CV_WKSWK_JOB_YR.01.80 – CV_WKSWK_JOB_YR.01.00
CV_WKSWK_JOB_YR.02.80 – CV_WKSWK_JOB_YR.02.00 etc. through job #9

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program creates variables for each of the respondent's jobs counting the number of weeks worked in each calendar year. A variable is created for each potential job even if the respondent has worked no jobs in a given year with the default value set to zero (0). The most jobs held by any respondent as of round 3 was nine, so variables are created for nine jobs for each respondent.

```
/* Section 1: Create the Round3 variable for number
of weeks worked at a given job in a given year. */
array starw (i) starw1-starw9;
array job1wks (i) wk1_1-wk1_1098;
array job2wks (i) wk2_1-wk2_1098;
  /*repeat for jobs 3-9*/

/** Calculate cumulative weeks on individual jobs for
each year **/
/* 1980 */
wksw801=0; wksw802=0; wksw803=0; wksw804=0;
wksw805=0; wksw806=0; wksw807=0; wksw808=0;
wksw809=0;
do i=1 to 52;
  if job1wks=1 then do; wksw801=wksw801+1; end;
  if job2wks=1 then do; wksw802=wksw802+1; end;
  /*repeat for jobs 3-9*/
end;
do i=1 to 52;
  if job1wks=-3 then do; wksw801=-3; end;
  if job2wks=-3 then do; wksw802=-3; end;
  /*repeat for jobs 3-9*/
end;

/*** The same three sets of commands applied to the
1980 weeks variables are used for each subsequent
year. However, this code is not shown here due to
space considerations. Researchers needing the
complete code should contact NLS User Services. The
variables and "do i" statements for each year are as
follows:
     1981   wksw811-wksw819   do i=53 to 104
     1982   wksw821-wksw829   do i=105 to 156
     1983   wksw831-wksw839   do i=157 to 209
     1984   wksw841-wksw849   do i=210 to 261
     1985   wksw851-wksw859   do i=262 to 313
     1986   wksw861-wksw869   do i=314 to 365
     1987   wksw871-wksw879   do i=366 to 417
     1988   wksw881-wksw889   do i=418 to 470
     1989   wksw891-wksw899   do i=471 to 522
     1990   wksw901-wksw909   do i=523 to 574
     1991   wksw911-wksw919   do i=575 to 626
     1992   wksw921-wksw929   do i=627 to 678
     1993   wksw931-wksw939   do i=679 to 730
     1994   wksw941-wksw949   do i=731 to 783
     1995   wksw951-wksw959   do i=784 to 835
     1996   wksw961-wksw969   do i=836 to 887
     1997   wksw971-wksw979   do i=888 to 939
     1998   wksw981-wksw989   do i=940 to 991
     1999   wksw991-wksw999   do i=992 to 1044
     2000   wksw001-wksw009   do i=1045 to 1098   **/

*** Insert valid skips;

if age14wk>intwk then do;
  wksw801=-4; /* and so on through wksw809=-4 */
  /*and so on through wksw001=-4 to wksw009=-4 */
end;

if e200=-3 then do;
  wksw801=-3; /* and so on through wksw809=-3 */
  /*and so on through wksw001=-3 to wksw009=-3 */
end;

if e200=-5 then do;
  wksw801=-5; /* and so on through wksw809=-5 */
  /*and so on through wksw001=-5 to wksw009=-5 */
end;

/*start date invalid*/
starw1<0 and starw1>-4
  if stopw1>1 then wksw801=-3;
  if stopw1>52 then wksw811=-3;
  if stopw1>104 then wksw821=-3;
  if stopw1>156 then wksw831=-3;
  if stopw1>209 then wksw841=-3;
  if stopw1>261 then wksw851=-3;
  if stopw1>313 then wksw861=-3;
  if stopw1>365 then wksw871=-3;
  if stopw1>417 then wksw881=-3;
  if stopw1>470 then wksw891=-3;
  if stopw1>522 then wksw901=-3;
  if stopw1>574 then wksw911=-3;
  if stopw1>626 then wksw921=-3;
  if stopw1>678 then wksw931=-3;
  if stopw1>730 then wksw941=-3;
  if stopw1>783 then wksw951=-3;
  if stopw1>835 then wksw961=-3;
  if stopw1>887 then wksw971=-3;
  if stopw1>939 then wksw981=-3;
  if stopw1>991 then wksw991=-3;
  if stopw1>1004 then wksw001=-3;
end;

/*At this point the above start date code repeats for
jobs 2 through 9.  For example, startw2, stopw2, and
wksw802-wksw002 are substituted for job 2.  For more
information please contact NLS User Services.*/

/* stop date invalid */
if stopw1<0 and stopw1>-4 then do;
  if starw1<53 then wksw801=-3;
  if starw1<105 then wksw811=-3;
  if starw1<157 then wksw821=-3;
  if starw1<210 then wksw831=-3;
  if starw1<262 then wksw841=-3;
  if starw1<314 then wksw851=-3;
  if starw1<366 then wksw861=-3;
  if starw1<418 then wksw871=-3;
  if starw1<471 then wksw881=-3;
  if starw1<523 then wksw891=-3;
  if starw1<575 then wksw901=-3;
  if starw1<627 then wksw911=-3;
  if starw1<679 then wksw921=-3;
  if starw1<731 then wksw931=-3;
  if starw1<784 then wksw941=-3;
  if starw1<836 then wksw951=-3;
  if starw1<888 then wksw961=-3;
  if starw1<940 then wksw971=-3;
  if starw1<991 then wksw981=-3;
  if starw1<1044 then wksw991=-3;
  if starw1<1096 then wksw001=-3;
end;

/*At this point the above start date code repeats for
jobs 2 through 9.  For example, stopw2, startw2, and
wksw802-wksw002 are substituted for job 2.  For more
information please contact NLS User Services.*/

/* WKS945 is the Round 2 created variable for 5th job
(i.e. 5th UID code) in 1994. WKSW945 is the Round 3
created varible of the same description. TWKS594 is
the variable that combines these two created variables
(of the same description). Clearly these infile variable
names are confusing, so we rename them to more
familiar "r1wks, r2wks" etc. TWKS remains the total
created variable. */

drop uid1-uid9;
array r1uid (i) r1uid1-r1uid9;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
/* Note there are 7 jobs in Round 1, array matched to
make programming easier */

/* Arrange the jobs by year for R2 (infile var names)*/
array wksj80 (i) wks801-wks809;
array wksj81 (i) wks811-wks819;
  /*and so on through*/
array wksj99 (i) wks991-wks999;

/*Arrange the jobs by year for R3 (original var names)
*/
array wksw80 (i) wksw801-wksw809;
array wksw81 (i) wksw811-wksw819;
  /*and so on through*/
array wksw00 (i) wksw001-wksw009;

/* Arrange the jobs by year for Round 1 */
array r1wks80 (i) r1wks801-r1wks809;
array r1wks81 (i) r1wks811-r1wks819;
  /*and so on through*/
array r1wks00 (i) r1wks001-r1wks009;

/* Arrange the jobs by year for Round 2 */
array r2wks80 (i) r2wks801-r2wks809;
array r2wks81 (i) r2wks811-r2wks819;
  /*and so on through*/
array r2wks00 (i) r2wks001-r2wks009;

/* Arrange the jobs by year for Round 3 */
array r3wks80 (i) r3wks801-r3wks809;
array r3wks81 (i) r3wks811-r3wks819;
  /*and so on through*/
array r3wks00 (i) r3wks001-r3wks009;

/* Arrange the jobs by year for total created value */
array twks80 (i) twks801-twks809;
array twks81 (i) twks811-twks819;
  /*and so on through*/
array twks00 (i) twks001-twks009;

drop wks801-wks809 wks811-wks819 wks821-wks829
wks831-wks839 wks841-wks849 wks851-wks859
wks861-wks869 wks871-wks879 wks881-wks889
wks891-wks899 wks901-wks909 wks911-wks919
wks921-wks929 wks931-wks939 wks941-wks949
wks951-wks959 wks961-wks969 wks971-wks979
wks981-wks989 wks991-wks999;

/* Rename Round 3 c.v. and drop replaced variables  */
do i=1 to 9;
    r3wks80=wksw80; r3wks81=wksw81;
     r3wks82=wksw82; r3wks83=wksw83;
     r3wks84=wksw84; r3wks85=wksw85;
     r3wks86=wksw86; r3wks87=wksw87;
     r3wks88=wksw88; r3wks89=wksw89;
    r3wks90=wksw90; r3wks91=wksw91;
    r3wks92=wksw92; r3wks93=wksw93;
    r3wks94=wksw94; r3wks95=wksw95;
    r3wks96=wksw96; r3wks97=wksw97;
    r3wks98=wksw98; r3wks99=wksw99;
    r3wks00=wksw00;
end;

drop wksw801-wksw809 wksw811-wksw819
wksw821-wksw829 wksw831-wksw839 wksw841-
wksw849 wksw851-wksw859 wksw861-wksw869
wksw871-wksw879 wksw881-wksw889 wksw891-
wksw899 wksw901-wksw909 wksw911-wksw919
wksw921-wksw929 wksw931-wksw939 wksw941-
wksw949 wksw951-wksw959 wksw961-wksw969
wksw971-wksw979 wksw981-wksw989 wksw991-
wksw999 wksw001-wksw009;

/* Initialize total weeks count to zero */
do i=1 to 9;
  twks80=0; twks81=0; twks82=0; twks83=0;
    twks84=0; twks85=0; twks86=0; twks87=0;
    twks88=0; twks89=0; twks90=0; twks91=0;
    twks92=0; twks93=0; twks94=0; twks95=0;
    twks96=0; twks97=0; twks98=0; twks99=0;
    twks00=0;
end;

/* Accounts for Round 3 non-interview cases */
do i=1 to 9;
  if e200=-5 then do;
    twks80=-5; twks81=-5; twks82=-5; twks83=-5;
         twks84=-5; twks85=-5; twks86=-5; twks87=-5;
         twks88=-5; twks89=-5; twks90=-5; twks91=-5;
         twks92=-5; twks93=-5; twks94=-5; twks95=-5;
         twks96=-5; twks97=-5; twks98=-5; twks99=-5;
         twks00=-5;
  end;
end;

/* Begin by updating the total created variable if the
Round3 created variable is positive. For this variable,
we are only worried about Round3 jobs and Round 1 or
2 jobs that were worked in Round3. If a negative value
is given (don't know, refusal, or non-interview) then
total created variable will be that negative value. */

/* 1980 */
do i=1 to 9;
  if r3wks80=>0 then do; twks80=r3wks80; end;
  if -4<r3wks80<0 then do; twks80=-3; end;
end;

/*repeat the above code for 1981-1997*/

/* 1998 */
/* Since 1998 and 1999 are the years in which the
Round 2 interview date occurred, need to account for
both Round 1 and Round 2 created variables. Begin
with negative conditions. */

do i=1 to 9;
  if -4<r3wks98<0 then do; twks98=-3; end;
  if r3wks98=>0 then do; twks98=r3wks98; end;
end;

/* 1999 */
do i=1 to 9;
  if -4<r3wks99<0 then do; twks99=-3; end;
  if r3wks99=>0 then do; twks99=r3wks99; end;
end;

/* When matching jobs from Round 2 to Round 3, we
only need to worry about jobs in Round 3 with a UID
beginning with "98" or "97" since only those jobs that
could have been worked in both rounds. */

array match1 (i) match11-match19;
array match2 (i) match21-match29;
  /*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
        match5=0; match6=0; match7=0; match8=0;
        match9=0;
end;

/* Define match14 as the dummy variable that equals
one when the first job on the Round1 UID roster and the
fourth job on the Round2 roster have the same UID. */

/* If any UID from the first position in dli Round
matches with any UID in Round 3 */
do over r3uid;
  if r2uid1>0 and r3uid>0 and r2int>0 and intwk>0 then
        do; if r2uid1=r3uid then do; match1=1; end;
        end;
  if r1uid1>0 and r3uid>0 and r2int=-5 and intwk>0
        then do; if r1uid1=r3uid then do; match1=1;
        end; end;
end;

/* The above code repeats for R1 jobs 2-7 and R2 jobs
2-9.  For example, r2uid2, r1uid2, and match2 are
substituted for r2uid1, r1uid1, and match 1.  Contact
NLS User Services for more information. */

/* Now reassign total created variable if there is a job
was worked in both rounds. Using the match variable, a
job worked in both rounds will update the total created
```

```
variable. Need to add positivity condition on the weeks
worked values since some of the values equal -3. */

/* For respondents interviewed in Round 2 */

if r2int_y=1998 then do;
  do over match1;
    if match1=1 then do;
      if r2wks801 not in (-5,-4,0,.) then
twks80=r2wks801;
      if r2wks811 not in (-5,-4,0,.) then
twks81=r2wks811;
      /*and so on through*/
      if r2wks971 not in (-5,-4,0,.) then
twks97=r2wks971;
    end;
  end;

  do over match1;
    if match1=1 and r3wks98=>0 and r2wks981=>0
            then do; twks98=r2wks981+r3wks98; end;
      if match1=1 and (-4<r3wks98<0 or -
            4<r2wks981<0) then do; twks98=-3; end;
   end;

/* The above code repeats for match 2-9.  For example,
match2, r2wks802, and r2wks982 are substituted for
match1, r1wks801, and r1wks981.  Contact NLS User
Services if more details are needed. */
end;

   /* 1999 */
   do i=1 to 9;
      if -4<r3wks99<0 then do; twks99=-3; end;
      /* Update if Round 3 created variable is positive */
      if r3wks99>0 then do; twks99=r3wks99; end;
   end;
end;

/* Now reassign total created variable if there is a job
was worked in both rounds. Using the match variable, a
job worked in both rounds will update the total created
variable. Need to add positivity condition on the weeks
worked values since some of the values equal -3. */

if r2int_y=1999 then do;
  do over match1;
    if match1=1 then do;
      if r2wks801 not in (-5,-4,0,.) then
twks80=r2wks801;
      if r2wks811 not in (-5,-4,0,.) then
twks81=r2wks811;
      /*and so on through*/
      if r2wks981 not in (-5,-4,0,.) then
twks98=r2wks981;
    end;
  end;

  do over match1;
    if match1=1 and r3wks99=>0 and r2wks991=>0 then
            do; twks99=r2wks991+r3wks99; end;
    if match1=1 and (-4<r3wks99<0 or -4<r2wks991<0)
            then do; twks99=-3; end;
   end;

/* The above code repeats for match 2-9.  For example,
match2, r2wks802, and r2wks982 are substituted for
match1, r1wks801, and r1wks981.  Contact NLS User
Services if more details are needed. */
end;

/* For respondents interviewed in R1 & R3, but not R2
*/
if r1int_y=1998 and r2int=-5 and intwk>0 then do;
   do over match1;
      if match1=1 then do;
      if r1wks801 not in (-5,-4,0,.) then
twks80=r1wks801;
      if r1wks811 not in (-5,-4,0,.) then
twks81=r1wks811;
      /*and so on through*/
      if r1wks971 not in (-5,-4,0,.) then
twks97=r1wks971;
      end;
   end;

   do over match1;
      if match1=1 and r3wks98=>0 and r1wks981=>0
            then do; twks98=r1wks981+r3wks98; end;
      if match1=1 and (-4<r3wks98<0 or -4<r1wks981<0)
            then do; twks98=-3; end;
   end;

/* The above code repeats for match 2-9.  For example,
match2, r2wks802, and r2wks982 are substituted for
match1, r1wks801, and r1wks981.  Contact NLS User
Services if more details are needed. */
   /* 1999 */
   do i=1 to 9;
      if -4<r3wks99<0 then do; twks99=-3; end;
      /* Update if Round 3 created variable is positive */
      if r3wks99>0 then do; twks99=r3wks99; end;
   end;
end;

/* For respondents interviewed in R1 & R3, but not
R2*/
if r1int_y=1997 and r2int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 then do;
      if r1wks801 not in (-5,-4,0,.) then
twks80=r1wks801;
```

```
     if r1wks811 not in (-5,-4,0,.) then
twks81=r1wks811;
     /*and so on through*/
     if r1wks961 not in (-5,-4,0,.) then
twks96=r1wks961;
    end;
  end;

   do over match1;
     if match1=1 and r3wks97=>0 and r1wks971=>0
           then do; twks97=r1wks971+r3wks97; end;
     if match1=1 and (-4<r3wks97<0 or -
           4<r1wks971<0) then do; twks97=-3; end;

   end;

/* The above code repeats for match 2-7 (because the
maximum number of jobs reported in R1 was 7).  For
example, match2, r2wks802, and r2wks982 are
substituted for match1, r1wks801, and r1wks981.
Contact NLS User Services if more details are needed.
*/

   /* 1998 */
   do i=1 to 9;
     if -4<r3wks98<0 then do; twks98=-3; end;
     /* Update if Round 3 created variable is positive */
     if r3wks98>0 then do; twks98=r3wks98; end;
   end;

   /* 1999 */
   do i=1 to 9;
     if -4<r3wks99<0 then do; twks99=-3; end;
     /* Update if Round 3 created variable is positive */
     if r3wks99>0 then do; twks99=r3wks99; end;
   end;
end;

/* 2000 can only use Round 3 created variables */
do i=1 to 9;
   if -4<r3wks00<0 then do; twks00=-3; end;
   if r3wks00>0 then do; twks00=r3wks00; end;
end;

/* The first case has a missing UID; the second two
cases have two identical UID's in the Round 3 roster.
They are hand-edited to avoid assigning a -3. */
if pubid=157 then do; twks994=5; twks992=2;
           ignore=.; end;
if pubid=175 then do; twks981=5; twks991=3;
           twks001=2; end;
if pubid=3196 then do; twks971=17; twks981=14;
           twks991=16; end;

/* The following respondents are backreporters that
had a job that began and ended before the last
interview date.  These jobs are included in the job-
specific and job count c.v.'s but not the year-specific
c.v.'s (except this one). */
if pubid=202 then do; twks981=27; twks981=2;
           ignore=.; end;
if pubid=1222 then do; twks981=14; ignore=.; end;
if pubid=1419 then do; twks981=11; ignore=.; end;
if pubid=1590 then do; twks972=11; ignore=.; end;
if pubid=2888 then do; twks981=16; ignore=.; end;
if pubid=3468 then do; twks972=24; twks982=33;
           ignore=.; end;
if pubid=3526 then do; twks982=14; ignore=.; end;
if pubid=4658 then do; twks973=26; twks983=23;
           ignore=.; end;
if pubid=5995 then do; twks972=7; ignore=.; end;
if pubid=7465 then do; twks981=9; ignore=.; end;
if pubid=8827 then do; twks981=1; ignore=.; end;

/* flag problem cases where total weeks count exceeds
52 weeks in 1997 or 1998 */
flag97=0;          flag98=0;          flag99=0;
do i=1 to 9;
   if twks97>52 then flag97=1;
   if twks98>52 then flag98=1;
   if twks99>53 then flag99=1;
end;

array starw (i) starw1-starw9;
/* 1997 */
do i=1 to 9;
   if ignore=1 and starw>0 and twks97>0 then do;
     twks97=-3;
   end;
end;

/* 1998 */
do i=1 to 9;
   if ignore=1 and starw>0 and twks98>0 then do;
     twks98=-3;
   end;
end;

/* 1999 */
do i=1 to 9;
   if ignore=1 and starw>0 and twks99>0 then do;
     twks99=-3;
   end;
end;

/* 2000 */
do i=1 to 9;
   if ignore=1 and starw>0 and twks00>0 then do;
     twks00=-3;
   end;
end;
```

```
endsas;
```

TOTAL TENURE AT JOB #X AS OF THE SURVEY DATE

Variables Created: CV_WKSWK_JOB_DLI.01 – CV_WKSWK_JOB_DLI.09

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program creates a variable for each job calculating the total length of job tenure in weeks (excluding within-job gaps) since the respondent's 14th birthday. A variable is created for each potential job even if the respondent has no data for that job, with the default value set to zero (0). The most jobs held by any respondent as of round 2 was nine, so variables are created for nine jobs for each respondent.

```
/* Section 1: Create a variable for each potential job
(9) relating the total length of job tenure in weeks
excluding within-job gaps.*/

array starw (i) starw1-starw9;
array job1wks (i) wk1_1-wk1_1098;
array job2wks (i) wk2_1-wk2_1098;
array job3wks (i) wk3_1-wk3_1098;
array job4wks (i) wk4_1-wk4_1098;
array job5wks (i) wk5_1-wk5_1098;
array job6wks (i) wk6_1-wk6_1098;
array job7wks (i) wk7_1-wk7_1098;
array job8wks (i) wk8_1-wk8_1098;
array job9wks (i) wk9_1-wk9_1098;

/* Calculate cumulative weeks on each job for all years
*/
tenure1=0;      tenure2=0;      tenure3=0;
tenure4=0;      tenure5=0;      tenure6=0;
tenure7=0;      tenure8=0;      tenure9=0;

do i=1 to 1098;
   if job1wks=1 then do; tenure1=tenure1+1; end;
   if job2wks=1 then do; tenure2=tenure2+1; end;
   /*repeat for jobs 3-9*/
end;

flag=0;

do i=1 to 1098;
   if job1wks=-3 then do; tenure1=-3; flag=1; end;
   if job2wks=-3 then do; tenure2=-3; flag=2; end;
   /*repeat for jobs 3-9*/
end;

do i=1 to 1098;
   if starw1<0 and starw1>-4 then do; tenure1=-3; end;
   if starw2<0 and starw2>-4 then do; tenure2=-3; end;
   /*repeat for jobs 3-9*/
   if stopw1<0 and stopw1>-4 then do; tenure1=-3; end;
   if stopw2<0 and stopw2>-4 then do; tenure2=-3; end;
   /*repeat for jobs 3-9*/
end;
```

```
if age14wk>intwk then do;
   tenure1=-4; tenure2=-4; tenure3=-4; tenure4=-4;
        tenure5=-4; tenure6=-4; tenure7=-4; tenure8=-4;
        tenure9=-4;
end;

if e200=-3 then do;
   tenure1=-3; tenure2=-3; tenure3=-3; tenure4=-3;
        tenure5=-3; tenure6=-3; tenure7=-3; tenure8=-3;
        tenure9=-3;
end;

if e200=-5 then do;
   tenure1=-5; tenure2=-5; tenure3=-5; tenure4=-5;
        tenure5=-5; tenure6=-5; tenure7=-5; tenure8=-5;
        tenure9=-5;
end;

/* Section 2: Calculate the total job tenure by matching
up the UID's from Round2 and Round3. */

/* Begin by renaming the infile variables */
array totten (i) totten1-totten9; /* R2 infile c.v. */
array r2ten (i) r2ten1-r2ten9; /* New R2 c.v. name */
array tenure (i) tenure1-tenure9; /* R3 infile c.v. */
array r3ten (i) r3ten1-r3ten9; /* New R3 c.v. name */
do i=1 to 9; r3ten=tenure; totten=0; end;

/* "TOTTEN" will be the new total (R1, R2, & R3)
tenure variable */
drop tenure1-tenure9;

/* When matching jobs from Round 2 to Round 3, we
only need to worry about jobs in Round 3 with a UID
beginning with "97" or "98" since only those jobs that
could have been worked in both rounds. */

array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array match1 (i) match11-match19;
array match2 (i) match21-match29;
array match3 (i) match31-match39;
array match4 (i) match41-match49;
array match5 (i) match51-match59;
array match6 (i) match61-match69;
array match7 (i) match71-match79;
array match8 (i) match81-match89;
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
    match5=0; match6=0; match7=0; match8=0;
    match9=0;
end;

/* Define match14 as the dummy variable that equals
one when the first job on the Round1 UID roster and
the fourth job on the Round2 roster have the same UID.
*/

/* If any UID from the first position in dli Round
matches with any UID in Round 3 */
do over r3uid;
  if r2uid1>0 and r3uid>0 and r2int>0 and intwk>0
           then do; if r2uid1=r3uid then match1=1; end;
  if r1uid1>0 and r3uid>0 and r2int=-5 and intwk>0
           then do; if r1uid1=r3uid then match1=1; end;
end;

/* The above code repeats for R1 jobs 2-7 and R2 jobs
2-9.  For example, r2uid2, r1uid2, and match2 are
substituted for r2uid1, r1uid1, and match 1.  Contact
NLS User Services for more information. */

/* Initialize Round 3 total tenure variable to zero */
do over totten; totten=0; end;

/* There can only be 9 jobs for the respondent, since
we don't count jobs that were only worked in R1 & 2.
Recall that the only Round 2 jobs that we are worried
about are the one that were also worked in Round 3.
This will take care of the negative values under the
Round 3 count. */
do over totten; totten=r3ten; end;

/* Now reassign total tenure if there is a job was
worked in both rounds. Using the match variable, a job
worked in both rounds 2 & 3 will update the total
tenure variable. Need to add positivity condition on the
tenure values since some of the values equal -3. Inside
of each match loop, the first two do loops are for
respondents who were interviewed in Round 2.  The
last two do loops are for respondents who were not
interviewed in Round 2. */

do over match1;
  if match1=1 and r3ten=>0 and r2ten1=>0 and r2int>0
          then do; totten=r2ten1+r3ten; end;
  if match1=1 and -4<r2ten1<0 and r2int>0 then do;
          totten=-3; end;
  if match1=1 and r3ten=>0 and r1ten1=>0 and r2int=-5
          and r1int>0 then do; totten=r1ten1+r3ten; end;
  if match1=1 and -4<r1ten1<0 and r2int=-5 and
          r1int>0 then do; totten=-3; end;
end;

/* The above code repeats for R1 jobs 2-7 and R2 jobs
2-9.  For example, match2, r2ten2, and r1ten2 are
substituted for match1, r2ten1, and r1ten1.  Contact
NLS User Services for more information. */

/* Anyone under 14 at Round 3 int. date has -4's in
employment variables. */
do over totten;
  if age14wk>intwk and intwk>0 then do;
    totten=-4; young=1;
  end;
end;

/* The following two people have identical UID's in the
Round 3 roster. Each respondent worked this identical
job in Round 2. So, the first loop of the c.v. will be a
positive number that adds both the separate Round 2
and Round 3 counts, and the second loop will be -3. */
if pubid=175 then do; totten1=10; totten2=-3; ignore=.;
          end;
if pubid=3196 then do; totten1=47; totten2=-3; ignore=.;
          end;

/* The following respondents are back reporters who
began and ended a job before the last interview date
(Round 2 in all cases below). These jobs are used only
in the job-specific and job count variables, but not in the
year specific c.v.'s (the one exception is the job and year
specific c.v., in which these jobs are included). */
if pubid=202 then do; totten1=29; ignore=.; end;
if pubid=1222 then do; totten1=14; ignore=.; end;
if pubid=1419 then do; totten1=11; ignore=.; end;
if pubid=1590 then do; totten2=11; ignore=.; end;
if pubid=2888 then do; totten1=16; ignore=.; end;
if pubid=3468 then do; totten2=57; ignore=.; end;
if pubid=3526 then do; totten2=14; ignore=.; end;
if pubid=4658 then do; totten3=49; ignore=.; end;
if pubid=5995 then do; totten2=7; ignore=.; end;
if pubid=7465 then do; totten1=9; ignore=.; end;
if pubid=8827 then do; totten1=1; ignore=.; end;

do over totten;
  if totten>0 and ignore=1 then do; totten=-3; end;
end;

endsas;
```

TOTAL HOURS WORKED IN 19XX

Variables Created: CV_HOURS_WK_YR.80 – CV_HOURS_WK_YR.00

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all **employee-type** jobs in each calendar year. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

```
/* Section 1:  Creates a variable for total annual hours
worked at employee-type job during a given year using
Round 3 data. */

*** Organize hours for each job;
array starw (k) starw1-starw9;
array stopw (k) stopw1-stopw9;

/* shrs1 - starting hours       ehrs1 - ending hours   */
array shrs (k) shrs1-shrs9;
array ehrs (k) ehrs1-ehrs9;
array hours (k) hours1-hours9;
array hrck (k) hrck1-hrck9;

/* To the right of the array statements are the true
ranges of the pulled variables. Values of variables
outside the ranges will be represented by dots. */
array e23901 (k) e239011-e239019;                /*
        1-9 */
array e24501 (k) e245011-e245019;                /*
        1-6,9 */
array e34402 (k) e344021-e344029;                /*
        1-5 */
array e34403 (k) e344031-e344039;                /*
        1-3 */
array e34428 (k) e344281-e344289;                /*
        1-5 */
array e37904 (k) e379041-e379049;                /*
        1-5,7 */
array e38000f (k) e38000f1-e38000f9;       /* 1-5,7 */
array e38103 (k) e381031-e381039;                /*
        1-3,5 */
array e38105 (k) e381051-e381059;                /*
        1-3 */
array e59901 (k) e599011-e599019;                /*
        1-7 */
array e88000 (k) e880001-e880009;                /*
        1-7 */
array e88501 (k) e885011-e885019;                /*
        1-5 */
array e98402 (k) e984021-e984209;                /*
        1-6 */
array e98403 (k) e984031-e984039;                /*
        1-3 */
array e98429 (k) e984291-e984299;                /*
        1-3 */
array e100256 (k) e1002561-e1002569;       /* 1,2 */

/* Define the number of hours per week worked at the
start date and the end date of a job. Question E59901
decides whether the job was listed on the Round 1
roster, for jobs under 13 weeks. Question E88000 also
decides whether the job was listed on the Round 1
roster, but only for jobs that last longer then 13 weeks.
The code below defines the starting hours and ending
hours per week at those jobs, including overtime. */

do k=1 to 9;
/* Define starting hours for jobs shorter than 13 weeks
*/
  if e23901>-4 then do; shrs=e23901; end;
  if e23901>-4 and e24501>-4 then do;
     shrs=e23901+e24501; end;
  if e34402>-4 then do; shrs=e34402; end;
  if e34403>-4 then do; shrs=e34403; end;
  if e34403>-4 and e34428>-4 then do;
     shrs=e34403+e34428; end;

/* Define ending hours for jobs less than 13 weeks long
     */
  if e37904>-4 then do; ehrs=e37904; end;
/* Define ending hours for jobs longer than 13 weeks */
  if e38000f>-4 then do; ehrs=e38000f; end;
  if e38103>-4 then do; ehrs=e38103; end;
  if e38103>-4 and e38105>-4 then do;
     ehrs=e38103+e38105; end;

/* These are jobs from dli less than 13 weeks long */
  if e59901>-4 then do; shrs=e59901; end;
/* These are jobs from dli that are longer than 13 weeks
     */
  if e88000>-4 then do; shrs=e88000; end;
/* E88501 is an overtime question. */
  if e88501>-4 and e88000>-4 then do;
     shrs=e88501+e88000; end;
  if e98402>-4 then do; shrs=e98402; end;
  if e98429>-4 then do; shrs=e98429; end;
  if e98429>-4 and e98403>-4 then do;
       shrs=e98402+e98429; end;
  if e100256>-4 then do; shrs=e100256; end;
end;

/* The code below decides which hours per week total
(starting or ending) will be used in determining hours
worked per year.  We prefer using ending hours as the
measure.  Hours per week starts with a default number
(0), is updating if starting hours are available, and is
updated once more if ending hours are available. */

do over hours; hours=0; hrck=0; end;

* Take starting hours if reported (eliminates -4's);
do over shrs;
  if shrs=>0 then do; hours=shrs; end;
/* hrck marks when invalid answers are given to the
             hrs/wk questions */
  if -4<shrs<0 then do; hrck=shrs; end;
end;

* Write over if end hours reported;
do over ehrs;
  if ehrs=>0 then do; hours=ehrs; end;
  if -4<ehrs<0 then do; hrck=ehrs; end;
end;

/* Fix a person who has job start & stop data at loop 4
and the hours data in loop three. Other than the hours
data, there is no other employment data in loop 3. */
if pubid=157 then do; hours3=.; hours4=14; end;

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
     /*and so on through*/
array job9wks (i) wk9_1-wk9_1096;

/* Calculate cumulative weeks on each job for each
             year */
/* 1980 */
wks801=0; wks802=0; wks803=0; wks804=0;
             wks805=0; wks806=0; wks807=0; wks808=0;
             wks809=0;
ah801=0; ah802=0; ah803=0; ah804=0; ah805=0;
             ah806=0; ah807=0; ah808=0; ah809=0;

do i=1 to 52;
/* Update hours counter when respondent reports
             employment in weeks 1 to 52. */
     if job1wks=1 then do; wks801=wks801+1; end;
     if job2wks=1 then do; wks802=wks802+1; end;
     /*and so on through*/
     if job9wks=1 then do; wks809=wks809+1; end;
end;

do i=1 to 52;
  if job1wks ne -3 and job2wks ne -3 and job3wks ne -3
     and job4wks ne -3 and job5wks ne -3 and job6wks
     ne -3 and job7wks ne -3 and job8wks ne -3 and
     job9wks ne -3 then do;
     if hours1>0 then ah801=hours1*wks801;
     if hours2>0 then ah802=hours2*wks802;
     /*and so on through*/
     if hours9>0 then ah809=hours9*wks809;
  end;
end;
do i=1 to 52;
  if job1wks=-3 or job2wks=-3 or job3wks=-3 or
        job4wks=-3 or job5wks=-3 or job6wks=-3 or
        job7wks=-3 or job8wks=-3 or job9wks=-3 then
        do;
     ah801=-3; ah802=-3; ah803=-3; ah804=-3; ah805=-
          3;
     ah806=-3; ah807=-3; ah808=-3; ah809=-3;
     goto exit1;
  end;
end; exit1:

if ah801=>0 and ah802=>0 and ah803=>0 and
     ah804=>0 and ah805=>0 and ah806=>0 and
     ah807=>0 and ah808=>0 and ah809=>0 then do;
  tothrs80=ah801 + ah802 + ah803 + ah804 + ah805 +
     ah806 + ah807 + ah808 + ah809;
end;

if ah801=-3 or ah802=-3 or ah803=-3 or ah804=-3 or
     ah805=-3 or ah806=-3 or ah807=-3 or ah808=-3 or
     ah809=-3 then do;
  tothrs80=-3;
end;

if wks801>0 and hrck1<0 then tothrs80=hrck1;
if wks802>0 and hrck2<0 then tothrs80=hrck2;
     /*and so on through*/
if wks809>0 and hrck9<0 then tothrs80=hrck9;

/*** The same three sets of commands applied to the
1980 weeks variables are used for each subsequent
year.  However, this code is not shown here due to
space considerations.  Researchers needing the
complete code should contact NLS User Services.  The
variables and “do i” statements for each year are as
follows:
   1981   wksw811-wksw819   do i=53 to 104    exit2
   1982   wksw821-wksw829   do i=105 to 156   exit3
   1983   wksw831-wksw839   do i=157 to 209   exit4
   1984   wksw841-wksw849   do i=210 to 261   exit5
   1985   wksw851-wksw859   do i=262 to 313   exit6
   1986   wksw861-wksw869   do i=314 to 365   exit7
   1987   wksw871-wksw879   do i=366 to 417   exit8
  1988   wksw881-wksw889   do i=418 to 470    exit9
  1989   wksw891-wksw899   do i=471 to 522    exit10
  1990   wksw901-wksw909   do i=523 to 574    exit11
  1991   wksw911-wksw919   do i=575 to 626    exit12
  1992   wksw921-wksw929   do i=627 to 678    exit13
  1993   wksw931-wksw939   do i=679 to 730    exit14
  1994   wksw941-wksw949   do i=731 to 783    exit15
  1995   wksw951-wksw959   do i=784 to 835    exit16
  1996   wksw961-wksw969   do i=836 to 887    exit17
  1997   wksw971-wksw979   do i=888 to 939    exit18
  1998   wksw981-wksw989   do i=940 to 991    exit19
  1999   wksw991-wksw999   do i=992 to 1044   exit20
  2000   wksw001-wksw009   do i=1045 to 1098  exit21
                                              **/

/* Fill in -3's for cases where start/stop date is
          unknown/refused. */

do k=1 to 9;

/* start date invalid */
  if starw<0 and starw>-4 then do;
    if stopw>1 then do; tothrs80=-3; end;
    if stopw>52 then do; tothrs81=-3; end;
    if stopw>104 then do; tothrs82=-3; end;
    if stopw>156 then do; tothrs83=-3; end;
    if stopw>209 then do; tothrs84=-3; end;
    if stopw>261 then do; tothrs85=-3; end;
    if stopw>313 then do; tothrs86=-3; end;
    if stopw>365 then do; tothrs87=-3; end;
    if stopw>417 then do; tothrs88=-3; end;
    if stopw>470 then do; tothrs89=-3; end;
    if stopw>522 then do; tothrs90=-3; end;
    if stopw>574 then do; tothrs91=-3; end;
    if stopw>626 then do; tothrs92=-3; end;
    if stopw>678 then do; tothrs93=-3; end;
    if stopw>730 then do; tothrs94=-3; end;
    if stopw>783 then do; tothrs95=-3; end;
    if stopw>835 then do; tothrs96=-3; end;
    if stopw>887 then do; tothrs97=-3; end;
    if stopw>939 then do; tothrs98=-3; end;
    if stopw>991 then do; tothrs99=-3; end;
    if stopw>1044 then do; tothrs00=-3; end;
  end;

/* stop date invalid */
  if stopw<0 and stopw>-4 then do;
    if starw<53 then do; tothrs80=-3; end;
    if starw<105 then do; tothrs81=-3; end;
    if starw<157 then do; tothrs82=-3; end;
    if starw<210 then do; tothrs83=-3; end;
    if starw<262 then do; tothrs84=-3; end;
    if starw<314 then do; tothrs85=-3; end;
    if starw<366 then do; tothrs86=-3; end;
    if starw<418 then do; tothrs87=-3; end;
    if starw<471 then do; tothrs88=-3; end;
    if starw<523 then do; tothrs89=-3; end;
    if starw<575 then do; tothrs90=-3; end;
    if starw<627 then do; tothrs91=-3; end;
    if starw<679 then do; tothrs92=-3; end;
    if starw<731 then do; tothrs93=-3; end;
    if starw<784 then do; tothrs94=-3; end;
    if starw<836 then do; tothrs95=-3; end;
    if starw<888 then do; tothrs96=-3; end;
    if starw<940 then do; tothrs97=-3; end;
    if starw<991 then do; tothrs98=-3; end;
    if starw<1044 then do; tothrs99=-3; end;
    if starw<1096 then do; tothrs00=-3; end;
  end;
end;

if age14wk>intwk then do;
  tothrs80=-4; tothrs81=-4; /*and so on through*/
    tothrs00=-4;
end;

if e200=-3 then do;
  tothrs80=-3; tothrs81=-3; /*and so on through*/
    tothrs00=-3;
end;

if e200=-5 then do;
  tothrs80=-5; tothrs81=-5; /*and so on through*/
    tothrs00=-5;
end;

/* Section 2:  Sum up weeks worked to get a total
weeks worked during a given year.  Round1 goes from
14th b-day until Round2 interview date, Round2 goes
from Round1 interview date to Round2 interview date
*/

/* Begin by renaming the variables */
/* Round 3 infile created variable */
array tothrs (i) tothrs80-tothrs99 tothrs00;
/* Round 2 infile created variable */
array allhrs (i) allhrs80-allhrs99 allhrs00;
array r1hrs (i) r1hrs80-r1hrs99 r1hrs00;
array r2hrs (i) r2hrs80-r2hrs99 r2hrs00;
/* Total from all rounds created variable */
array r3hrs (i) r3hrs80-r3hrs99 r3hrs00;

do over tothrs; r3hrs=tothrs; tothrs=0; end;

/* Begin by splitting up periods where R2 and R3
exclusively collect hours worked information. Any
hours worked data collected before the R2 interview
year should be independent of data collected in R3. */
do over tothrs;
  if r2hrs>0 then do; tothrs=r2hrs; end;
  if r1hrs>0 and r2int=-5 and intwk>0 then do;
        tothrs=r1hrs; end;
  if r3hrs>0 then do; tothrs=r3hrs; end;
  if r2hrs>0 and r3hrs>0 and r2int>0 then do;
        tothrs=r2hrs+r3hrs; end;
  if r1hrs>0 and r3hrs>0 and r2int=-5 and intwk>0 then
        do; tothrs=r1hrs+r3hrs; end;
end;

/* Define neagtive values for the total created variable.
        */
do over tothrs;
  if -4<r1hrs<0 and r2int=-5 and intwk>0 then do;
        tothrs=r1hrs; end;
  if -4<r2hrs<0 then do; tothrs=r2hrs; end;
  if r3hrs=-1 or r3hrs=-2 or r3hrs=-3 or r3hrs=-5 then
        do; tothrs=r3hrs; end;
end;

/* The following "miss" people do not have a stop date
in the raw data. So, their weeks worked for years 1999
and 2000 are unknown. */
if miss=1 then do; tothrs99=-3; tothrs00=-3; end;

endsas;
```

TOTAL HOURS WORKED SINCE AGE 14

Variables Created: CV_HOURS_WK_EVER

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all **employee-type** jobs since turning 14. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures

```
/* Section 1:  Create a variable for total hours worked
at employee-type jobs since age 14 using round 3 data.
*/

*** Organize hours for each job;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

/* shrs1 - starting hours       ehrs1 - ending hours   */
array shrs (i) shrs1-shrs9;
array ehrs (i) ehrs1-ehrs9;
array hours (i) hours1-hours9;
array hrck (i) hrck1-hrck9;

/* To the right of the array statements are the true
ranges of the pulled variables. Values of variables
outside the ranges will be represented by dots. */
array e23901 (i) e239011-e239019;          /* 1-9 */
array e24501 (i) e245011-e245019;
        /* 1-6,9 */
array e34402 (i) e344021-e344029;          /* 1-5 */
array e34403 (i) e344031-e344039;          /* 1-3 */
array e34428 (i) e344281-e344289;          /* 1-5 */
array e37904 (i) e379041-e379049;          /* 1-5,7 */
array e38000f (i) e38000f1-e38000f9;       /* 1-5,7 */
array e38103 (i) e381031-e381039;
        /* 1-3,5 */
array e38105 (i) e381051-e381059;          /* 1-3 */
array e59901 (i) e599011-e599019;
        /* 1-7 */
array e88000 (i) e880001-e880009;
        /* 1-7 */
array e88501 (i) e885011-e885019;
        /* 1-5 */
array e98402 (i) e984021-e984209;          /* 1-6 */
array e98403 (i) e984031-e984039;
        /* 1-3 */
array e98429 (i) e984291-e984299;
        /* 1-3 */
array e100256 (i) e1002561-e1002569;       /* 1,2 */

/* Define the number of hours per week worked at the
start date and the end date of a job. */
```

```
do i=1 to 9;
 /* Define starting hours for jobs shorter than 13 weeks
          */
   if e23901>-4 then do; shrs=e23901; end;
   if e23901>-4 and e24501>-4 then do;
          shrs=e23901+e24501; end;
   if e34402>-4 then do; shrs=e34402; end;
   if e34403>-4 then do; shrs=e34403; end;
   if e34403>-4 and e34428>-4 then do;
          shrs=e34403+e34428; end;
 /* Define ending hours for jobs less than 13 weeks long
          */
   if e37904>-4 then do; ehrs=e37904; end;
 /* Define ending hours for jobs longer than 13 weeks */
   if e38000f>-4 then do; ehrs=e38000f; end;
   if e38103>-4 then do; ehrs=e38103; end;
   if e38103>-4 and e38105>-4 then do;
          ehrs=e38103+e38105; end;
 /* These are jobs from dli less than 13 weeks long */
   if e59901>-4 then do; shrs=e59901; end;
 /* These are jobs from dli longer than 13 weeks */
   if e88000>-4 then do; shrs=e88000; end;
 /* E88501 is an overtime question. */
   if e88501>-4 and e88000>-4 then do;
          shrs=e88501+e88000; end;
   if e98402>-4 then do; shrs=e98402; end;
   if e98429>-4 then do; shrs=e98429; end;
   if e98429>-4 and e98403>-4 then do;
          shrs=e98402+e98429; end;
   if e100256>-4 then do; shrs=e100256; end;
end;

/* The code below decides which hours per week total
(starting or ending) will be used in determining hours
worked per year.  We prefer using ending hours as the
measure, hours per week starts with a default number
(0), is updating if starting hours are available, and is
updated once more if ending hours are available. */

* Set default hours to zero (0);
do over hours; hours=0; hrck=0; end;

* Take starting hours if reported (eliminates -4's);
```

```
do over shrs;
   if shrs=>0 then do; hours=shrs; end;
      /* hrck marks when invalid answers are given to the
      hrs/wk questions */
   if -4<shrs<0 then do; hrck=shrs; end;
end;

* Write over if end hours reported;
do over ehrs;
   if ehrs=>0 then do; hours=ehrs; end;
   if -4<ehrs<0 then do; hrck=ehrs; end;
end;

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
   /*and so on through*/
array job9wks (i) wk9_1-wk9_1096;

/*Calculate cumulative weeks on each job for each
year*/
wks1=0; wks2=0; wks3=0; wks4=0; wks5=0; wks6=0;
              wks7=0; wks8=0; wks9=0;
thrs1=0; thrs2=0; thrs3=0; thrs4=0; thrs5=0; thrs6=0;
              thrs7=0; thrs8=0; thrs9=0;

if 0<age14wk<1096 then do;
   do i=age14wk to 1096;
      if job1wks=1 then do; wks1=wks1+1; end;
      if job2wks=1 then do; wks2=wks2+1; end;
      /*and so on through*/
      if job9wks=1 then do; wks9=wks9+1; end;
   end;
end;

if 0<age14wk<1096 then do;
   do i=age14wk to 1096;
      if job1wks ne -3 and job2wks ne -3 and job3wks ne
              -3 and job4wks ne -3 and job5wks ne -3 and
              job6wks ne -3 and job7wks ne -3  and
              job8wks ne -3 and job9wks ne -3 then do;
         thrs1=hours1*wks1;     thrs2=hours2*wks2;
         /*and so on through*/  thrs9=hours9*wks9;
      end;
   end;
   do i=age14wk to 1096;
      if job1wks=-3 or job2wks=-3 or job3wks=-3 or
          job4wks=-3 or job5wks=-3 or job6wks=-3 or
          job7wks=-3 or job8wks=-3 or job9wks=-3 then
          do;
         thrs1=-3; r3hrs14=-3; goto exit1;
      end;
   end;
end; exit1:
```

```
if thrs1=>0 and thrs2=>0 and thrs3=>0 and thrs4=>0
and thrs5=>0 and thrs6=>0 and thrs7=>0 and thrs8=>0
and thrs9=>0 then do;
   r3hrs14=thrs1 + thrs2 + thrs3 + thrs4 + thrs5 + thrs6 +
              thrs7 + thrs8 + thrs9;
end;

if thrs1=-3 or thrs2=-3 or thrs3=-3 or thrs4=-3 or thrs5=-
3 or thrs6=-3 or thrs7=-3 or thrs8=-3 or thrs9=-3 then
do;
   r3hrs14=-3;
end;

/* hrck is used when invalid hrs./wk answers are given
*/
if wks1>0 and hrck1<0 and r3hrs14>0 then
              r3hrs14=hrck1;
if wks2>0 and hrck2<0 and r3hrs14>0 then
              r3hrs14=hrck2;
   /*and so on through*/
if wks9>0 and hrck9<0 and r3hrs14>0 then
              r3hrs14=hrck9;

/* Remove invalid start/stop dates */
do i=1 to 9;
  if starw<0 and starw>-4 then do;
    if age14wk<53 and stopw>1 then r3hrs14=-3;
    if age14wk<105 and stopw>52 then r3hrs14=-3;
    if age14wk<157 and stopw>104 then r3hrs14=-3;
    if age14wk<210 and stopw>156 then r3hrs14=-3;
    if age14wk<262 and stopw>209 then r3hrs14=-3;
    if age14wk<314 and stopw>261 then r3hrs14=-3;
    if age14wk<366 and stopw>313 then r3hrs14=-3;
    if age14wk<418 and stopw>365 then r3hrs14=-3;
    if age14wk<471 and stopw>417 then r3hrs14=-3;
    if age14wk<523 and stopw>470 then r3hrs14=-3;
    if age14wk<575 and stopw>522 then r3hrs14=-3;
    if age14wk<627 and stopw>574 then r3hrs14=-3;
    if age14wk<679 and stopw>626 then r3hrs14=-3;
    if age14wk<731 and stopw>678 then r3hrs14=-3;
    if age14wk<784 and stopw>730 then r3hrs14=-3;
    if age14wk<836 and stopw>783 then r3hrs14=-3;
    if age14wk<888 and stopw>835 then r3hrs14=-3;
    if age14wk<940 and stopw>887 then r3hrs14=-3;
    if age14wk<991 and stopw>939 then r3hrs14=-3;
    if age14wk<1044 and stopw>991 then r3hrs14=-3;
    if age14wk<1096 and stopw>1044 then r3hrs14=-3;
  end;

  if stopw<0 and stopw>-4 then do;
    if age14wk<53 and starw<53 then r3hrs14=-3;
    if age14wk<105 and starw<105 then r3hrs14=-3;
    if age14wk<157 and starw<157 then r3hrs14=-3;
    if age14wk<210 and starw<210 then r3hrs14=-3;
    if age14wk<262 and starw<262 then r3hrs14=-3;
    if age14wk<314 and starw<314 then r3hrs14=-3;
    if age14wk<366 and starw<366 then r3hrs14=-3;
    if age14wk<418 and starw<418 then r3hrs14=-3;
    if age14wk<471 and starw<471 then r3hrs14=-3;
    if age14wk<523 and starw<523 then r3hrs14=-3;
    if age14wk<575 and starw<575 then r3hrs14=-3;
    if age14wk<627 and starw<627 then r3hrs14=-3;
    if age14wk<679 and starw<679 then r3hrs14=-3;
    if age14wk<731 and starw<731 then r3hrs14=-3;
    if age14wk<784 and starw<784 then r3hrs14=-3;
    if age14wk<836 and starw<836 then r3hrs14=-3;
    if age14wk<888 and starw<888 then r3hrs14=-3;
    if age14wk<940 and starw<940 then r3hrs14=-3;
    if age14wk<991 and starw<991 then r3hrs14=-3;
    if age14wk<1045 and starw<1045 then r3hrs14=-3;
    if age14wk<1097 and starw<1097 then r3hrs14=-3;
  end;
end;

do i=1 to 9;
   if stopw<0 and stopw>-4 and starw<=1096 then
            r3hrs14=-3;
end;

if e200=-5 then r3hrs14=-5;
if e200=-3 then r3hrs14=-3;

/* Section 2:  Take the created variables for total hours
worked since age 14 for R2 and R3 and combine them.
Since the created variables from R2 and R3 are only
calculated during the interview period this program
adds hours worked to get total hours worked since age
14. */

/* Initialize created variable for both rounds to zero. */
tothrs14=0;

/* Account for non-interview cases */
if e200=-5 then r3hrs14=-5;

/* By the construction of the R3 created variable, we
can simply add the two created variables from R2 and
R3 (or from R1 and R3 if not interviewed in R2) if they
are both postive. If one is positive and one is not, then
the positive value will be the total created variable for
both rounds. If neither is positive, then the total created
variable for both rounds will be zero. */

/* Round 2 positive */
if r2hrs14>0 then do; tothrs14=r2hrs14; end;
/* Round 1 positive, Round 2 non-interview case */
if r1hrs14>0 and r2int=-5 and intwk>0 then do;
            tothrs14=r1hrs14; end;

/* Round 3 positive */
if r3hrs14>0 then do; tothrs14=r3hrs14; end;

/* Both Round 2 and Round 3 positive */
if r2hrs14>0 and r3hrs14>0 then do;
            tothrs14=r2hrs14+r3hrs14; end;

/* Round 1 & Round 3 positive, non-interview case */
if r1hrs14>0 and r3hrs14>0 and r2int=-5 and intwk>0
            then do; tothrs14=r1hrs14+r3hrs14; end;

/* Define negative values for the created variables. */
if -4<r2hrs14<0 then tothrs14=-3;
if -4<r1hrs14<0 and r2int=-5 and intwk>0 then
            tothrs14=-3;
if -4<r3hrs14<0 then tothrs14=-3;
if r3hrs14=-5 then tothrs14=-5;

/* The "miss=1" cases need to have -3's in the c.v., but
the backreporters do not since their information does not
adversely affect any created variable. */
if miss=1 then tothrs14=-3;

endsas;
```

NUMBER OF JOBS HELD DURING 19XX

Variables Created: CV_TTL_JOBS_YR.80 – CV_TTL_JOBS_YR.00

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program calculates the number of employee-type jobs the respondent held during each calendar year. This variable is created only for respondents who have worked at least one week since age 14.

```
/* Section 1:  Create a variable for each year for
number of jobs held in that year using R3 information.
*/

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
   /*and so on through*/
array job9wks (i) wk9_1-wk9_1096;

array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

/** Indicate if worked at least one week on a job in a
given year **/

/* 1980 */
job801=0;          job802=0;          job803=0;
job804=0;          job805=0;          job806=0;
job807=0;          job808=0;          job809=0;

do i=1 to 52;
   if job1wks=-3 then do; job801=-3; end;
   if job2wks=-3 then do; job802=-3; end;
      /*and so on through*/
   if job9wks=-3 then do; job809=-3; end;
end;

do i=1 to 52;
   if job1wks=1 then do; job801=1; end;
   if job2wks=1 then do; job802=1; end;
      /*and so on through*/
   if job9wks=1 then do; job809=1; end;
end;

if job801 ne -3 and job802 ne -3 and job803 ne -3 and
job804 ne -3 and job805 ne -3 and job806 ne -3 and
job807 ne -3 and job808 ne -3 and job809 ne -3 then
do;
   njobs80=sum(job801, job802, job803, job804,
             job805, job806, job807, job808, job809);
end;
if job801=-3 or job802=-3 or job803=-3 or job804=-3
or job805=-3 or job806=-3 or job807=-3 or job808=-3
or job809=-3 then do;
   njobs80=-3;
end;

/* At this point the program loops through and creates
the same job variables for each year (1981–2000).  The
code is not shown here due to space constraints.  Users
needing the entire program should contact NLS User
Services.  The variables and “do i” statements for each
year are:
        1981     job811-job819; njobs81     do i=53 to 104
        1982     job821-job829; njobs82     do i=105 to 156
        1983     job831-job839; njobs83     do i=157 to 209
        1984     job841-job849; njobs84     do i=210 to 261
        1985     job851-job859; njobs85     do i=262 to 313
        1986     job861-job869; njobs86     do i=314 to 365
        1987     job871-job879; njobs87     do i=366 to 417
        1988     job881-job889; njobs88     do i=418 to 470
        1989     job891-job899; njobs89     do i=471 to 522
        1990     job901-job909; njobs90     do i=523 to 574
        1991     job911-job919; njobs91     do i=575 to 626
        1992     job921-job929; njobs92     do i=627 to 679
        1993     job931-job939; njobs93     do i=679 to 730
        1994     job941-job949; njobs94     do i=731 to 783
        1995     job951-job959; njobs95     do i=784 to 835
        1996     job961-job969; njobs96     do i=836 to 887
        1997     job971-job979; njobs97     do i=888 to 939
        1998     job981-job989; njobs98     do i=940 to 991
        1999     job991-job999; njobs99     do i=992 to 1044
        2000     job001-job009; njobs00     do i=1045 to 1096
                                            */

do i=1 to 9;
/* start date invalid */
   if starw<0 and starw>-4 then do;
      if stopw>1 then njobs80=njobs80+1;
      if stopw>52 then njobs81=njobs81+1;
      if stopw>104 then njobs82=njobs82+1;
      if stopw>156 then njobs83=njobs83+1;
      if stopw>209 then njobs84=njobs84+1;
      if stopw>261 then njobs85=njobs85+1;
      if stopw>313 then njobs86=njobs86+1;
      if stopw>365 then njobs87=njobs87+1;
      if stopw>417 then njobs88=njobs88+1;
      if stopw>470 then njobs89=njobs89+1;
      if stopw>522 then njobs90=njobs90+1;
      if stopw>574 then njobs91=njobs91+1;
      if stopw>626 then njobs92=njobs92+1;
```

```
    if stopw>678 then njobs93=njobs93+1;
    if stopw>730 then njobs94=njobs94+1;
    if stopw>783 then njobs95=njobs95+1;
    if stopw>835 then njobs96=njobs96+1;
    if stopw>887 then njobs97=njobs97+1;
    if stopw>939 and intwk>939 then
          njobs98=njobs98+1;
    if stopw>991 and intwk>991 then
          njobs99=njobs99+1;
    if stopw>1044 and intwk>1044 then
          njobs00=njobs00+1;
  end;

/*stop date invalid*/
  if stopw<0 and stopw>-4 then do;
    if starw<53 then njobs80=njobs80+1;
    if starw<105 then njobs81=njobs81+1;
    if starw<157 then njobs82=njobs82+1;
    if starw<210 then njobs83=njobs83+1;
    if starw<262 then njobs84=njobs84+1;
    if starw<314 then njobs85=njobs85+1;
    if starw<366 then njobs86=njobs86+1;
    if starw<418 then njobs87=njobs87+1;
    if starw<471 then njobs88=njobs88+1;
    if starw<523 then njobs89=njobs89+1;
    if starw<575 then njobs90=njobs90+1;
    if starw<627 then njobs91=njobs91+1;
    if starw<679 then njobs92=njobs92+1;
    if starw<731 then njobs93=njobs93+1;
    if starw<784 then njobs94=njobs94+1;
    if starw<836 then njobs95=njobs95+1;
    if starw<888 then njobs96=njobs96+1;
    if starw<940 then njobs97=njobs97+1;
    if starw<991 and intwk>939 then
          njobs98=njobs98+1;
    if starw<1044 and intwk>991 then
          njobs99=njobs99+1;
    if starw<1097 and intwk>1044 then
          njobs00=njobs00+1;
  end;
end;

*** Include valid skips;
if e200=-3 then do;
  njobs80=-3; njobs81=-3; /*and so on through*/
    njobs00=-3;
end;

if e200=-5 then do;
  njobs80=-5; njobs81=-5; /*and so on through*/
    njobs00=-5;
end;

/* Section 2:  Calculates the total number of jobs in a
given year by matching up the UID's from R2 and R3.
*/
```

```
/* Begin by renaming the infile c.v. variables */
array r1job (i) r1job80-r1job99 r1job00;
array r2job (i) r2job80-r2job99 r2job00;
array totjob (i) totjob80-totjob99 totjob00;
/* Round 3 created variable */
array njobs (i) njobs80-njobs99 njobs00;
/* Round 2 created variable */
array alljob (i) alljob80-alljob99 alljob00;
/* Total from both rounds created variable */
array r3job (i) r3job80-r3job99 r3job00;

do over totjob; r2job=alljob; r3job=njobs; totjob=0; end;

drop alljob80-alljob99 alljob00 njobs80-njobs99
njobs00;

/* Initialize created variable */
do over totjob; totjob=0; end;

/* Done because UID's are loaded in by din4
specifically by round */
drop UID1-UID9;

/* Begin by adding together the created variables from
both rounds. Jobs that are double counted will be
subtracted off later in the program. */
do over totjob;
  if r2job>0 then do; totjob=r2job; end;
  if r3job>0 then do; totjob=r3job; end;
  if r1job>0 and r2int=-5 and intwk>0 then do;
     totjob=r1job; end;
  if r2job>0 and r3job>0 then do; totjob=r2job+r3job;
     end;
  if r1job>0 and r3job>0 and r2int=-5 and intwk>0 then
     do; totjob=r1job+r3job; end;
end;

/* Define negative values for the total created variable.
*/
do over totjob;
  if -4<r2job<0 then totjob=-3;
  if -4<r1job<0 and r2int=-5 and intwk>0 then
     totjob=r1job;
  if -4<r3job<0 or r3job=-5 then totjob=r3job;
end;

/* Jobs that are worked in 1998 or 1999 could be double
counted since they are counted in each Round's created
variable programs. Here we compare the UIDs to see
which jobs are double counted. */
array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;

array match1 (i) match11-match19;
array match2 (i) match21-match29;
  /*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
     match5=0; match6=0; match7=0; match8=0;
     match9=0;
end;

/* If any UID from the first position in dli Round
matches with any UID in Round 3 */
do over r3uid;
  if r2uid1>0 and r3uid>0 and r2int>0 and intwk>0
     then do; if r2uid1=r3uid then match1=1; end;
  if r1uid1>0 and r3uid>0 and r2int=-5 and intwk>0
     then do; if r1uid1=r3uid then match1=1; end;
end;

/* The above code repeats for R1 jobs 2-7 and R2 jobs
2-9.  For example, r2uid2, r1uid2, and match2 are
substituted for r2uid1, r1uid1, and match 1.  Contact
NLS User Services for more information. */

/* Define "same98" as a counter variable that adds up
how many matching UIDs are in the respondent's UID
roster. This will be subtracted from the total number of
jobs created variable to avoid double counting the same
job. Begin by initializing the "same" variables to zero.
*/
same97=0;          same98=0;          same99=0;

/* Consider the respondents with a Round 1 interview
date in 1997, non-interview cases only. */
array r1j97 (i) r1j971-r1j979;
array r1j98 (i) r1j981-r1j989;
array r2j98 (i) r2j981-r2j989;
array r2j99 (i) r2j991-r2j999;
array job97 (i) job971-job979;
array job98 (i) job981-job989;
array job99 (i) job991-job999;

if r1int_y=1997 and r2int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 and job97=1 and r1j971=1 then do;
      same97=same97+1; end;
  end;
  do over match2;
    if match2=1 and job97=1 and r1j972=1 then do;
      same97=same97+1; end;
  end;                    /*and so on through*/
  do over match9;
    if match9=1 and job97=1 and r1j979=1 then do;
      same97=same97+1; end;
  end;
end;
```

```
/* Consider the respondents with a Round 1 interview
date in 1998, non-interview cases only. */
if r1int_y=1998 and r2int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 and job98=1 and r1j981=1 then do;
      same98=same98+1; end;
  end;
  do over match2;
    if match2=1 and job98=1 and r1j982=1 then do;
      same98=same98+1; end;
  end;                    /*and so on through*/
  do over match9;
    if match9=1 and job98=1 and r1j989=1 then do;
      same98=same98+1; end;
  end;
end;

/* Consider respondents with a R2 int date in 1998. */
if r2int_y=1998 then do;
  do over match1;
    if match1=1 and job98=1 and r2j981=1 then do;
      same98=same98+1; end;
  end;
  do over match2;
    if match2=1 and job98=1 and r2j982=1 then do;
      same98=same98+1; end;
  end;                    /*and so on through*/
  do over match9;
    if match9=1 and job98=1 and r2j989=1 then do;
      same98=same98+1; end;
  end;
end;

/* Consider respondents with a R2 int date in 1999. */
if r2int_y=1999 then do;
  do over match1;
    if match1=1 and job99=1 and r2j991=1 then do;
      same99=same99+1;      end;
  end;
  do over match2;
    if match2=1 and job99=1 and r2j992=1 then do;
      same99=same99+1; end;
  end;                    /*and so on through*/
  do over match9;
    if match9=1 and job99=1 and r2j999=1 then do;
      same99=same99+1; end;
  end;
end;

/* Now subtract the "same" count variable from the total
created variable computed above. */
if totjob97=>0 then do; totjob97=totjob97-same97; end;
if totjob98=>0 then do; totjob98=totjob98-same98; end;
if totjob99=>0 then do; totjob99=totjob99-same99; end;
```

```
/* These hand edit cases have 2 identical UID's in
either Round 2 or Round 3, which affects the same
variable above.  The first four hand edits have Round 3
matching UID's, the last two have Round 2 matching
UID's. */
if pubid=175 then totjob99=1;
if pubid=3196 then totjob99=1;
if pubid=960 then totjob99=1;
if pubid=6535 then totjob99=1;
if pubid=1586 then totjob98=2;
if pubid=2163 then totjob98=1;

endsas;
```

TOTAL NUMBER OF JOBS HELD SINCE AGE 14

Variables Created: CV_TTL_JOBS_EVER

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program calculates the total number of employee-type jobs held by the respondent since age 14. It is only created for respondents who have worked at least one week since age 14.

```
/* Section 1:  Create a variable for number of jobs held
since age 14 using R3 data.  Set default to zero. */

array job1wks (i) wk1_1-wk1_1096;
array job2wks (i) wk2_1-wk2_1096;
  /*and so on through*/
array job9wks (i) wk9_1-wk9_1096;

/** Indicate if worked at least one week on a given job
since age 14 **/
job1=0;  job2=0;  job3=0;  job4=0;  job5=0;
job6=0;  job7=0;  job8=0;  job9=0;

if age14wk>0 then do;
  do i=age14wk to 1096;
    if job1wks=1 then do; job1=1; end;
    if job2wks=1 then do; job2=1; end;
    /*and so on through*/
    if job9wks=1 then do; job9=1; end;
  end;
end;

njobs14=sum(job1, job2, job3, job4, job5, job6, job7,
job8, job9);

if starw1>-4 and starw1<0 and stopw1>0 and
    stopw1>age14wk then do; njobs14=njobs14+1;
    end;
if starw2>-4 and starw2<0 and stopw2>0 and
    stopw2>age14wk then do; njobs14=njobs14+1;
    end;
  /*and so on through*/
if starw9>-4 and starw9<0 and stopw9>0 and
    stopw9>age14wk then do; njobs14=njobs14+1;
    end;

if stopw1>-4 and stopw1<0 and starw1>0 and
    starw1=>age14wk then do; njobs14=njobs14+1;
    end;
if stopw2>-4 and stopw2<0 and starw2>0 and
    starw2=>age14wk then do; njobs14=njobs14+1;
    end;
  /*and so on through*/
if stopw9>-4 and stopw9<0 and starw9>0 and
    starw9=>age14wk then do; njobs14=njobs14+1;
    end;

if stopw1>-4 and stopw1<0 and starw1>0 then do;
    njobs14=-3; end;
if stopw2>-4 and stopw2<0 and starw2>0 then do;
    njobs14=-3; end;
  /*and so on through*/
if stopw9>-4 and stopw9<0 and starw9>0 then do;
    njobs14=-3; end;

if e200=-3 then do; njobs14=-3; end;
if e200=-5 then do; njobs14=-5; end;

/* Section 2:  Calculate the total number of jobs  by
matching up the UID's from Round 2 and Round 3. */

/* Rename infile c.v., drop replaced names */
r3job14=njobs14;          totjob14=0;
drop njobs14;

/* Begin by adding together the created variables from
both rounds.  Jobs that are double counted will be
subtracted off later in the program. */
if r1job14>0 and r2int=-5 and intwk>0 then do;
    totjob14=r1job14; end;
if r2job14>0 then do; totjob14=r2job14; end;
if r3job14>0 then do; totjob14=r3job14; end;
if r2job14>0 and r3job14>0 then do;
    totjob14=r2job14+r3job14; end;
if r1job14>0 and r3job14>0 and r2int=-5 and intwk>0
    then do; totjob14=r1job14+r3job14; end;

/* Define negative values for the total created variable.
*/
if -4<r2job14<0 then totjob14=-3;
if -4<r3job14<0 or r3job14=-5 then totjob14=r3job14;

/* Jobs worked in 1998 or 1999 have the possibility of
being double counted since they are counted in each
Round's created variable programs. Here we compare
the UID's to see which jobs are double counted. */

array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;

array match1 (i) match11-match19;
array match2 (i) match21-match29;
  /*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
    match5=0; match6=0; match7=0; match8=0;
    match9=0;
end;

/* If any UID from the first position in dli Round
matches with any UID in Round 3 */
do over r3uid;
  if r2uid1>0 and r3uid>0 and r2int>0 and intwk>0
    then do; if r2uid1=r3uid then match1=1; end;
  if r1uid1>0 and r3uid>0 and r2int=-5 and intwk>0
    then do; if r1uid1=r3uid then match1=1; end;
end;

/* The above code repeats for R1 jobs 2-7 and R2 jobs
2-9.  For example, r2uid2, r1uid2, and match2 are
substituted for r2uid1, r1uid1, and match 1.  Contact
NLS User Services for more information. */

/* Define "same" as a counted variable that adds up how
many matching UIDs are in the respondent's UID roster.
This will be subtracted from the total number of jobs
created variable to avoid double counting the same job.
Begin by initializing the "same" variables to zero. */
same=0;

do over match1;
  if match1=1 then do; same=same+1; end;
end;
do over match2;
  if match2=1 then do; same=same+1; end;
end;
    /*and so on through*/
do over match9;
  if match9=1 then do;
    same=same+1;
  end;
end;

/* Now subtract the "same" count variable from the total
created variable computed above. */
if totjob14=>0 then do; totjob14=totjob14-same; end;

endsas;
```

HOURLY RATE OF PAY, HOURLY MONETARY COMPENSATION AND JOB LENGTH < 13 WEEKS

Variables Created: CV_HRLY_PAY
CV_HRLY_COMPENSATION
CV_JOB<13_WKS

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
e200a	YEMP-200A	e2120011-e2120016	YEMP-21200.01~000001-~000006
e6001-e60017	YEMP-600.01-.17	e2120021-e2120026	YEMP-21200.02~000001-~000006
e192001-e192009	YEMP-19200.01-.09	e2120031-e2120036	YEMP-21200.03~000001-~000006
e207001-e207009	YEMP-20700.01-.09	e2120041-e2120046	YEMP-21200.04~000001-~000006
e229001-e229009	YEMP-22900.01-.09	e2120051-e2120056	YEMP-21200.05~000001-~000006
e230001-e230008	YEMP-23000.01-.08	e2120061-e2120066	YEMP-21200.06~000001-~000006
e232001-e232003	YEMP-23200.01-.03	e2120071-e2120076	YEMP-21200.07~000001-~000006
e239001	YEMP-23900.01	e2120081-e2120086	YEMP-21200.08~000001-~000006
e239011-e239019	YEMP-23901.01-.09	e2120091-e2120096	YEMP-21200.09~000001-~000006
e245011-e245019	YEMP-24501.01-.09	e2260411-e2260419, e2260410	YEMP-22604.01~000001-~000009, ~000010
e245021-e245029	YEMP-24502.01-.09	e2260421-e2260429, e2260420	YEMP-22604.02~000001-~000009, ~000010
e245141-e245145	YEMP-24514.01-.05	e2260431-e2260439, e2260430	YEMP-22604.03~000001-~000009, ~000010
e24514b1	YEMP-24514B.01	e2260441-e2260449, e2260440	YEMP-22604.04~000001-~000009, ~000010
e334001-e334005	YEMP-33400.01-.05	e1022511-e1022519, e1022510	YEMP-100225.01~000001-~000009, ~000010
e335001-e335004	YEMP-33500.01-.04	e1022521-e1022529, e1022520	YEMP-100225.02~000001-~000009, ~000010
e336001-e336004	YEMP-33600.01-.04	e1022531-e1022539, e1022530	YEMP-100225.03~000001-~000009, ~000010
e344021-e344025	YEMP-34402.01-.05	e1022541-e1022549, e1022540	YEMP-100225.04~000001-~000009, ~000010
e34402b1-e34402b3	YEMP-34402AB.01-.03	e381161-e381165	YEMP-38116.01-.05
e344031-e344033	YEMP-34403.01-.03	e382011-e382015	YEMP-38201.01-.05
e344041-e344043	YEMP-34404.01-.03	e382021-e382023	YEMP-38202.01-.03
e34413c1-e34413c3	YEMP-34413C.01-.03	e382111-e382113	YEMP-38211.01-.03
e34413e1, e34413e2	YEMP-34413E.01, .02	e383131-e383139	YEMP-38313.01-.09
e344281-e344285	YEMP-34428.01-.05	e383291-e383297	YEMP-38329.01-.07
e344291-e344295	YEMP-34429.01-.05	e38329b1-e38329b3	YEMP-38329B.01-.03
e344301, e344302	YEMP-34430.01, .02	e38329d1-e38329d3	YEMP-38329D.01-.03
e356001-e356004	YEMP-35600.01-.04	e383301, e383302	YEMP-38330.01, .02
e361001-e361003	YEMP-36100.01-.03	e3840711-e3840715	YEMP-38407.01.01-.05
e362001-e362005	YEMP-36200.01-.05	e3840722-e3840725	YEMP-38407.02.02-.05
e2160011-e2160015	YEMP-21600.01.02-.06	e3840731	YEMP-38407.03.01
e2160021-e2160025	YEMP-21600.02.02-.06	e3841611-e3841615	YEMP-38416.01.01-.05
e2160031-e2160034	YEMP-21600.03.02-.05	e3841622-e3841625	YEMP-38416.02.02-.05
e2160041, e2160043	YEMP-21600.04.02, .04	e3841631	YEMP-38416.03.01
e2160051, e2160052	YEMP-21600.05.02, .03	e582011-e582017	YEMP-58201.01-.07
e2160071	YEMP-21600.07.02	e599001-e599007	YEMP-59900.01-.07
e2160081, e2160083	YEMP-21600.08.02, .04	e831001-e831007	YEMP-83100.01-.07
e2250011-e2250015	YEMP-22500.01.02-.06	e868001-e868005	YEMP-86800.01-.05
e2250021-e2250025	YEMP-22500.02.02-.06	e869001-e869007	YEMP-86900.01-.07
e2250031-e2250034	YEMP-22500.03.02-.05	e871001-e871004	YEMP-87100.01-.04
e2250041, e2250043	YEMP-22500.04.02, .04	e880001-e880007	YEMP-88000.01-.07
e2250051, e2250052	YEMP-22500.05.02, .03	e885011-e885015	YEMP-88501.01-.05
e2250071	YEMP-22500.07.02	e885021-e885025	YEMP-88502.01-.05
e2250081, e2250083	YEMP-22500.08.02, .04	e885121-e885125	YEMP-88512.01-.05
e226091-e226094	YEMP-22609.01.01-.04.01	e973001-e973006	YEMP-97300.01-.06
e226101, e226102	YEMP-22610.01.02, .02.02	e974001-e974005	YEMP-97400.01-.05
e226112-e226114	YEMP-22611.02.03-.04.03	e975001-e975006	YEMP-97500.01-.06
e226121	YEMP-22612.01.04	e983001-e983003	YEMP-98300.01-.03
e226131, e226132	YEMP-22613.01.05, .02.05	e984021-e984026	YEMP-98402.01-.06
e226143	YEMP-22614.03.06	e98402d1-e98402d4	YEMP-98402D.01-.04
e226151	YEMP-22615.01.07	e984031-e984033	YEMP-98403.01-.03
e226161	YEMP-22616.01.08	e98404a1-e98404a3	YEMP-98404A.01-.03
e226171	YEMP-22617.01.09	e984141-e984143	YEMP-98414.01-.03
e226261-e226264	YEMP-22626.01.01-.04.01	e98414b1	YEMP-98414B.01
e226271	YEMP-22627.01.04	e984291-e984293	YEMP-98429.01-.03

e226281, e226282	YEMP-22628.01.05, .02.05
e226293	YEMP-22629.03.06
e226301	YEMP-22630.01.07
e226311	YEMP-22631.01.08
e226321	YEMP-22632.01.09
e377011-e377019	YEMP-37701.01-.09
e37901b1-e37901b9	YEMP-37901B.01-.09
e3800b1-e3800b7	YEMP-38000B.01-.07
e3800f1-e3800f7	YEMP-38000F.01-.07
e380011-e380017	YEMP-38001.01-.07
e380021-e380025	YEMP-38002.01-.05
e380031-e380035	YEMP-38003.01-.05
e380121-e380125	YEMP-38012.01-.05
e380131-e380137	YEMP-38013.01-.07
e380141-e380147	YEMP-38014.01-.07
e380231-e380237	YEMP-38023.01-.07
e380271-e380273	YEMP-38027.01-.03
e381011-e381019	YEMP-38101.01-.09
e381021-e381025	YEMP-38102.01-.05
e381031-e381035	YEMP-38103.01-.05
e381041-e381045	YEMP-38104.01-.05
e381051-e381053	YEMP-38105.01-.03
e381061-e381069	YEMP-38106.01-.09
e381071-e381075	YEMP-38107.01-.05
e995001-e995006	YEMP-99500.01-.06
e1000001-e1000003	YEMP-100000.01-.03
e1001001-e1001004	YEMP-100100.01-.04
e1020511-e1020515	YEMP-100205.01.02-.06
e1020521-e1020525	YEMP-100205.02.02-.06
e1020531-e1020535	YEMP-100205.03.02-.06
e1020541-e1020544	YEMP-100205.04.02-.05
e1021411-e1021415	YEMP-100214.01.02-.06
e1021421-e1021425	YEMP-100214.02.02-.06
e1021431-e1021435	YEMP-100214.03.02-.06
e1021441-e1021444	YEMP-100214.04.02-.05
e1021451-e1021452	YEMP-100214.05.02, .03
e102301-e102304	YEMP-100230.01.01-.04.01
e102311-e102314	YEMP-100231.01.02-.04.02
e102321	YEMP-100232.01.03
e102341	YEMP-100234.01.05
e102361	YEMP-100236.01.07
e102481-e102484	YEMP-100248.01.01-.04.01
e102501	YEMP-100250.01.05
e102521	YEMP-100252.01.07
e584011-e584019	YEMP-58401.01-.09
euid1-euid9	YEMP_UID.01-.09
norcid	NORCID

Codes for Created Variable

Note that hourly rate of pay is reported with two implied decimal places.

This program creates the hourly rate of pay for NLSY97 respondents. The hourly rate of pay is constructed from stop date information for respondents whose job lasted more than 13 weeks. For all other respondents the start wage is used.

In addition, this program creates an hourly monetary compensation variable for NLSY97 respondents. This variable that includes information about all compensation received by the respondent, such as tips, bonuses, commissions, overtime, etc., in the calculation. Hourly monetary compensation differs from hourly rate of pay variable, which calculates only the base pay rate. This variable is constructed from stop date information for respondents with jobs longer than 13 weeks and start date information for other.

Finally, a variable indicating whether the jobs lasted more than 13 weeks is also created. There are up to 9 jobs reported, so each variable is created for 9 jobs.

```
/* DECLARING THE ARRAYS TO BE USED LATER IN THE PROGRAM*/
array e19200 e192001-e192009;      array e37901b e37901b1-e37901b9;   array e59900 e599001-e599009;
array e22900 e229001-e229009;      array e23000 e230001-e230009;      array e23200 e232001-e232009;
array e23900 e239001-e239009;      array e33400 e334001-e334009;      array e34402 e344021-e344029;
array e34402b e34402b1-e34402b9;   array e33600 e336001-e336009;      array e33500 e335001-e335009;
array e34428 e344281-e344289;      array e34430 e344301-e344309;      array e36100 e361001-e361009;
array e36200 e362001-e362009;      array e20700 e207001-e207009;      array e24501 e245011-e245019;
array e24514 e245141-e245149;      array e24514b e24514b1-e24514b9;   array e38013 e380131-e380139;
array e38014 e380141-e380149;      array e38023 e380231-e380239;      array e38106 e381061-e381069;
array e38107 e381071-e381079;      array e38116 e381161-e381169;      array e3800b e3800b1-e3800b9;
array e38027 e380271-e380279;      array e23901 e239011-e239019;      array e3800f e3800f1-e3800f9;
array e38101 e381011-e381019;      array e38102 e381021-e381029;      array e38103 e381031-e381039;
array e38201 e382011-e382019;      array e38105 e381051-e381059;      array e24502 e245021-e245029;
array e38202 e382021-e382029;      array e38211 e382111-e382119;      array e38211b e38211b1-e38211b9;
array e34400 e344001-e344009;      array e38313 e383131-e383139;      array e83100 e831001-e831009;
array e86800 e868001-e868009;      array e86900 e869001-e869009;      array e87100 e871001-e871009;
array e87800 e878001-e878009;       array e88000 e880001-e880009;       array e97300 e973001-e973009;
array e97400 e974001-e974009;       array e97500 e975001-e975009;       array e98300 e983001-e983009;
array e98402 e984021-e984029;       array e98402d e98402d1-e98402d9;    array e98429 e984291-e984299;
array e98429l e98429l1-e98429l9;    array e100000 e1000001-e1000009;    array e100100 e1001001-e1001009;
array e99500 e995001-e995009;       array e88501 e885011-e885019;       array e88502 e885021-e885029;
array e88512 e885121-e885129;       array e88512b e88512b1-e88512b9;    array e34403 e344031-e344039;
array e34404 e344041-e344049;       array e34413c e34413c1-e34413c9;    array e34413e e34413e1-e34413e9;
array e98403 e984031-e984039;       array e98404a e98404a1-e98404a9;    array e98414 e984141-e984149;
array e98414b e98414b1-e98414b9;    array e38001 e380011-e380019;       array e38002 e380021-e380029;
array e38003 e380031-e380039;       array e38012 e380121-e380129;       array e38012b e38012b1-e38012b9;
array e38329b e38329b1-e38329b9;    array e38329d e38329d1-e38329d9;    array e38329 e383291-e383299;
array e38330 e383301-e383309;       array e35600 e356001-e356009;       array e22609 e226091-e226099;
array e22610 e226101-e226109;       array e22611 e226111-e226119;       array e22612 e226121-e226129;
array e22613 e226131-e226139;       array e22614 e226141-e226149;       array e22615 e226151-e226159;
array e22616 e226161-e226269;       array e22617 e226171-e226179;       array e22626 e226261-e226269;
array e22627 e226271-e226279;       array e22628 e226281-e226289;       array e22629 e226291-e226299;
array e22630 e226301-e226309;       array e22631 e226311-e226319;       array e22632 e226321-e226329;
array e100230 e102301-e102309;      array e100231 e102311-e102319;      array e100232 e102321-e102329;
array e100233 e102331-e102339;      array e100234 e102341-e102349;      array e100235 e102351-e102359;
array e100236 e102361-e102369;      array e100237 e102371-e102379;      array e100239 e102391-e102399;
array e100248 e102481-e102489;      array e100249 e102491-e102499;      array e100250 e102501-e102509;
array e100251 e102511-e102519;      array e100252 e102521-e102529;      array e100253 e102531-e102539;
array e100254 e102541-e102549;      array e37701 e377011-e377019;       array e58401 e584011-e584019;
array e58201 e582011-e582019;       array euid euid1-euid9;             array e38116b e38116b1-e38116b9;

array e225001 e2250011 e2250021 e2250031 e2250041 e2250051 e2250061 e2250071 e2250081 e2250091;
array e225002 e2250012 e2250022 e2250032 e2250042 e2250052 e2250062 e2250072 e2250082 e2250092;
array e225003 e2250013 e2250023 e2250033 e2250043 e2250053 e2250063 e2250073 e2260083 e2250093;
array e225004 e2250014 e2250024 e2250034 e2250044 e2250054 e2250064 e2250074 e2250084 e2250094;
array e225005 e2250015 e2250025 e2250035 e2250045 e2250055 e2250065 e2250075 e2250085 e2250095;

array e216001 e2160011 e2160021 e2160031 e2160041 e2160051 e2160061 e2160071 e2160081 e2160091;
array e216002 e2160012 e2160022 e2160032 e2160042 e2160052 e2160062 e2160072 e2160082 e2160092;
array e216003 e2160013 e2160023 e2160033 e2160043 e2160053 e2160063 e2160073 e2160083 e2160093;
array e216004 e2160014 e2160024 e2160034 e2160044 e2160054 e2160064 e2160074 e2160084 e2160094;
array e216005 e2160015 e2160025 e2160035 e2160045 e2160055 e2160065 e2160075 e2160085 e2160095;

array e212001 e2120011 e2120021 e2120031 e2120041 e2120051 e2120061 e2120071 e2120081 e2120091;
array e212002 e2120012 e2120022 e2120032 e2120042 e2120052 e2120062 e2120072 e2120082 e2120092;
array e212003 e2120013 e2120023 e2120033 e2120043 e2120053 e2120063 e2120073 e2120083 e2120093;
array e212004 e2120014 e2120024 e2120034 e2120044 e2120054 e2120064 e2120074 e2120084 e2120094;
array e212005 e2120015 e2120025 e2120035 e2120045 e2120055 e2120065 e2120075 e2120085 e2120095;
array e212006 e2120016 e2120026 e2120036 e2120046 e2120056 e2120066 e2120076 e2120086 e2120096;

array e384161 e3841611 e3841621 e3841631 e3841641 e3841651 e3841661 e3841671 e3841681 e3841691;
array e384162 e3841612 e3841622 e3841632 e3841642 e3841652 e3841662 e3841672 e3841682 e3841692;
array e384163 e3841613 e3841623 e3841633 e3841643 e3841653 e3841663 e3841673 e3841683 e3841693;
array e384164 e3841614 e3841624 e3841634 e3841644 e3841654 e3841664 e3841674 e3841684 e3841694;
array e384165 e3841615 e3841625 e3841635 e3841645 e3841655 e3841665 e3841675 e3841685 e3841695;

array e384071 e3840711 e3840721 e3840731 e3840741 e3840751 e3840761 e3840771 e3840781 e3840791;
array e384072 e3840712 e3840722 e3840732 e3840742 e3840752 e3840762 e3840772 e3840782 e3840792;
array e384073 e3840713 e3840723 e3840733 e3840743 e3840753 e3840763 e3840773 e3840783 e3840793;
array e384074 e3840714 e3840724 e3840734 e3840744 e3840754 e3840764 e3840774 e3840784 e3840794;
array e384075 e3840715 e3840725 e3840735 e3840745 e3840755 e3840765 e3840775 e3840785 e3840795;
array e102051 e1020511 e1020521 e1020531 e1020541 e1020551 e1020561 e1020571 e1020581 e1020591;
array e102052 e1020512 e1020522 e1020532 e1020542 e1020552 e1020562 e1020572 e1020582 e1020592;
array e102053 e1020513 e1020523 e1020533 e1020543 e1020553 e1020563 e1020573 e1020583 e1020593;
array e102054 e1020514 e1020524 e1020534 e1020544 e1020554 e1020564 e1020574 e1020584 e1020594;
array e102055 e1020515 e1020525 e1020535 e1020545 e1020555 e1020565 e1020575 e1020585 e1020595;

array e102141 e1021411 e1021421 e1021431 e1021441 e1021451 e1021461 e1021471 e1021481 e1021491;
array e102142 e1021412 e1021422 e1021432 e1021442 e1021452 e1021462 e1021472 e1021482 e1021492;
array e102143 e1021413 e1021423 e1021433 e1021443 e1021453 e1021463 e1021473 e1021483 e1021493;
array e102144 e1021414 e1021424 e1021434 e1021444 e1021454 e1021464 e1021474 e1021484 e1021494;
array e102145 e1021415 e1021425 e1021435 e1021445 e1021455 e1021465 e1021475 e1021485 e1021495;

array e226041 e2260411 e2260421 e2260431 e2260441 e2260451 e2260461 e2260471 e2260481 e2260491;
array e226042 e2260412 e2260422 e2260432 e2260442 e2260452 e2260462 e2260472 e2260482 e2260492;
array e226043 e2260413 e2260423 e2260433 e2260443 e2260453 e2260463 e2260473 e2260083 e2260493;
array e226044 e2260414 e2260424 e2260434 e2260444 e2260454 e2260464 e2260474 e2260484 e2260494;
array e226045 e2260415 e2260425 e2260435 e2260445 e2260455 e2260465 e2260475 e2260485 e2260495;
array e226046 e2260416 e2260426 e2260436 e2260446 e2260456 e2260466 e2260476 e2260486 e2260496;
array e226047 e2260417 e2260427 e2260437 e2260447 e2260457 e2260467 e2260477 e2260487 e2260497;
array e226048 e2260418 e2260428 e2260438 e2260448 e2260458 e2260468 e2260478 e2260088 e2260498;
array e226049 e2260419 e2260429 e2260439 e2260449 e2260459 e2260469 e2260479 e2260489 e2260499;
array e226040 e2260410 e2260420 e2260430 e2260440 e2260450 e2260460 e2260470 e2260480 e2260490;

array e102251 e1022511 e1022521 e1022531 e1022541 e1022551 e1022561 e1022571 e1022581 e1022591;
array e102252 e1022512 e1022522 e1022532 e1022542 e1022552 e1022562 e1022572 e1022582 e1022592;
array e102253 e1022513 e1022523 e1022533 e1022543 e1022553 e1022563 e1022573 e2260083 e1022593;
array e102254 e1022514 e1022524 e1022534 e1022544 e1022554 e1022564 e1022574 e1022584 e1022594;
array e102255 e1022515 e1022525 e1022535 e1022545 e1022555 e1022565 e1022575 e1022585 e1022595;
array e102256 e1022516 e1022526 e1022536 e1022546 e1022556 e1022566 e1022576 e1022586 e1022596;
array e102257 e1022517 e1022527 e1022537 e1022547 e1022557 e1022567 e1022577 e1022587 e1022597;
array e102258 e1022518 e1022528 e1022538 e1022548 e1022558 e1022568 e1022578 e2260088 e1022598;
array e102259 e1022519 e1022529 e1022539 e1022549 e1022559 e1022569 e1022579 e1022589 e1022599;
array e102250 e1022510 e1022520 e1022530 e1022540 e1022550 e1022560 e1022570 e1022580 e1022590;


/***************** Create the hourly wages.  First, the start wages, then the end wages
******************/
/*****************Section 1:  Start wages for the youth******************/

/* for each time unit, the information could either be given in the earlier part (from e19200) or in the latet part (from
        e83100). and it also depends on if the respondent has other compensation or not. */
/*set up the hourly wage for youths who report an hourly wage*/
hrwage01=-4;    hrwage02=-4;    hrwage03=-4;    hrwage04=-4;    hrwage05=-4;    hrwage06=-4;
hrwage07=-4;    hrwage08=-4;    hrwage09=-4;

array hrwage hrwage01 hrwage02 hrwage03 hrwage04 hrwage05 hrwage06 hrwage07 hrwage08 hrwage09;

do i=1 to 9;
  if e19200[i]=1 then do;
    if (e22900[i]>=0) then hrwage[i]=e22900[i];
    if (e23000[i]>=0) then hrwage[i]=e23000[i];
    if (e22900[i]=-2 or e23000[i]=-2) then hrwage[i]=e23200[i];
    if (e22900[i]=-3 or e23000[i]=-3) then hrwage[i]=e23200[i];
    if (e22900[i]=-1 or e23000[i]=-1) then hrwage[i]=-1;
    if e23900[i]>=0 then hrwage[i]=e23900[i];
  end;
  if e83100[i]=1 then do;
     if (e86800[i]>=0) then hrwage[i]=e86800[i];
     if (e86900[i]>=0) then hrwage[i]=e86900[i];
     if (e86800[i]=-2 or e86900[i]=-2) then hrwage[i]=e87100[i];
     if (e86800[i]=-3 or e86900[i]=-3) then hrwage[i]=e87100[i];
     if (e86800[i]=-1 or e86900[i]=-1) then hrwage[i]=-1;
     if e87800[i]>=0 then hrwage[i]=e87800[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in daily units.*/
daily01=-4;     daily02=-4;     daily03=-4;     daily04=-4;     daily05=-4;     daily06=-4;
daily07=-4;     daily08=-4;     daily09=-4;

array daily daily01 daily02 daily03 daily04 daily05 daily06 daily07 daily08 daily09;

do i=1 to 9;       /* daily start wage divided by the number of hours worked per day*/
   if e19200[i]=2 then do;
     /*no compensation*/
     if (e33400[i]>=0 and e34402[i]>0 and e34402b[i]>0) then daily[i]=(e33400[i]*e34402b[i]/e34402[i]);
     if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e34402b[i]>0) then
            daily[i]=(e33600[i]*e34402b[i]/e34402[i]);
     if e34400[i]>=0 and e34402[i]>0 and e34402b[i]>0 then daily[i]=e34400[i]*e34402b[i]/e34402[i];
     /* missing value*/  if -4<e34402[i]<0 then daily[i]=e34402[i];
     /*received compensation*/    /*without overtime*/
     if (e33500[i]>=0 and e34402[i]>0 and e34402b[i]>0) then daily[i]=(e33500[i]*e34402b[i]/e34402[i]);
     if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e34402b[i]>0) then
            daily[i]=(e33600[i]*e34402b[i]/e34402[i]);
     if e34400[i]>=0 and e34402[i]>0 and e34402b[i]>0 then daily[i]=e34400[i]*e34402b[i]/e34402[i];
     /*missing value*/    if -4<e34402[i]<0 then daily[i]=e34402[i];
     if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then daily[i]=-3;
     if -4<e34402b[i]<0 then daily[i]=e34402b[i];
     if e34402b[i]=0 then daily[i]=-3;
     /*with overtime*/
     if (e33500[i]>=0 and e34428[i]>0 and e34430[i]>0) then daily[i]=(e33500[i]*e34430[i]/e34428[i]);
     if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0 and e34430[i]>0) then
            daily[i]=(e33600[i]*e34430[i]/e34428[i]);
     if e34400[i]>=0 and e34428[i]>0 and e34430[i]>0 then daily[i]=e34400[i]*e34430[i]/e34428[i];
     /*missing value*/   if -4<e34428[i]<0 then daily[i]=e34428[i];
     if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then daily[i]=-3;
     if -4<e34430[i]<0 then daily[i]=e34430[i];
     if e34430[i]=0 then daily[i]=-3;
     /*if still paid hourly...*/
     if e36200[i]>=0 then daily[i]=e36200[i];
     if e36100[i]>=0 then daily[i]=e36100[i];
   end;

   if e83100[i]=2 then do;
     /*no compensation*/
     if (e97300[i]>=0 and e98402[i]>0 and e98402d[i]>0) then daily[i]=(e97300[i]*e98402d[i]/e98402[i]);
     if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e98402d[i]>0) then
            daily[i]=(e97500[i]*e98402d[i]/e98402[i]);
     if e98300[i]>=0 and e98402[i]>0 and e98402d[i]>0 then daily[i]=e98300[i]*e98402d[i]/e98402[i];
     /* missing value*/   if -4<e98402[i]<0 then daily[i]=e98402[i];
     /*received compensation*/    /*with overtime */
```

```
    if (e97400[i]>=0 and e98429[i]>0 and e98429l[i]>0) then daily[i]=(e97400[i]*e98429l[i]/e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0 and e98429l[i]>0) then
         daily[i]=(e97500[i]*e98429l[i]/e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 and e98429l[i]>0 then daily[i]=e98300[i]*e98429l[i]/e98429[i];
    /*missing value*/   if -4<e98429[i]<0 then daily[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then daily[i]=-3;
    if -4<e98429l[i]<0 then daily[i]=e98429l[i];
    if e98429l[i]=0 then daily[i]=-3;
    /* without overtime */
    if (e97400[i]>=0 and e98402[i]>0 and e98402d[i]>0) then daily[i]=(e97400[i]*e98402d[i]/e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e98402d[i]>0) then
         daily[i]=(e97500[i]*e98402d[i]/e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 and e98402d[i]>0 then daily[i]=e98300[i]*e98402d[i]/e98402[i];
    /*missing value*/   if -4<e98402[i]<0 then daily[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then daily[i]=-3;
    if -4<e98402d[i]<0 then daily[i]=e98402d[i];
    if e98402d[i]=0 then daily[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then daily[i]=e100100[i];
    if e100000[i]>=0 then daily[i]=e100000[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in weekly units.*/
weekly01=-4;    weekly02=-4;    weekly03=-4;    weekly04=-4;    weekly05=-4;    weekly06=-4;
weekly07=-4;    weekly08=-4;    weekly09=-4;

array weekly weekly01 weekly02 weekly03 weekly04 weekly05 weekly06 weekly07 weekly08 weekly09;

do i=1 to 9; /* weekly start wage divided by the number of hours worked per week*/
  if e19200[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0,-2) then do;
    /*no compensation*/
    if (e33400[i]>=0 and e34402[i]>0) then weekly[i]=(e33400[i]/e34402[i]);
    if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then weekly[i]=(e33600[i]/e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then weekly[i]=e34400[i]/e34402[i];
    /*missing value*/ if -4<e34402[i]<0 then weekly[i]=e34402[i];
    /*received compensation*/   /*with overtime*/
    if (e33500[i]>=0 and e34428[i]>0) then weekly[i]=(e33500[i]/e34428[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then weekly[i]=(e33600[i]/e34428[i]);
    if e34400[i]>=0 and e34428[i]>0 then weekly[i]=e34400[i]/e34428[i];
    /*missing value*/ if -4<e34428[i]<0 then weekly[i]=e34428[i];
    if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then weekly[i]=-3;
    /*without overtime*/
    if (e33500[i]>=0 and e34402[i]>0) then weekly[i]=(e33500[i]/e34402[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then weekly[i]=(e33600[i]/e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then weekly[i]=e34400[i]/e34402[i];
    /*missing value*/ if -4<e34402[i]<0 then weekly[i]=e34402[i];
    if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then weekly[i]=-3;
    /*if still paid hourly...*/
    if e36200[i]>=0 then weekly[i]=e36200[i];
    if e36100[i]>=0 then weekly[i]=e36100[i];
  end;

  if e83100[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0,-2) then do;
    /*no compensation*/
    if (e97300[i]>=0 and e98402[i]>0) then weekly[i]=(e97300[i]/e98402[i]);
    if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then weekly[i]=(e97500[i]/e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then weekly[i]=e98300[i]/e98402[i];
    /*missing value*/ if -4<e98402[i]<0 then weekly[i]=e98402[i];
    /*received compensation*/    /*with overtime*/
    if (e97400[i]>=0 and e98429[i]>0) then weekly[i]=(e97400[i]/e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then weekly[i]=(e97500[i]/e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 then weekly[i]=e98300[i]/e98429[i];
    /*missing value*/ if -4<e98429[i]<0 then weekly[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then weekly[i]=-3;
    /*without overtime*/
    if (e97400[i]>=0 and e98402[i]>0) then weekly[i]=(e97400[i]/e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then weekly[i]=(e97500[i]/e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then weekly[i]=e98300[i]/e98402[i];
    /*missing value*/ if -4<e98402[i]<0 then weekly[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then weekly[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then weekly[i]=e100100[i];
    if e100000[i]>=0 then weekly[i]=e100000[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in biweekly units.*/
biwkly01=-4;    biwkly02=-4;    biwkly03=-4;    biwkly04=-4;    biwkly05=-4;    biwkly06=-4;
biwkly07=-4;    biwkly08=-4;    biwkly09=-4;

array biwkly biwkly01 biwkly02 biwkly03 biwkly04 biwkly05 biwkly06 biwkly07 biwkly08 biwkly09;

do i=1 to 9;  /* biwkly start wage divided by 2x the number of hours worked per week*/
  if e19200[i]=4 then do;
    /*no compensation*/
    if (e33400[i]>=0 and e34402[i]>0) then biwkly[i]=e33400[i]/(2*e34402[i]);
    if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then biwkly[i]=e33600[i]/(2*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then biwkly[i]=e34400[i]/(2*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then biwkly[i]=e34402[i];
    /*received compensation*/    /*with overtime*/
    if (e33500[i]>=0 and e34428[i]>0) then biwkly[i]=e33500[i]/(2*e34428[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then biwkly[i]=e33600[i]/(2*e34428[i]);
    if e34400[i]>=0 and e34428[i]>0 then biwkly[i]=e34400[i]/(2*e34428[i]);
    /*missing value*/ if -4<e34428[i]<0 then biwkly[i]=e34428[i];
    if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then biwkly[i]=-3;
    /*without overtime*/
    if (e33500[i]>=0 and e34402[i]>0) then biwkly[i]=e33500[i]/(2*e34402[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then biwkly[i]=e33600[i]/(2*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then biwkly[i]=e34400[i]/(2*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then biwkly[i]=e34402[i];
    if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then biwkly[i]=-3;
    /*if still paid hourly...*/
    if e36200[i]>=0 then biwkly[i]=e36200[i];
    if e36100[i]>=0 then biwkly[i]=e36100[i];
  end;

  if e83100[i]=4 then do;
    /*no compensation*/
    if (e97300[i]>=0 and e98402[i]>0) then biwkly[i]=e97300[i]/(2*e98402[i]);
    if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then biwkly[i]=e97500[i]/(2*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then biwkly[i]=e98300[i]/(2*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then biwkly[i]=e98402[i];
    /*received compensation*/    /*with overtime*/
    if (e97400[i]>=0 and e98429[i]>0) then biwkly[i]=e97400[i]/(2*e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then biwkly[i]=e97500[i]/(2*e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 then biwkly[i]=e98300[i]/(2*e98429[i]);
    /*missing value*/ if -4<e98429[i]<0 then biwkly[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then biwkly[i]=-3;
    /*without overtime*/
    if (e97400[i]>=0 and e98402[i]>0) then biwkly[i]=e97400[i]/(2*e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then biwkly[i]=e97500[i]/(2*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then biwkly[i]=e98300[i]/(2*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then biwkly[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then biwkly[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then biwkly[i]=e100100[i];
    if e100000[i]>=0 then biwkly[i]=e100000[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in monthly units.*/
month01=-4;     month02=-4;     month03=-4;     month04=-4;     month05=-4;     month06=-4;
month07=-4;     month08=-4;     month09=-4;

array month month01 month02 month03 month04 month05 month06 month07 month08 month09;

do i=1 to 9; /* month start wage divided by 4.3x the number of hours worked per week*/
  if e19200[i]=5 then do;
    /*no compensation*/
    if (e33400[i]>=0 and e34402[i]>0) then month[i]=e33400[i]/(4.3*e34402[i]);
    if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then month[i]=e33600[i]/(4.3*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then month[i]=e34400[i]/(4.3*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then month[i]=e34402[i];
    /*received compensation*/    /*with overtime*/
    if (e33500[i]>=0 and e34428[i]>0) then month[i]=e33500[i]/(4.3*e34428[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then month[i]=e33600[i]/(4.3*e34428[i]);
    if e34400[i]>=0 and e34428[i]>0 then month[i]=e34400[i]/(4.3*e34428[i]);
    /*missing value*/ if -4<e34428[i]<0 then month[i]=e34428[i];
    if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then month[i]=-3;
    /*without overtime*/
    if (e33500[i]>=0 and e34402[i]>0) then month[i]=e33500[i]/(4.3*e34402[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then month[i]=e33600[i]/(4.3*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then month[i]=e34400[i]/(4.3*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then month[i]=e34402[i];
    if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then month[i]=-3;
    /*if still paid hourly...*/
    if e36200[i]>=0 then month[i]=e36200[i];
    if e36100[i]>=0 then month[i]=e36100[i];
  end;

  if e83100[i]=5 then do;
    /*no compensation*/
    if (e97300[i]>=0 and e98402[i]>0) then month[i]=e97300[i]/(4.3*e98402[i]);
    if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then month[i]=e97500[i]/(4.3*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then month[i]=e98300[i]/(4.3*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then month[i]=e98402[i];
    /*received compensation*/    /*with overtime*/
    if (e97400[i]>=0 and e98429[i]>0) then month[i]=e97400[i]/(4.3*e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then month[i]=e97500[i]/(4.3*e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 then month[i]=e98300[i]/(4.3*e98429[i]);
    /*missing value*/ if -4<e98429[i]<0 then month[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then month[i]=-3;
    /*without overtime*/
    if (e97400[i]>=0 and e98402[i]>0) then month[i]=e97400[i]/(4.3*e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then month[i]=e97500[i]/(4.3*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then month[i]=e98300[i]/(4.3*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then month[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then month[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then month[i]=e100100[i];
    if e100000[i]>=0 then month[i]=e100000[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in annual units.*/
annual01=-4;     annual02=-4;     annual03=-4;     annual04=-4;     annual05=-4;     annual06=-4;
annual07=-4;     annual08=-4;     annual09=-4;

array annual annual01 annual02 annual03 annual04 annual05 annual06 annual07 annual08 annual09;

do i=1 to 9; /* annual start wage divided by (number of hours worked per week * number of weeks worked per
          year)*/
  if e19200[i]=6 then do;
    /*no compensation*/
    if (e33400[i]>=0 and e34402[i]>0 and e35600[i]>0) then annual[i]=e33400[i]/(e35600[i]*e34402[i]);
    if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e35600[i]>0) then
          annual[i]=e33600[i]/(e35600[i]*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then annual[i]=e34402[i];
    /*received compensation*/    /*with overtime*/
    if (e33500[i]>=0 and e34428[i]>0 and e35600[i]>0) then annual[i]=e33500[i]/(e35600[i]*e34428[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0 and e35600[i]>0) then
          annual[i]=e33600[i]/(e35600[i]*e34428[i]);
    if e34400[i]>=0 and e34428[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34428[i]);
    /*missing value*/ if -4<e34428[i]<0 then annual[i]=e34428[i];
    if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then annual[i]=-3;
    /*without overtime*/
    if (e33500[i]>=0 and e34402[i]>0 and e35600[i]>0) then annual[i]=e33500[i]/(e35600[i]*e34402[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e35600[i]>0) then
          annual[i]=e33600[i]/(e35600[i]*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then annual[i]=e34402[i];
    if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then annual[i]=-3;
    if -4<e35600[i]<0 then annual[i]=e35600[i];
    if e35600[i]=0 then annual[i]=-3;
    /*if still paid hourly...*/
    if e36200[i]>=0 then annual[i]=e36200[i];
    if e36100[i]>=0 then annual[i]=e36100[i];
  end;

  if e83100[i]=6 then do;
    /*no compensation*/
    if (e97300[i]>=0 and e98402[i]>0 and e99500[i]>0) then annual[i]=e97300[i]/(e99500[i]*e98402[i]);
    if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e99500[i]>0) then
        annual[i]=e97500[i]/(e99500[i]*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then annual[i]=e98402[i];
    /*received compensation*/    /*with overtime*/
    if (e97400[i]>=0 and e98429[i]>0 and e99500[i]>0) then annual[i]=e97400[i]/(e99500[i]*e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0 and e99500[i]>0) then
        annual[i]=e97500[i]/(e99500[i]*e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98429[i]);
    /*missing value*/ if -4<e99500[i]<0 then annual[i]=e99500[i];
    if -4<e98429[i]<0 then annual[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then annual[i]=-3;
    /* without overtime*/
    if (e97400[i]>=0 and e98402[i]>0 and e99500[i]>0) then annual[i]=e97400[i]/(e99500[i]*e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e99500[i]>0) then
        annual[i]=e97500[i]/(e99500[i]*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98402[i]);
    /*missing value*/ if -4<e99500[i]<0 then annual[i]=e99500[i];
    if -4<e98402[i]<0 then annual[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then annual[i]=-3;
    if -4<e99500[i]<0 then annual[i]=e99500[i];
    if e99500[i]=0 then annual[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then annual[i]=e100100[i];
    if e100000[i]>=0 then annual[i]=e100000[i];
  end;
end;

/*set up the hourly wage for youths who report their wage in semimonthly units.*/
semim01=-4;     semim02=-4;     semim03=-4;     semim04=-4;     semim05=-4;     semim06=-4;
semim07=-4;     semim08=-4;     semim09=-4;

array semim semim01 semim02 semim03 semim04 semim05 semim06 semim07 semim08 semim09;

do i=1 to 9;  /* semim start wage divided by 2.15x the number of hours worked per week*/
  if e19200[i]=8 then do;
    /*no compensation*/
    if (e33400[i]>=0 and e34402[i]>0) then semim[i]=e33400[i]/(2.15*e34402[i]);
    if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then semim[i]=e33600[i]/(2.15*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then semim[i]=e34400[i]/(2.15*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then semim[i]=e34402[i];
    /*received compensation*/    /*with overtime*/
    if (e33500[i]>=0 and e34428[i]>0) then semim[i]=e33500[i]/(2.15*e34428[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then semim[i]=e33600[i]/(2.15*e34428[i]);
    if e34400[i]>=0 and e34428[i]>0 then semim[i]=e34400[i]/(2.15*e34428[i]);
    /*missing value*/ if -4<e34428[i]<0 then semim[i]=e34428[i];
    if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then semim[i]=-3;
    /*without overtime*/
    if (e33500[i]>=0 and e34402[i]>0) then semim[i]=e33500[i]/(2.15*e34402[i]);
    if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then semim[i]=e33600[i]/(2.15*e34402[i]);
    if e34400[i]>=0 and e34402[i]>0 then semim[i]=e34400[i]/(2.15*e34402[i]);
    /*missing value*/ if -4<e34402[i]<0 then semim[i]=e34402[i];
    if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then semim[i]=-3;
    /*if still paid hourly...*/
    if e36200[i]>=0 then semim[i]=e36200[i];
    if e36100[i]>=0 then semim[i]=e36100[i];
  end;

if e83100[i]=8 then do;
    /*no compensation*/
    if (e97300[i]>=0 and e98402[i]>0) then semim[i]=e97300[i]/(2.15*e98402[i]);
    if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then semim[i]=e97500[i]/(2.15*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then semim[i]=e98300[i]/(2.15*e98402[i]);
    /*missing value*/  if -4<e98402[i]<0 then semim[i]=e98402[i];
    /*received compensation*/    /*with overtime*/
    if (e97400[i]>=0 and e98429[i]>0) then semim[i]=e97400[i]/(2.15*e98429[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then semim[i]=e97500[i]/(2.15*e98429[i]);
    if e98300[i]>=0 and e98429[i]>0 then semim[i]=e98300[i]/(2.15*e98429[i]);
    /*missing value*/ if -4<e98429[i]<0 then semim[i]=e98429[i];
    if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then semim[i]=-3;
    /*without overtime*/
    if (e97400[i]>=0 and e98402[i]>0) then semim[i]=e97400[i]/(2.15*e98402[i]);
    if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then semim[i]=e97500[i]/(2.15*e98402[i]);
    if e98300[i]>=0 and e98402[i]>0 then semim[i]=e98300[i]/(2.15*e98402[i]);
    /*missing value*/ if -4<e98402[i]<0 then semim[i]=e98402[i];
    if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then semim[i]=-3;
    /*if still paid hourly...*/
    if e100100[i]>=0 then semim[i]=e100100[i];
    if e100000[i]>=0 then semim[i]=e100000[i];
  end;
end;

/*for the people who report other time units. */
array otherf otherf1-otherf9;
do i=1 to 9; otherf[i]=0; end;
do i=1 to 9;
if e19200[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) or
  e83100[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then otherf[i]=otherf[i]+1;
end;

/***************************** create the hourly rate of pay based on the start wage
          *****************************/
hrwg01=-4;      hrwg02=-4;      hrwg03=-4;      hrwg04=-4;      hrwg05=-4;      hrwg06=-4;
hrwg07=-4;      hrwg08=-4;      hrwg09=-4;

array hrwg hrwg01 hrwg02 hrwg03 hrwg04 hrwg05 hrwg06 hrwg07 hrwg08 hrwg09;

/* report hourly wage to be 0 for the family business and to be -3 for the poorly reported time units.*/
do i=1 to 9; if e19200[i] in (9,14) or e83100[i] in (9,14)  then hrwg[i]=0; end;

/* report hourly rate -1 or -2 if amount is -1 or -2*/
do i=1 to 9;
  if e19200[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) then do;
    if e33400[i]=-1  or e33500[i]=-1 then hrwg[i]=-1;
    if (e33400[i]=-2 and e33600[i]=-2) or (e33500[i]=-2 and e33600[i]=-2) then hrwg[i]=-2;
    if e33400[i]=-3 or e33500[i]=-3 then hrwg[i]=-3;
  end;
  if e83100[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) then do;
    if e97300[i]=-1 or e97400[i]=-1 then hrwg[i]=-1;
    if (e97300[i]=-2 and e97500[i]=-2) or (e97400[i]=-2 and e97500[i]=-2) then hrwg[i]=-2;
    if e97300[i]=-3 or e97400[i]=-3 then hrwg[i]=-3;
  end;
```

```
end;

/* report hourly rate -3 if no hours reported*/
do i=1 to 9; if e212001[i]=1 and e34428[i]=-4 then hrwg[i]=-3; end;

do i=1 to 9;
  if annual[i] ge 0 then hrwg[i]=annual[i];    if month[i] ge 0 then hrwg[i]=month[i];
  if biwkly[i] ge 0 then hrwg[i]=biwkly[i];    if weekly[i] ge 0 then hrwg[i]=weekly[i];
  if daily[i] ge 0 then hrwg[i]=daily[i];      if hrwage[i] ge 0 then hrwg[i]=hrwage[i];
  if semim[i] ge 0 then hrwg[i]=semim[i];
  if hrwage[i] eq -1 or daily[i]=-1 or weekly[i] eq -1 or biwkly[i] eq -1 or month[i] eq -1 or annual[i] eq -1 or
          semim[i]=-1 then hrwg[i]=-1;
  if hrwage[i] eq -2 or daily[i]=-2 or weekly[i] eq -2 or biwkly[i] eq -2 or month[i] eq -2 or annual[i] eq -2 or
          semim[i]=-2 then hrwg[i]=-2;
  if hrwage[i] eq -3 or daily[i]=-3 or weekly[i] eq -3 or biwkly[i] eq -3 or month[i] eq -3 or annual[i] eq -3 or
          semim[i]=-3 then hrwg[i]=-3;
end;

/****************************** report the corrected wage if the correction is made
          ******************************/
do i=1 to 9;
  /* rate incorrect but hours correct*/
  if e226041[i]=1 and e226042[i]=0 then do;
    /* no overtime, use e34402 for hours */
    if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
    if e22609[i]=2 and e34402[i]>0 and e34402b[i]>0 and e22626[i]>=0 then
          hrwg[i]=e22626[i]*e34402b[i]/e34402[i];
    if e22609[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0,-2) and e34402[i]>0 and e22626[i]>=0
          then hrwg[i]=e22626[i]/e34402[i];
    if e22609[i]=4 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e34402[i]);
    if e22609[i]=5 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e34402[i]);
    if e22609[i]=6 and e34402[i]>0 and e35600[i]>0 and e22626[i]>=0 then
          hrwg[i]=e22626[i]/(e35600[i]*e34402[i]);
    if e22609[i]=8 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e34402[i]);
    if e22609[i] in (9,14) then hrwg[i]=0;
    if e22609[i]=2 and e34402b[i] le 0 then hrwg[i]=-3;
    if e22609[i]=2 and -4<e34402b[i]<0 then hrwg[i]=e34402b[i];
    if e22609[i]=6 and e35600[i] le 0 then hrwg[i]=-3;
    if e22609[i]=6 and -4<e35600[i]<0 then hrwg[i]=e35600[i];
    if e22609[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e22626[i]<0 then
          hrwg[i]=e22626[i];
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e34402[i]=0 then hrwg[i]=-3;
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e34402[i]<0 then
          hrwg[i]=e34402[i];
    /* overtime, use e34428 for hours*/
    if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
    if e22609[i]=2 and e34428[i]>0 and e34430[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]*e34430[i]/e34428[i];
    if e22609[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0,99) and e34428[i]>0 and
          e22626[i]>=0 then hrwg[i]=e22626[i]/e34428[i];
    if e22609[i]=4 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e34428[i]);
    if e22609[i]=5 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e34428[i]);
    if e22609[i]=6 and e34428[i]>0 and e35600[i]>0 and e22626[i]>=0 then
          hrwg[i]=e22626[i]/(e35600[i]*e34428[i]);
    if e22609[i]=8 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e34428[i]);
    if e22609[i] in (9,14) then hrwg[i]=0;
    if e22609[i]=2 and e34430[i] le 0 then hrwg[i]=-3;
    if e22609[i]=2 and -4<e34430[i]<0 then hrwg[i]=e34430[i];
    if e22609[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e22626[i]<0 then
          hrwg[i]=e22626[i];
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e34428[i]=0 then hrwg[i]=-3;
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e34428[i]<0 then
          hrwg[i]=e34428[i];
  end;
  /*rate correct but hours incorrect*/
  if e226041[i]=0 and e226042[i]=1 then do;
    if e19200[i] ne 1 and e22610[i]>0 and e34402[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e34402[i]/e22610[i];
    if e19200[i] ne 1 and e22610[i]>0 and e34428[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e34428[i]/e22610[i];
    if e19200[i] ne 1 and e22610[i]=0 then hrwg[i]=-3;
    if e19200[i] ne 1 and -4<e22610[i]<0 then hrwg[i]=e22610[i];
  end;
  /* neither rate nor hours is correct*/
  if e226041[i]=1 and e226042[i]=1 then do;
    if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
    if e22609[i]=2 and e22610[i]>0 and e34402b[i]>0 and e22626[i]>=0 then
          hrwg[i]=e22626[i]*e34402b[i]/e22610[i];
    if e22609[i]=2 and e22610[i]>0 and e34430[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]*e34430[i]/e22610[i];
    if e22609[i]=3 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/e22610[i];
    if e22609[i]=4 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e22610[i]);
    if e22609[i]=5 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e22610[i]);
    if e22609[i]=6 and e22610[i]>0 and e35600[i]>0 and e22626[i]>=0 then
          hrwg[i]=e22626[i]/(e35600[i]*e22610[i]);
    if e22609[i]=8 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e22610[i]);
    if e22609[i]=9 then hrwg[i]=0;
    if e22609[i]=2 and e34430[i] le 0 then hrwg[i]=-3;
    if e22609[i]=2 and e34402b[i] le 0 then hrwg[i]=-3;
    if e22609[i]=2 and -4<e34430[i]<0 then hrwg[i]=e34430[i];
    if e22609[i]=2 and -4<e34402b[i]<0 then hrwg[i]=e34402b[i];
    if e22609[i]=6 and e35600[i] le 0 then hrwg[i]=-3;
    if e22609[i]=6 and -4<e35600[i]<0 then hrwg[i]=e35600[i];
    if e22609[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e22626[i]<0 then
          hrwg[i]=e22626[i];
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e22610[i]=0 then hrwg[i]=-3;
    if e22609[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e22610[i]<0 then
          hrwg[i]=e22610[i];
  end;
end;

do i=1 to 9;
  /* rate incorrect but hours correct*/
  if e102251[i]=1 and e102252[i]=0 then do;
    /* no overtime, use e98402 for hours */
    if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];
    if e100230[i]=2 and e98402[i]>0 and e98402d[i]>0 and e100248[i]>=0 then
          hrwg[i]=e100248[i]*e98402d[i]/e98402[i];
    if e100230[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0) and e98402[i]>0 and e100248[i]>=0
          then hrwg[i]=e100248[i]/e98402[i];
    if e100230[i]=4 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e98402[i]);
    if e100230[i]=5 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e98402[i]);
    if e100230[i]=6 and e98402[i]>0 and e99500[i]>0 and e100248[i]>=0 then
          hrwg[i]=e100248[i]/(e99500[i]*e98402[i]);
    if e100230[i]=8 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e98402[i]);
    if e100230[i] in (9,14) then hrwg[i]=0;
```

```
   if e100230[i]=2 and e98402d[i] le 0 then hrwg[i]=-3;
   if e100230[i]=2 and -4<e98402d[i]<0 then hrwg[i]=e98402d[i];
   if e100230[i]=6 and e99500[i] le 0 then hrwg[i]=-3;
   if e100230[i]=6 and -4<e99500[i]<0 then hrwg[i]=e99500[i];
   if e100230[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e100248[i]<0 then
         hrwg[i]=e100248[i];
   if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e98402[i]=0 then hrwg[i]=-3;
   if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e98402[i]<0 then
         hrwg[i]=e98402[i];
   /* overtime, use e98429 for hours*/
   if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];
   if e100230[i]=2 and e98429[i]>0 and e98429l[i]>0 and e100248[i]>=0 then
         hrwg[i]=e100248[i]*e98429l[i]/e98429[i];
   if e100230[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0) and e98429[i]>0 and e100248[i]>=0
         then hrwg[i]=e100248[i]/e98429[i];
   if e100230[i]=4 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e98429[i]);
   if e100230[i]=5 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e98429[i]);
   if e100230[i]=6 and e98429[i]>0 and e99500[i]>0 and e100248[i]>=0 then
         hrwg[i]=e100248[i]/(e99500[i]*e98429[i]);
   if e100230[i]=8 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e98429[i]);
   if e100230[i] in (9,14) then hrwg[i]=0;
   if e100230[i]=2 and e98429l[i] le 0 then hrwg[i]=-3;
   if e100230[i]=2 and -4<e98429l[i]<0 then hrwg[i]=e98429l[i];
   if e100230[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e100248[i]<0 then
         hrwg[i]=e100248[i];
   if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e98429[i]=0 then hrwg[i]=-3;
   if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e98429[i]<0 then
         hrwg[i]=e98429[i];
end;
/*rate correct but hours incorrect*/
if e102251[i]=0 and e102252[i]=1 then do;
   if e83100[i] ne 1 and e100231[i]>0 and e98402[i]>0 and hrwg[i]>=0 then
         hrwg[i]=hrwg[i]*e98402[i]/e100231[i];
   if e83100[i] ne 1 and e100231[i]>0 and e98429[i]>0 and hrwg[i]>=0 then
         hrwg[i]=hrwg[i]*e98429[i]/e100231[i];
   if e83100[i] ne 1 and e100231[i]=0 then hrwg[i]=-3;
   if e83100[i] ne 1 and -4<e100231[i]<0 then hrwg[i]=e100231[i];
end;
/* neither rate nor hours is correct*/
if e102251[i]=1 and e102252[i]=1 then do;
   if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];
   if e100230[i]=2 and e100231[i]>0 and e98402d[i]>0 and e100248[i]>=0 then
         hrwg[i]=e100248[i]*e98402d[i]/e100231[i];
   if e100230[i]=2 and e100231[i]>0 and e98429l[i]>0 and e100248[i]>=0 then
         hrwg[i]=e100248[i]*e98429l[i]/e100231[i];
   if e100230[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0,99) and e100231[i]>0 and
         e100248[i]>=0 then hrwg[i]=e100248[i]/e100231[i];
   if e100230[i]=4 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e100231[i]);
   if e100230[i]=5 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e100231[i]);
   if e100230[i]=6 and e100231[i]>0 and e99500[i]>0 and e100248[i]>=0 then
         hrwg[i]=e100248[i]/(e99500[i]*e100231[i]);
   if e100230[i]=8 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e100231[i]);
   if e100230[i] in (9,14) then hrwg[i]=0;
   if e100230[i]=2 and e98429l[i] le 0 then hrwg[i]=-3;
   if e100230[i]=2 and e98402d[i] le 0 then hrwg[i]=-3;
   if e100230[i]=2 and -4<e98429l[i]<0 then hrwg[i]=e98429l[i];
```

```
      if e100230[i]=2 and -4<e98402d[i]<0 then hrwg[i]=e98402d[i];
      if e100230[i]=6 and e99500[i] le 0 then hrwg[i]=-3;
      if e100230[i]=6 and -4<e99500[i]<0 then hrwg[i]=e99500[i];
      if e100230[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e100248[i]<0 then
            hrwg[i]=e100248[i];
      if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e100231[i]=0 then hrwg[i]=-
            3;
      if e100230[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e100231[i]<0 then
            hrwg[i]=e100231[i];
   end;
end;

/*******************************add the start compensation******************************/
/******************************* amount paid of overtime ********************************/
ot1=-4;  ot2=-4;  ot3=-4;  ot4=-4;  ot5=-4;  ot6=-4;  ot7=-4;  ot8=-4;  ot9=-4;
otf1=0;  otf2=0;  otf3=0;  otf4=0;  otf5=0;  otf6=0;  otf7=0;  otf8=0;  otf9=0;

array ot ot1-ot9;  array otf otf1-otf9;

/* if report hourly rate of pay */
do i=1 to 9;
   if e24501[i]>0 and e24502[i]=1 and e24514[i]>=0 then ot[i]=e24514[i];
   if e24501[i]>0 and e24502[i]=2 and e24514b[i]>0 and e24514[i]>=0 then ot[i]=e24514[i]*e24514b[i]/e24501[i];
   if e24501[i]>0 and e24502[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e24514[i]>=0 then
            ot[i]=e24514[i]/e24501[i];
   if e24501[i]>0 and e24502[i]=4 and e24514[i]>=0 then ot[i]=e24514[i]/(2*e24501[i]);
   if e24501[i]>0 and e24502[i]=5 and e24514[i]>=0 then ot[i]=e24514[i]/(4.3*e24501[i]);
   if e24501[i]>0 and e24502[i]=6 and e24514[i]>=0 then ot[i]=-3; /* since the weeks per year is unknown.*/
   if e24501[i]>0 and e24502[i]=8 and e24514[i]>=0 then ot[i]=e24514[i]/(2.15*e24501[i]);
   if e24501[i]>0 and e24502[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 0) then otf[i]=otf[i]+1;
   if e24501[i]>0 and e24502[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
   if e24501[i]>0 and e24502[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
   /*missing value*/; if -4<e24501[i]<0 then ot[i]=e24501[i];
   if -4<e24514[i]<0 then ot[i]=e24514[i];
   if e24514[i]>=0 and e24501[i]=0 then ot[i]=-3;
   if e24502[i]=2 and e24514b[i]=0 then ot[i]=-3;
   if e24502[i]=2 and -4<e24514b[i]<0 then ot[i]=e24514b[i];
   if e24502[i]=0 then ot[i]=-3; /* report more then one wage rate for overtime. */
end;

/*if report payment in other units*/
do i=1 to 9;
   if e34403[i]>0 and e34404[i]=1 and e34413c[i]>=0 then ot[i]=e34413c[i];
   if e34403[i]>0 and e34404[i]=2 and e34413e[i]>0 and e34413c[i]>=0 then ot[i]=e34413c[i]*e34413e[i]/e34403[i];
   if e34403[i]>0 and e34404[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e34413c[i]>=0 then
            ot[i]=e34413c[i]/e34403[i];
   if e34403[i]>0 and e34404[i]=4  and e34413c[i]>=0 then ot[i]=e34413c[i]/(2*e34403[i]);
   if e34403[i]>0 and e34404[i]=5  and e34413c[i]>=0 then ot[i]=e34413c[i]/(4.3*e34403[i]);
   if e34403[i]>0 and e34404[i]=6  and e34413c[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
   if e34403[i]>0 and e34404[i]=8  and e34413c[i]>=0 then ot[i]=e34413c[i]/(2.15*e34403[i]);
   if e34403[i]>0 and e34404[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
   if e34403[i]>0 and e34404[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
   /*missing value*/; if -4<e34403[i]<0 then ot[i]=e34403[i];
   if -4<e34413c[i]<0 then ot[i]=e34413c[i];
   if e34413c[i]>=0 and e34403[i]=0 then ot[i]=-3;
   if e34404[i]=2 and e34413e[i]=0 then ot[i]=-3;
   if e34404[i]=2 and -4<e34413e[i]<0 then ot[i]=e34413e[i];
   if e34404[i]=0 then ot[i]=-3;
end;

/*if report overtime pay later*/
do i=1 to 9;
   if e88501[i]>0 and e88502[i]=1  and e88512[i]>=0 then ot[i]=e88512[i];
   if e88501[i]>0 and e88502[i]=2  and e88512b[i]>0 and e88512[i]>=0 then ot[i]=e88512[i]*e88512b[i]/e88501[i];
   if e88501[i]>0 and e88502[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and e88512[i]>=0 then
            ot[i]=e88512[i]/e88501[i];
   if e88501[i]>0 and e88502[i]=4  and e88512[i]>=0 then ot[i]=e88512[i]/(2*e88501[i]);
   if e88501[i]>0 and e88502[i]=5  and e88512[i]>=0 then ot[i]=e88512[i]/(4.3*e88501[i]);
   if e88501[i]>0 and e88502[i]=6  and e88512[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
   if e88501[i]>0 and e88502[i]=8  and e88512[i]>=0 then ot[i]=e88512[i]/(2.15*e88501[i]);
   if e88501[i]>0 and e88502[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
   if e88501[i]>0 and e88502[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
   if e88502[i] in (9,14) then ot[i]=0;
   if e88502[i]=0 then ot[i]=-3;
   if e88501[i]>0 and e88502[i] in (0,7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28,-2)  then otf[i]=otf[i]+1;
   /*missing value*/; if -4<e88501[i]<0 then ot[i]=e88501[i];
   if -4<e88512[i]<0 then ot[i]=e88512[i];
   if e88512[i]>=0 and e88501[i]=0 then ot[i]=-3;
end;

do i=1 to 9;
   if e98403[i]>0 and e98404a[i]=1  and e98414[i]>=0 then ot[i]=e98414[i];
   if e98403[i]>0 and e98404a[i]=2  and e98414b[i]>0 and e98414[i]>=0 then ot[i]=e98414[i]*e98414b[i]/e98403[i];
   if e98403[i]>0 and e98404a[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and e98414[i]>=0 then
            ot[i]=e98414[i]/e98403[i];
   if e98403[i]>0 and e98404a[i]=4  and e98414[i]>=0 then ot[i]=e98414[i]/(2*e98403[i]);
   if e98403[i]>0 and e98404a[i]=5  and e98414[i]>=0 then ot[i]=e98414[i]/(4.3*e98403[i]);
   if e98403[i]>0 and e98404a[i]=6  and e98414[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
   if e98403[i]>0 and e98404a[i]=8  and e98414[i]>=0 then ot[i]=e98414[i]/(2.15*e98403[i]);
   if e98403[i]>0 and e98404a[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
   if e98403[i]>0 and e98404a[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
   if e98404a[i] in (9,14) then ot[i]=0;
   if e98404a[i]=0 then ot[i]=-3;
   if e98403[i]>0 and e98404a[i] in (0,7,9,12,13,14,15,16,17,21,22,23,24,25,26,28,-2)  then otf[i]=otf[i]+1;
   /*missing value*/;
   if -4<e98403[i]<0 then ot[i]=e98403[i];
   if -4<e98414[i]<0 then ot[i]=e98414[i];
   if e98414[i]>=0 and e98403[i]=0 then ot[i]=-3;
end;

/**** report the corrected overtime payment if the correction is made *******/
do i=1 to 9;
   /*rate incorrect but hours correct*/
   if e226044[i]=1 and e226043[i]=0 then do;
      /* report hourly payrate*/ if e19200[i]=1 then do;
         if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];
         if e22612[i]=2 and e22627[i]>=0 and e24514b[i]>0 and e24501[i]>0 then
            ot[i]=e22627[i]*e24514b[i]/e24501[i];
         if e22612[i] in (3,7) and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/e24501[i];
         if e22612[i]=4 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(2*e24501[i]);
         if e22612[i]=5 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(4.3*e24501[i]);
         if e22612[i]=6 and e22627[i]>=0 and e24501[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
      if e22612[i]=8 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(2.15*e24501[i]);
      if e22612[i]=9 then ot[i]=0;
      if e24501[i]=0 then ot[i]=-3;
      if -4<e24501[i]<0 then ot[i]=e24501[i];
      if -4<e22627[i]<0 then ot[i]=e22627[i];
      if e22612[i]=2 and e24514b[i] le 0 then ot[i]=-3;
      if e22612[i]=2 and -4<e24514b[i]<0 then ot[i]=e24514b[i];
    end;
    /*report non-hourly payrate*/ if e19200[i] ne 1 then do;
      if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];
      if e22612[i]=2 and e22627[i]>=0 and e34413e[i]>0 and e34403[i]>0 then
           ot[i]=e22627[i]*e34413e[i]/e34403[i];
      if e22612[i] in (3,7) and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/e34403[i];
      if e22612[i]=4 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(2*e34403[i]);
      if e22612[i]=5 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(4.3*e34403[i]);
      if e22612[i]=6 and e22627[i]>=0 and e34403[i]>0 then ot[i]=-3; /* since there is no weeks per year.*/
      if e22612[i]=8 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(2.15*e34403[i]);
      if e22612[i]=9 then ot[i]=0;
      if e34403[i]=0 then ot[i]=-3;
      if -4<e34403[i]<0 then ot[i]=e34403[i];
      if -4<e22627[i]<0 then ot[i]=e22627[i];
      if e22612[i]=2 and e34413e[i] le 0 then ot[i]=-3;
      if e22612[i]=2 and -4<e34413e[i]<0 then ot[i]=e34413e[i];
    end;
  end;
  /*rate correct but hours incorrect*/
  if e226044[i]=0 and e226043[i]=1 then do;
    if e24502[i] ne 1 and ot[i]>=0 and e24501[i]>0 and e22611[i]>0 then ot[i]=ot[i]*e24501[i]/e22611[i];
    if e34404[i] ne 1 and ot[i]>=0 and e34403[i]>0 and e22611[i]>0 then ot[i]=ot[i]*e34403[i]/e22611[i];
    if e24502[i] ne 1 and e34404[i] ne 1 and e22611[i]=0 then ot[i]=-3;
    if e24502[i] ne 1 and e34404[i] ne 1 and -4<e22611[i]<0 then ot[i]=e22611[i];
  end;
  /* neither rate nor the hours is correct*/
  if e226044[i]=1 and e226043[i]=1 then do;
    if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];
    if e22612[i]=2 and e22627[i]>=0 and e24514b[i]>0 and e22611[i]>0 then ot[i]=e22627[i]*e24514b[i]/e22611[i];
    if e22612[i]=2 and e22627[i]>=0 and e34413e[i]>0 and e22611[i]>0 then ot[i]=e22627[i]*e34413e[i]/e22611[i];
    if e22612[i] in (3,7) and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/e22611[i];
    if e22612[i]=4 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(2*e22611[i]);
    if e22612[i]=5 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(4.3*e22611[i]);
    if e22612[i]=6 and e22627[i]>=0 and e22611[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
    if e22612[i]=8 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(2.15*e22611[i]);
    if e22612[i]=9 then ot[i]=0;
    if e22612[i]=2 and e24514b[i] le 0 then ot[i]=-3;
    if e22612[i]=2 and e34413e[i] le 0 then ot[i]=-3;
    if e22612[i]=2 and -4<e24514b[i]<0 then ot[i]=e24514b[i];
    if e22612[i]=2 and -4<e34413e[i]<0 then ot[i]=e34413e[i];
    if e22612[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e22611[i]=0 then ot[i]=-3;
    if e22612[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e22611[i]<0 then
          ot[i]=e22611[i];
    if e22612[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e22627[i]<0 then
          ot[i]=e22627[i];
  end;
end;

/**** correction made in the later part ****/
```

```
do i=1 to 9;
  /*rate incorrect but hours correct*/
  if e102254[i]=1 and e102253[i]=0 then do;
    /* report hourly payrate*/ if e83100[i]=1 then do;
      if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
      if e100233[i]=2 and e100249[i]>=0 and e88512b[i]>0 and e88501[i]>0 then
          ot[i]=e100249[i]*e88512b[i]/e88501[i];
      if e100233[i] in (3,7) and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/e88501[i];
      if e100233[i]=4 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(2*e88501[i]);
      if e100233[i]=5 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(4.3*e88501[i]);
      if e100233[i]=6 and e100249[i]>=0 and e88501[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
      if e100233[i]=8 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(2.15*e88501[i]);
      if e100233[i]=9 then ot[i]=0;
      if e88501[i]=0 then ot[i]=-3;
      if -4<e88501[i]<0 then ot[i]=e88501[i];
      if -4<e100249[i]<0 then ot[i]=e100249[i];
      if e100233[i]=2 and e88512b[i] le 0 then ot[i]=-3;
      if e100233[i]=2 and -4<e88512b[i]<0 then ot[i]=e88512b[i];
    end;
    /*report non-hourly payrate*/ if e83100[i] ne 1 then do;
      if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
      if e100233[i]=2 and e100249[i]>=0 and e98414b[i]>0 and e98403[i]>0 then
          ot[i]=e100249[i]*e98414b[i]/e98403[i];
      if e100233[i] in (3,7) and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/e98403[i];
      if e100233[i]=4 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(2*e98403[i]);
      if e100233[i]=5 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(4.3*e98403[i]);
      if e100233[i]=6 and e100249[i]>=0 and e98403[i]>0 then ot[i]=-3; /* since there is no weeks per year.*/
      if e100233[i]=8 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(2.15*e98403[i]);
      if e100233[i]=9 then ot[i]=0;
      if e98403[i]=0 then ot[i]=-3;
      if -4<e98403[i]<0 then ot[i]=e98403[i];
      if -4<e100249[i]<0 then ot[i]=e100249[i];
      if e100233[i]=2 and e98414b[i] le 0 then ot[i]=-3;
      if e100233[i]=2 and -4<e98414b[i]<0 then ot[i]=e98414b[i];
    end;
  end;
  /*rate correct but hours incorrect*/
  if e102254[i]=0 and e102253[i]=1 then do;
    if e88502[i] ne 1 and ot[i]>=0 and e88501[i]>0 and e100232[i]>0 then ot[i]=ot[i]*e88501[i]/e100232[i];
    if e98404a[i] ne 1 and ot[i]>=0 and e98403[i]>0 and e100232[i]>0 then ot[i]=ot[i]*e98403[i]/e100232[i];
    if e88502[i] ne 1 and e98404a[i] ne 1 and e100232[i]=0 then ot[i]=-3;
    if e88502[i] ne 1 and e98404a[i] ne 1 and -4<e100232[i]<0 then ot[i]=e100232[i];
  end;
  /* neither rate nor the hours is correct*/
  if e102254[i]=1 and e102253[i]=1 then do;
    if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
    if e100233[i]=2 and e100249[i]>=0 and e88512b[i]>0 and e100232[i]>0 then
          ot[i]=e100249[i]*e88512b[i]/e100232[i];
    if e100233[i]=2 and e100249[i]>=0 and e98414b[i]>0 and e100232[i]>0 then
          ot[i]=e100249[i]*e98414b[i]/e100232[i];
    if e100233[i] in (3,7) and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/e100232[i];
    if e100233[i]=4 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(2*e100232[i]);
    if e100233[i]=5 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(4.3*e100232[i]);
    if e100233[i]=6 and e100249[i]>=0 and e100232[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
    if e100233[i]=8 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(2.15*e100232[i]);
    if e100233[i]=9 then ot[i]=0;
    if e100233[i]=2 and e88512b[i] le 0 then ot[i]=-3;
    if e100233[i]=2 and e98414b[i] le 0 then ot[i]=-3;
    if e100233[i]=2 and -4<e88512b[i]<0 then ot[i]=e88512b[i];
    if e100233[i]=2 and -4<e98414b[i]<0 then ot[i]=e98414b[i];
    if e100233[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and e100232[i]=0 then ot[i]=-3;
    if e100233[i] in (2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-2) and -4<e100232[i]<0 then
          ot[i]=e100232[i];
    if e100233[i] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,18,19,21,22,23,24,25,26,28,-1,-2) and -4<e100249[i]<0 then
          ot[i]=e100249[i];
  end;
end;

/* amount paid on tips, commissions, bonuses, incentive pay or other*/
array othpay1 othpay11 othpay21 othpay31 othpay41 othpay51 othpay61 othpay71 othpay81 othpay91;
array othpay2 othpay12 othpay22 othpay32 othpay42 othpay52 othpay62 othpay72 othpay82 othpay92;
array othpay3 othpay13 othpay23 othpay33 othpay43 othpay53 othpay63 othpay73 othpay83 othpay93;
array othpay4 othpay14 othpay24 othpay34 othpay44 othpay54 othpay64 othpay74 othpay84 othpay94;
array othpay5 othpay15 othpay25 othpay35 othpay45 othpay55 othpay65 othpay75 othpay85 othpay95;

array othpf1 othpf11 othpf21 othpf31 othpf41 othpf51 othpf61 othpf71 othpf81 othpf91;
array othpf2 othpf12 othpf22 othpf32 othpf42 othpf52 othpf62 othpf72 othpf82 othpf92;
array othpf3 othpf13 othpf23 othpf33 othpf43 othpf53 othpf63 othpf73 othpf83 othpf93;
array othpf4 othpf14 othpf24 othpf34 othpf44 othpf54 othpf64 othpf74 othpf84 othpf94;
array othpf5 othpf15 othpf25 othpf35 othpf45 othpf55 othpf65 othpf75 othpf85 othpf95;

do i=1 to 9;
  othpay1[i]=-4;  othpay2[i]=-4;     othpay3[i]=-4;     othpay4[i]=-4;     othpay5[i]=-4;
  othpf1[i]=0;    othpf2[i]=0;       othpf3[i]=0;       othpf4[i]=0;       othpf5[i]=0;
end;

/* do it for non-overtime payment*/
/********* for tips ***********/   /*report hourly wage at the beginning*/
do i=1 to 9;
  if e23901[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
  if e23901[i]>0 and e216001[i]=2 and e34430[i]>0 and e225001[i]>=0 then
          othpay1[i]=(e225001[i]*e34430[i])/e23901[i];
  if e23901[i]>0 and e216001[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225001[i]>=0
          then othpay1[i]=e225001[i]/e23901[i];
  if e23901[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e23901[i]);
  if e23901[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then othpay1[i]=e225001[i]/(e35600[i]
          *e23901[i]);
  if e23901[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e23901[i]);
  /*missing value*/; if e225001[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
  if e225001[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
end;
/*non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e34428[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
  if e34428[i]>0 and e216001[i]=2 and e34430[i]>0 and e225001[i]>=0 then
          othpay1[i]=(e225001[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e216001[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225001[i]>=0
          then othpay1[i]=e225001[i]/e34428[i];
  if e34428[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e34428[i]);
  if e34428[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then othpay1[i]=e225001[i]/(e35600[i]
         *e34428[i]);
  if e34428[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e34428[i]);
  /*missing value*/; if e225001[i]>=0 and -4<e34428[i]<0 then othpay1[i]=e34428[i];
  if -4<e225001[i]<0 then othpay1[i]=e225001[i];
  if e225001[i]>=0 and e34428[i]=0 then othpay1[i]=-3;
end;
/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e34402[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
  if e34402[i]>0 and e216001[i]=2 and e34402b[i]>0 and e225001[i]>=0 then
         othpay1[i]=(e225001[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e216001[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225001[i]>=0
         then othpay1[i]=e225001[i]/e34402[i];
  if e34402[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e34402[i]);
  if e34402[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then
         othpay1[i]=e225001[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e34402[i]);
  if e216001[i] in (9,14) then othpay1[i]=0;
  if e216001[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
  /*missing value*/; if e225001[i]>=0 and -4<e34402[i]<0 then othpay1[i]=e34402[i];
  if e225001[i]>=0 and e34402[i]=0 then othpay1[i]=-3;
  if e216001[i]=2 and e34402b[i] le 0 and e34430[i] le 0  then othpay1[i]=-3;
  if e216001[i]=2 and -4<e34402b[i]<0 then othpay1[i]=e34402b[i];
  if e216001[i]=2 and -4<e34430[i]<0 then othpay1[i]=e34430[i];
  if e216001[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
  if e216001[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
end;

/* if tips is corrected in the later part */
do i=1 to 9;
  if e226045[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e22613[i]=1 and e22628[i]>=0 then othpay1[i]=e22628[i];
    if e22613[i]=2 and e22628[i]>=0 and e34430[i]>0 and e23901[i]>0 then
         othpay1[i]=e22628[i]*e34430[i]/e23901[i];
    if e22613[i] in (3,7) and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/e23901[i];
    if e22613[i]=4 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(2*e23901[i]);
    if e22613[i]=5 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(4.3*e23901[i]);
    if e22613[i]=6 and e22628[i]>=0 and e23901[i]>0 and e35600[i]>0 then
         othpay1[i]=e22628[i]/(e35600[i]*e23901[i]);
    if e22613[i]=8 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(2.15*e23901[i]);
    /*missing value*/ if -4<e23901[i]<0 then othpay1[i]=e23901[i];
    if e23901[i]=0 then othpay1[i]=-3;
    /* non-hourly wage at the beginning, with overtime*/
    if e22613[i]=1 and e22628[i]>=0 then othpay1[i]=e22628[i];
    if e22613[i]=2 and e22628[i]>=0 and e34430[i]>0 and e34428[i]>0 then
         othpay1[i]=e22628[i]*e34430[i]/e34428[i];
    if e22613[i] in (3,7) and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/e34428[i];
    if e22613[i]=4 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(2*e34428[i]);
    if e22613[i]=5 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(4.3*e34428[i]);
    if e22613[i]=6 and e22628[i]>=0 and e34428[i]>0 and e35600[i]>0 then
         othpay1[i]=e22628[i]/(e35600[i]*e34428[i]);
    if e22613[i]=8 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(2.15*e34428[i]);
    /*missing value*/ if -4<e34428[i]<0 then othpay1[i]=e34428[i];
    if -4<e22628[i]<0 then othpay1[i]=e22628[i];
    if e34428[i]=0 then othpay1[i]=-3;
    /* non-hourly wage at the beginning, without overtime*/
    if e22613[i]=1 and e22628[i]>=0 then othpay1[i]=e22628[i];
    if e22613[i]=2 and e22628[i]>=0 and e34402b[i]>0 and e34402[i]>0 then
          othpay1[i]=e22628[i]*e34402b[i]/e34402[i];
    if e22613[i] in (3,7) and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/e34402[i];
    if e22613[i]=4 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(2*e34402[i]);
    if e22613[i]=5 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(4.3*e34402[i]);
    if e22613[i]=6 and e22628[i]>=0 and e34402[i]>0 and e35600[i]>0 then
          othpay1[i]=e22628[i]/(e35600[i]*e34402[i]);
    if e22613[i]=8 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(2.15*e34402[i]);
    if e22613[i]=9 then othpay1[i]=0;
    /*missing value*/ if -4<e34402[i]<0 then othpay1[i]=e34402[i];
    if e34402[i]=0 then othpay1[i]=-3;
    if e22613[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay1[i]=-3;
    if e22613[i]=2 and -4<e34402b[i]<0 then othpay1[i]=e34402b[i];
    if e22613[i]=2 and -4<e34430[i]<0 then othpay1[i]=e34430[i];
    if e22613[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
    if e22613[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
  end;
end;

/********* for commissions **********/   /*report hourly wage at the beginning*/
do i=1 to 9;
  if e23901[i]>0 and e216002[i]=1 and e225002[i]>=0 then othpay2[i]=e225002[i];
  if e23901[i]>0 and e216002[i]=2 and e34430[i]>0 and e225002[i]>=0 then
          othpay2[i]=(e225002[i]*e34430[i])/e23901[i];
  if e23901[i]>0 and e216002[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225002[i]>=0
          then othpay2[i]=e225002[i]/e23901[i];
  if e23901[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e23901[i]);
  if e23901[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
          othpay2[i]=e225002[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e23901[i]);
  *missing value*/;    if e225002[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
  if e225002[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
end;
  /* non-hourly wage at the beginning,with overtime*/
do i=1 to 9;
  if e34428[i]>0 and e216002[i]=1 and e225002[i]>=0 then othpay2[i]=e225002[i];
  if e34428[i]>0 and e216002[i]=2 and e34430[i]>0 and e225002[i]>=0 then
          othpay2[i]=(e225002[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e216002[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225002[i]>=0
          then othpay2[i]=e225002[i]/e34428[i];
  if e34428[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e34428[i]);
  if e34428[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
          othpay2[i]=e225002[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e34428[i]);
  *missing value*/;    if e225002[i]>=0 and -4<e34428[i]<0 then othpay2[i]=e34428[i];
  if -4<e225002[i]<0 then othpay2[i]=e225002[i];
  if e225002[i]>=0 and e34428[i]=0 then othpay2[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
```

```
  if e34402[i]>0 and e216002[i]=1 and e225002[i]>=0 then othpay2[i]=e225002[i];
  if e34402[i]>0 and e216002[i]=2 and e34402b[i]>0 and e225002[i]>=0 then
          othpay2[i]=(e225002[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e216002[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225002[i]>=0
          then othpay2[i]=e225002[i]/e34402[i];
  if e34402[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e34402[i]);
  if e34402[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
          othpay2[i]=e225002[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e216002[i]=18 and e225002[i]>=0 then othpay2[i]=e225002[i]/(13*e34402[i]);
  if e34402[i]>0 and e216002[i]=19 and e225002[i]>=0 then othpay2[i]=e225002[i]/(26*e34402[i]);
  if e216002[i] in (9,14) then othpay2[i]=0;
  if e216002[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf2[i]=othpf2[i]+1;
  *missing value*/;    if e225002[i]>=0 and -4<e34402[i]<0 then othpay2[i]=e34402[i];
  if e225002[i]>=0 and e34402[i]=0 then othpay2[i]=-3;
  if e216002[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay2[i]=-3;
  if e216002[i]=2 and -4<e34402b[i]<0 then othpay2[i]=e34402b[i];
  if e216002[i]=2 and -4<e34430[i]<0 then othpay2[i]=e34430[i];
  if e216002[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e216002[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
end;
  /* if commissions is corrected in the later part */
do i=1 to 9;
  if e226046[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e22614[i]=1 and e22629[i]>=0 then othpay2[i]=e22629[i];
    if e22614[i]=2 and e22629[i]>=0 and e34430[i]>0 and e23901[i]>0 then
          othpay2[i]=e22629[i]*e34430[i]/e23901[i];
    if e22614[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22629[i]>=0 and e23901[i]>0 then
          othpay2[i]=e22629[i]/e23901[i];
    if e22614[i]=4 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(2*e23901[i]);
    if e22614[i]=5 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(4.3*e23901[i]);
    if e22614[i]=6 and e22629[i]>=0 and e23901[i]>0 and e35600[i]>0 then
          othpay2[i]=e22629[i]/(e35600[i]*e23901[i]);
    if e22614[i]=8 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(2.15*e23901[i]);
    /*missing value*/   if -4<e23901[i]<0 then othpay2[i]=e23901[i];
    if e23901[i]=0 then othpay2[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
    if e22614[i]=1 and e22629[i]>=0 then othpay2[i]=e22629[i];
    if e22614[i]=2 and e22629[i]>=0 and e34430[i]>0 and e34428[i]>0 then
          othpay2[i]=e22629[i]*e34430[i]/e34428[i];
    if e22614[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22629[i]>=0 and e34428[i]>0 then
          othpay2[i]=e22629[i]/e34428[i];
    if e22614[i]=4 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(2*e34428[i]);
    if e22614[i]=5 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(4.3*e34428[i]);
    if e22614[i]=6 and e22629[i]>=0 and e34428[i]>0 and e35600[i]>0 then
          othpay2[i]=e22629[i]/(e35600[i]*e34428[i]);
    if e22614[i]=8 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(2.15*e34428[i]);
    /*missing value*/    if -4<e34428[i]<0 then othpay2[i]=e34428[i];
    if -4<e22629[i]<0 then othpay2[i]=e22629[i];
    if e34428[i]=0 then othpay2[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e22614[i]=1 and e22629[i]>=0 then othpay2[i]=e22629[i];
    if e22614[i]=2 and e22629[i]>=0 and e34402b[i]>0 and e34402[i]>0 then
          othpay2[i]=e22629[i]*e34402b[i]/e34402[i];
    if e22614[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22629[i]>=0 and e34402[i]>0 then
          othpay2[i]=e22629[i]/e34402[i];
    if e22614[i]=4 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(2*e34402[i]);
    if e22614[i]=5 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(4.3*e34402[i]);
    if e22614[i]=6 and e22629[i]>=0 and e34402[i]>0 and e35600[i]>0 then
          othpay2[i]=e22629[i]/(e35600[i]*e34402[i]);
    if e22614[i]=8 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(2.15*e34402[i]);
    if e22614[i] in (9,14) then othpay2[i]=0;
    /*missing value*/   if -4<e34402[i]<0 then othpay2[i]=e34402[i];
    if e34402[i]=0 then othpay2[i]=-3;
    if e22614[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay2[i]=-3;
    if e22614[i]=2 and -4<e34402b[i]<0 then othpay2[i]=e34402b[i];
    if e22614[i]=2 and -4<e34430[i]<0 then othpay2[i]=e34430[i];
    if e22614[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
    if e22614[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
  end;
end;

/********* for bonuses **********/   /*report hourly wage at the beginning*/
do i=1 to 9;
  if e23901[i]>0 and e216003[i]=1 and e225003[i]>=0 then othpay3[i]=e225003[i];
  if e23901[i]>0 and e216003[i]=2 and e34430[i]>0 and e225003[i]>=0 then
          othpay3[i]=(e225003[i]*e34430[i])/e23901[i];
  if e23901[i]>0 and e216003[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225003[i]>=0
          then othpay3[i]=e225003[i]/e23901[i];
  if e23901[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e23901[i]);
  if e23901[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
          othpay3[i]=e225003[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e23901[i]);
  *missing value*/;    if e225003[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
  if e225003[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
end;
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e34428[i]>0 and e216003[i]=1 and e225003[i]>=0 then othpay3[i]=e225003[i];
  if e34428[i]>0 and e216003[i]=2 and e34430[i]>0 and e225003[i]>=0 then
          othpay3[i]=(e225003[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e216003[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225003[i]>=0
          then othpay3[i]=e225003[i]/e34428[i];
  if e34428[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e34428[i]);
  if e34428[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
          othpay3[i]=e225003[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e34428[i]);
  /*missing value*/;    if e225003[i]>=0 and -4<e34428[i]<0 then othpay3[i]=e34428[i];
  if -4<e225003[i]<0 then othpay3[i]=e225003[i];
  if e225003[i]>=0 and e34428[i]=0 then othpay3[i]=-3;
end;
  /* non-hourly wage at the beginning,without overtime*/
do i=1 to 9;
  if e34402[i]>0 and e216003[i]=1 and e225003[i]>=0 then othpay3[i]=e225003[i];
  if e34402[i]>0 and e216003[i]=2 and e34402b[i]>0 and e225003[i]>=0 then
          othpay3[i]=(e225003[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e216003[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225003[i]>=0
          then othpay3[i]=e225003[i]/e34402[i];
```

```
 if e34402[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e34402[i]);
 if e34402[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e34402[i]);
 if e34402[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
        othpay3[i]=e225003[i]/(e35600[i]*e34402[i]);
 if e34402[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e34402[i]);
 if e216003[i] in (9,14) then othpay3[i]=0;
 if e216003[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf3[i]=othpf3[i]+1;
 *missing value*/;   if e225003[i]>=0 and -4<e34402[i]<0 then othpay3[i]=e34402[i];
 if e225003[i]>=0 and e34402[i]=0 then othpay3[i]=-3;
 if e216003[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay3[i]=-3;
 if e216003[i]=2 and -4<e34402b[i]<0 then othpay3[i]=e34402b[i];
 if e216003[i]=2 and -4<e34430[i]<0 then othpay3[i]=e34430[i];
 if e216003[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
 if e216003[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;
 /* if bonus is corrected in the later part */
do i=1 to 9;
 if e226047[i]=1 then do;
 /*report hourly wage at the beginning*/
   if e22615[i]=1 and e22630[i]>=0 then othpay3[i]=e22630[i];
   if e22615[i]=2 and e22630[i]>=0 and e34430[i]>0 and e23901[i]>0 then
        othpay3[i]=e22630[i]*e34430[i]/e23901[i];
   if e22615[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22630[i]>=0 and e23901[i]>0 then
        othpay3[i]=e22630[i]/e23901[i];
   if e22615[i]=4 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(2*e23901[i]);
   if e22615[i]=5 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(4.3*e23901[i]);
   if e22615[i]=6 and e22630[i]>=0 and e23901[i]>0 and e35600[i]>0 then
        othpay3[i]=e22630[i]/(e35600[i]*e23901[i]);
   if e22615[i]=8 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(2.15*e23901[i]);
   /*missing value*/  if -4<e23901[i]<0 then othpay3[i]=e23901[i];
   if e23901[i]=0 then othpay3[i]=-3;
 /* non-hourly wage at the beginning, with overtime*/
   if e22615[i]=1 and e22630[i]>=0 then othpay3[i]=e22630[i];
   if e22615[i]=2 and e22630[i]>=0 and e34430[i]>0 and e34428[i]>0 then
        othpay3[i]=e22630[i]*e34430[i]/e34428[i];
   if e22615[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22630[i]>=0 and e34428[i]>0 then
        othpay3[i]=e22630[i]/e34428[i];
   if e22615[i]=4 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(2*e34428[i]);
   if e22615[i]=5 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(4.3*e34428[i]);
   if e22615[i]=6 and e22630[i]>=0 and e34428[i]>0 and e35600[i]>0 then
        othpay3[i]=e22630[i]/(e35600[i]*e34428[i]);
   if e22615[i]=8 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(2.15*e34428[i]);
   /*missing value*/  if -4<e34428[i]<0 then othpay3[i]=e34428[i];
   if -4<e22630[i]<0 then othpay3[i]=e22630[i];
   if e34428[i]=0 then othpay3[i]=-3;
 /* non-hourly wage at the beginning, without overtime*/
   if e22615[i]=1 and e22630[i]>=0 then othpay3[i]=e22630[i];
   if e22615[i]=2 and e22630[i]>=0 and e34402b[i]>0 and e34402[i]>0 then
        othpay3[i]=e22630[i]*e34402b[i]/e34402[i];
   if e22615[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22630[i]>=0 and e34402[i]>0 then
        othpay3[i]=e22630[i]/e34402[i];
   if e22615[i]=4 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(2*e34402[i]);
   if e22615[i]=5 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(4.3*e34402[i]);
   if e22615[i]=6 and e22630[i]>=0 and e34402[i]>0 and e35600[i]>0 then
        othpay3[i]=e22630[i]/(e35600[i]*e34402[i]);
   if e22615[i]=8 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(2.15*e34402[i]);
```

```
      if e22615[i] in (9,14) then othpay3[i]=0;
      /*missing value*/   if -4<e34402[i]<0 then othpay3[i]=e34402[i];
      if e34402[i]=0 then othpay3[i]=-3;
      if e22615[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay3[i]=-3;
      if e22615[i]=2 and -4<e34402b[i]<0 then othpay3[i]=e34402b[i];
      if e22615[i]=2 and -4<e34430[i]<0 then othpay3[i]=e34430[i];
      if e22615[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
      if e22615[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
   end;
end;

/********** for incentive pay ***********/   /*report hourly wage at the beginning*/
do i=1 to 9;
   if e23901[i]>0 and e216004[i]=1 and e225004[i]>=0 then othpay4[i]=e225004[i];
   if e23901[i]>0 and e216004[i]=2 and e34430[i]>0 and e225004[i]>=0 then
            othpay4[i]=(e225004[i]*e34430[i])/e23901[i];
   if e23901[i]>0 and e216004[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225004[i]>=0
            then othpay4[i]=e225004[i]/e23901[i];
   if e23901[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e23901[i]);
   if e23901[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e23901[i]);
   if e23901[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
            othpay4[i]=e225004[i]/(e35600[i]*e23901[i]);
   if e23901[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e23901[i]);
   *missing value*/;    if e225004[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
   if e225004[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
end;
   /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
   if e34428[i]>0 and e216004[i]=1 and e225004[i]>=0 then othpay4[i]=e225004[i];
   if e34428[i]>0 and e216004[i]=2 and e34430[i]>0 and e225004[i]>=0 then
            othpay4[i]=(e225004[i]*e34430[i])/e34428[i];
   if e34428[i]>0 and e216004[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225004[i]>=0
            then othpay4[i]=e225004[i]/e34428[i];
   if e34428[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e34428[i]);
   if e34428[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e34428[i]);
   if e34428[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
            othpay4[i]=e225004[i]/(e35600[i]*e34428[i]);
   if e34428[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e34428[i]);
   *missing value*/;    if e225004[i]>=0 and -4<e34428[i]<0 then othpay4[i]=e34428[i];
   if -4<e225004[i]<0 then othpay4[i]=e225004[i];
   if e225004[i]>=0 and e34428[i]=0 then othpay4[i]=-3;
end;
   /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
   if e34402[i]>0 and e216004[i]=1 and e225004[i]>=0 then othpay4[i]=e225004[i];
   if e34402[i]>0 and e216004[i]=2 and e34402b[i]>0 and e225004[i]>=0 then
            othpay4[i]=(e225004[i]*e34402b[i])/e34402[i];
   if e34402[i]>0 and e216004[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225004[i]>=0
            then othpay4[i]=e225004[i]/e34402[i];
   if e34402[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e34402[i]);
   if e34402[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e34402[i]);
   if e34402[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
            othpay4[i]=e225004[i]/(e35600[i]*e34402[i]);
   if e34402[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e34402[i]);
   if e216004[i] in (9,14) then othpay4[i]=0;
   if e216004[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf4[i]=othpf4[i]+1;
```

```
  *missing value*/;    if e225004[i]>=0 and -4<e34402[i]<0 then othpay4[i]=e34402[i];
  if e225004[i]>=0 and e34402[i]=0 then othpay4[i]=-3;
  if e216004[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay4[i]=-3;
  if e216004[i]=2 and -4<e34402b[i]<0 then othpay4[i]=e34402b[i];
  if e216004[i]=2 and -4<e34430[i]<0 then othpay4[i]=e34430[i];
  if e216004[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
  if e216004[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
end;
  /* if incentive pay is corrected in the later part */
do i=1 to 9;
  if e226048[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e22616[i]=1 and e22631[i]>=0 then othpay4[i]=e22631[i];
    if e22616[i]=2 and e22631[i]>=0 and e34430[i]>0 and e23901[i]>0 then
         othpay4[i]=e22631[i]*e34430[i]/e23901[i];
    if e22616[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22631[i]>=0 and e23901[i]>0 then
         othpay4[i]=e22631[i]/e23901[i];
    if e22616[i]=4 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(2*e23901[i]);
    if e22616[i]=5 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(4.3*e23901[i]);
    if e22616[i]=6 and e22631[i]>=0 and e23901[i]>0 and e35600[i]>0 then
         othpay4[i]=e22631[i]/(e35600[i]*e23901[i]);
    if e22616[i]=8 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(2.15*e23901[i]);
    /*missing value*/   if -4<e23901[i]<0 then othpay4[i]=e23901[i];
    if e23901[i]=0 then othpay4[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
    if e22616[i]=1 and e22631[i]>=0 then othpay4[i]=e22631[i];
    if e22616[i]=2 and e22631[i]>=0 and e34430[i]>0 and e34428[i]>0 then
         othpay4[i]=e22631[i]*e34430[i]/e34428[i];
    if e22616[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22631[i]>=0 and e34428[i]>0 then
         othpay4[i]=e22631[i]/e34428[i];
    if e22616[i]=4 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(2*e34428[i]);
    if e22616[i]=5 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(4.3*e34428[i]);
    if e22616[i]=6 and e22631[i]>=0 and e34428[i]>0 and e35600[i]>0 then
         othpay4[i]=e22631[i]/(e35600[i]*e34428[i]);
    if e22616[i]=8 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(2.15*e34428[i]);
    /*missing value*/   if -4<e34428[i]<0 then othpay4[i]=e34428[i];
    if -4<e22631[i]<0 then othpay4[i]=e22631[i];
    if e34428[i]=0 then othpay4[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e22616[i]=1 and e22631[i]>=0 then othpay4[i]=e22631[i];
    if e22616[i]=2 and e22631[i]>=0 and e34402b[i]>0 and e34402[i]>0 then
         othpay4[i]=e22631[i]*e34402b[i]/e34402[i];
    if e22616[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22631[i]>=0 and e34402[i]>0 then
         othpay4[i]=e22631[i]/e34402[i];
    if e22616[i]=4 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(2*e34402[i]);
    if e22616[i]=5 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(4.3*e34402[i]);
    if e22616[i]=6 and e22631[i]>=0 and e34402[i]>0 and e35600[i]>0 then
         othpay4[i]=e22631[i]/(e35600[i]*e34402[i]);
    if e22616[i]=8 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(2.15*e34402[i]);
    if e22616[i] in (9,14) then othpay4[i]=0;
    /*missing value*/   if -4<e34402[i]<0 then othpay4[i]=e34402[i];
    if e34402[i]=0 then othpay4[i]=-3;
    if e22616[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay4[i]=-3;
    if e22616[i]=2 and -4<e34402b[i]<0 then othpay4[i]=e34402b[i];
    if e22616[i]=2 and -4<e34430[i]<0 then othpay4[i]=e34430[i];
    if e22616[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
```

```
      if e22616[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
   end;
end;

/********* for others *********/   /*report hourly wage at the beginning*/
do i=1 to 9;
   if e23901[i]>0 and e216005[i]=1 and e225005[i]>=0 then othpay5[i]=e225005[i];
   if e23901[i]>0 and e216005[i]=2 and e34430[i]>0 and e225005[i]>=0 then
            othpay5[i]=(e225005[i]*e34430[i])/e23901[i];
   if e23901[i]>0 and e216005[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225005[i]>=0
            then othpay5[i]=e225005[i]/e23901[i];
   if e23901[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e23901[i]);
   if e23901[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e23901[i]);
   if e23901[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
            othpay5[i]=e225005[i]/(e35600[i]*e23901[i]);
   if e23901[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e23901[i]);
   *missing value*/;    if e225005[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
   if e225005[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
end;
   /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
   if e34428[i]>0 and e216005[i]=1 and e225005[i]>=0 then othpay5[i]=e225005[i];
   if e34428[i]>0 and e216005[i]=2 and e34430[i]>0 and e225005[i]>=0 then
            othpay5[i]=(e225005[i]*e34430[i])/e34428[i];
   if e34428[i]>0 and e216005[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225005[i]>=0
            then othpay5[i]=e225005[i]/e34428[i];
   if e34428[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e34428[i]);
   if e34428[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e34428[i]);
   if e34428[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
            othpay5[i]=e225005[i]/(e35600[i]*e34428[i]);
   if e34428[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e34428[i]);
   *missing value*/;    if e225005[i]>=0 and -4<e34428[i]<0 then othpay5[i]=e34428[i];
   if -4<e225005[i]<0 then othpay5[i]=e225005[i];
   if e225005[i]>=0 and e34428[i]=0 then othpay5[i]=-3;
end;
   /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
   if e34402[i]>0 and e216005[i]=1 and e225005[i]>=0 then othpay5[i]=e225005[i];
   if e34402[i]>0 and e216005[i]=2 and e34402b[i]>0 and e225005[i]>=0 then
            othpay5[i]=(e225005[i]*e34402b[i])/e34402[i];
   if e34402[i]>0 and e216005[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e225005[i]>=0
            then othpay5[i]=e225005[i]/e34402[i];
   if e34402[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e34402[i]);
   if e34402[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e34402[i]);
   if e34402[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
            othpay5[i]=e225005[i]/(e35600[i]*e34402[i]);
   if e34402[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e34402[i]);
   if e216005[i] in (9,14) then othpay5[i]=0;
   if e216005[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf5[i]=othpf5[i]+1;
   *missing value*/;    if e225005[i]>=0 and -4<e34402[i]<0 then othpay5[i]=e34402[i];
   if e225005[i]>=0 and e34402[i]=0 then othpay5[i]=-3;
   if e216005[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay5[i]=-3;
   if e216005[i]=2 and -4<e34402b[i]<0 then othpay5[i]=e34402b[i];
   if e216005[i]=2 and -4<e34430[i]<0 then othpay5[i]=e34430[i];
   if e216005[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
   if e216005[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
```

```
end;
  /* if other compensation is corrected in the later part */
do i=1 to 9;
  if e226049[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e22617[i]=1 and e22632[i]>=0 then othpay5[i]=e22632[i];
    if e22617[i]=2 and e22632[i]>=0 and e34430[i]>0 and e23901[i]>0 then
         othpay5[i]=e22632[i]*e34430[i]/e23901[i];
    if e22617[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22632[i]>=0 and e23901[i]>0 then
         othpay5[i]=e22632[i]/e23901[i];
    if e22617[i]=4 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(2*e23901[i]);
    if e22617[i]=5 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(4.3*e23901[i]);
    if e22617[i]=6 and e22632[i]>=0 and e23901[i]>0 and e35600[i]>0 then
         othpay5[i]=e22632[i]/(e35600[i]*e23901[i]);
    if e22617[i]=8 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(2.15*e23901[i]);
    /*missing value*/  if -4<e23901[i]<0 then othpay5[i]=e23901[i];
    if e23901[i]=0 then othpay5[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
    if e22617[i]=1 and e22632[i]>=0 then othpay5[i]=e22632[i];
    if e22617[i]=2 and e22632[i]>=0 and e34430[i]>0 and e34428[i]>0 then
         othpay5[i]=e22632[i]*e34430[i]/e34428[i];
    if e22617[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22632[i]>=0 and e34428[i]>0 then
         othpay5[i]=e22632[i]/e34428[i];
    if e22617[i]=4 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(2*e34428[i]);
    if e22617[i]=5 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(4.3*e34428[i]);
    if e22617[i]=6 and e22632[i]>=0 and e34428[i]>0 and e35600[i]>0 then
         othpay5[i]=e22632[i]/(e35600[i]*e34428[i]);
    if e22617[i]=8 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(2.15*e34428[i]);
    /*missing value*/  if -4<e34428[i]<0 then othpay5[i]=e34428[i];
    if -4<e22632[i]<0 then othpay5[i]=e22632[i];
    if e34428[i]=0 then othpay5[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e22617[i]=1 and e22632[i]>=0 then othpay5[i]=e22632[i];
    if e22617[i]=2 and e22632[i]>=0 and e34402b[i]>0 and e34402[i]>0 then
         othpay5[i]=e22632[i]*e34402b[i]/e34402[i];
    if e22617[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e22632[i]>=0 and e34402[i]>0 then
         othpay5[i]=e22632[i]/e34402[i];
    if e22617[i]=4 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(2*e34402[i]);
    if e22617[i]=5 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(4.3*e34402[i]);
    if e22617[i]=6 and e22632[i]>=0 and e34402[i]>0 and e35600[i]>0 then
         othpay5[i]=e22632[i]/(e35600[i]*e34402[i]);
    if e22617[i]=8 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(2.15*e34402[i]);
    if e22617[i] in (9,14) then othpay5[i]=0;
    /*missing value*/   if -4<e34402[i]<0 then othpay5[i]=e34402[i];
    if e34402[i]=0 then othpay5[i]=-3;
    if e22617[i]=2 and e34402b[i] le 0 and e34430[i] le 0 then othpay5[i]=-3;
    if e22617[i]=2 and -4<e34402b[i]<0 then othpay5[i]=e34402b[i];
    if e22617[i]=2 and -4<e34430[i]<0 then othpay5[i]=e34430[i];
    if e22617[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
    if e22617[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
  end;
end;

/******** do it for non-overtime payment if answered in the later part *********/
/******** for tips ********/   /*report hourly wage at the beginning*/
do i=1 to 9;
```

```
  if e88000[i]>0 and e102051[i]=1 and e102141[i]>=0 then othpay1[i]=e102141[i];
  if e88000[i]>0 and e102051[i]=2 and e98429l[i]>0 and e102141[i]>=0 then
          othpay1[i]=(e102141[i]*e98429l[i])/e88000[i];
  if e88000[i]>0 and e102051[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102141[i]>=0
          then othpay1[i]=e102141[i]/e88000[i];
  if e88000[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e88000[i]);
  if e88000[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e88000[i]);
  if e88000[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then
          othpay1[i]=e102141[i]/(e99500[i]*e88000[i]);
  if e88000[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e88000[i]);
  /*missing value*/;    if e102141[i]>=0 and -4<e88000[i]<0 then othpay1[i]=e88000[i];
  if e102141[i]>=0 and e88000[i]=0 then othpay1[i]=-3;
end;
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e98429[i]>0 and e102051[i]=1 and e102141[i]>=0 then othpay1[i]=e102141[i];
  if e98429[i]>0 and e102051[i]=2 and e98429l[i]>0 and e102141[i]>=0 then
          othpay1[i]=(e102141[i]*e98429l[i])/e98429[i];
  if e98429[i]>0 and e102051[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102141[i]>=0
          then othpay1[i]=e102141[i]/e98429[i];
  if e98429[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e98429[i]);
  if e98429[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e98429[i]);
  if e98429[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then
          othpay1[i]=e102141[i]/(e99500[i]*e98429[i]);
  if e98429[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e98429[i]);
  /*missing value*/;    if e102141[i]>=0 and -4<e98429[i]<0 then othpay1[i]=e98429[i];
  if -4<e102141[i]<0 then othpay1[i]=e102141[i];
  if e102141[i]>=0 and e98429[i]=0 then othpay1[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e98402[i]>0 and e102051[i]=1 and e102141[i]>=0 then othpay1[i]=e102141[i];
  if e98402[i]>0 and e102051[i]=2 and e98402d[i]>0 and e102141[i]>=0 then
          othpay1[i]=(e102141[i]*e98402d[i])/e98402[i];
  if e98402[i]>0 and e102051[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102141[i]>=0
          then othpay1[i]=e102141[i]/e98402[i];
  if e98402[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e98402[i]);
  if e98402[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e98402[i]);
  if e98402[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then othpay1[i]=e102141[i]/(52*e98402[i]);
  if e98402[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e98402[i]);
  if e102051[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
  if e102051[i] in (9,14) then othpay1[i]=0;
  /*missing value*/;    if e102141[i]>=0 and -4<e98402[i]<0 then othpay1[i]=e98402[i];
  if e102141[i]>=0 and e98402[i]=0 then othpay1[i]=-3;
  if e102051[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay1[i]=-3;
  if e102051[i]=2 and -4<e98402d[i]<0 then othpay1[i]=e98402d[i];
  if e102051[i]=2 and -4<e98429l[i]<0 then othpay1[i]=e98429l[i];
  if e102051[i]=6 and e99500[i] le 0 then othpay1[i]=-3;
  if e102051[i]=6 and -4<e99500[i]<0 then othpay1[i]=e99500[i];
end;
/* if tips is corrected in the later part */
do i=1 to 9;
  if e102255[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e100234[i]=1 and e100250[i]>=0 then othpay1[i]=e100250[i];
```

```
      if e100234[i]=2 and e100250[i]>=0 and e98429l[i]>0 and e88000[i]>0 then
            othpay1[i]=e100250[i]*e98429l[i]/e88000[i];
      if e100234[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100250[i]>=0 and e88000[i]>0
            then othpay1[i]=e100250[i]/e88000[i];
      if e100234[i]=4 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(2*e88000[i]);
      if e100234[i]=5 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(4.3*e88000[i]);
      if e100234[i]=6 and e100250[i]>=0 and e88000[i]>0 and e99500[i]>0 then
            othpay1[i]=e100250[i]/(e99500[i]*e88000[i]);
      if e100234[i]=8 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(2.15*e88000[i]);
      /*missing value*/   if -4<e88000[i]<0 then othpay1[i]=e88000[i];
      if e88000[i]=0 then othpay1[i]=-3;
   /* non-hourly wage at the beginning, with overtime*/
      if e100234[i]=1 and e100250[i]>=0 then othpay1[i]=e100250[i];
      if e100234[i]=2 and e100250[i]>=0 and e98429l[i]>0 and e98429[i]>0 then
            othpay1[i]=e100250[i]*e98429l[i]/e98429[i];
      if e100234[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100250[i]>=0 and e98429[i]>0
            then othpay1[i]=e100250[i]/e98429[i];
      if e100234[i]=4 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(2*e98429[i]);
      if e100234[i]=5 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(4.3*e98429[i]);
      if e100234[i]=6 and e100250[i]>=0 and e98429[i]>0 and e99500[i]>0 then
            othpay1[i]=e100250[i]/(e99500[i]*e98429[i]);
      if e100234[i]=8 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(2.15*e98429[i]);
      /*missing value*/   if -4<e98429[i]<0 then othpay1[i]=e98429[i];
      if -4<e100250[i]<0 then othpay1[i]=e100250[i];
      if e98429[i]=0 then othpay1[i]=-3;
   /* non-hourly wage at the beginning, without overtime*/
      if e100234[i]=1 and e100250[i]>=0 then othpay1[i]=e100250[i];
      if e100234[i]=2 and e100250[i]>=0 and e98402d[i]>0 and e98402[i]>0 then
            othpay1[i]=e100250[i]*e98402d[i]/e98402[i];
      if e100234[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100250[i]>=0 and e98402[i]>0
            then othpay1[i]=e100250[i]/e98402[i];
      if e100234[i]=4 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(2*e98402[i]);
      if e100234[i]=5 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(4.3*e98402[i]);
      if e100234[i]=6 and e100250[i]>=0 and e98402[i]>0 and e99500[i]>0 then
            othpay1[i]=e100250[i]/(e99500[i]*e98402[i]);
      if e100234[i]=8 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(2.15*e98402[i]);
      if e100234[i] in (9,14) then othpay1[i]=0;
      /*missing value*/   if -4<e98402[i]<0 then othpay1[i]=e98402[i];
      if e98402[i]=0 then othpay1[i]=-3;
      if e100234[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay1[i]=-3;
      if e100234[i]=2 and -4<e98402d[i]<0 then othpay1[i]=e98402d[i];
      if e100234[i]=2 and -4<e98429l[i]<0 then othpay1[i]=e98429l[i];
      if e100234[i]=6 and e99500[i] le 0 then othpay1[i]=-3;
      if e100234[i]=6 and -4<e99500[i]<0 then othpay1[i]=e99500[i];
   end;
end;

/********* for commissions *********/   /*report hourly wage at the beginning*/
do i=1 to 9;
   if e88000[i]>0 and e102052[i]=1 and e102142[i]>=0 then othpay2[i]=e102142[i];
   if e88000[i]>0 and e102052[i]=2 and e98429l[i]>0 and e102142[i]>=0 then
            othpay2[i]=(e102142[i]*e98429l[i])/e88000[i];
   if e88000[i]>0 and e102052[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102142[i]>=0
            then othpay2[i]=e102142[i]/e88000[i];
   if e88000[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e88000[i]);
   if e88000[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e88000[i]);
```

```
  if e88000[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
         othpay2[i]=e102142[i]/(e99500[i]*e88000[i]);
  if e88000[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e88000[i]);
  /*missing value*/;    if e102142[i]>=0 and -4<e88000[i]<0 then othpay2[i]=e88000[i];
  if e102142[i]>=0 and e88000[i]=0 then othpay2[i]=-3;
end;
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e98429[i]>0 and e102052[i]=1 and e102142[i]>=0 then othpay2[i]=e102142[i];
  if e98429[i]>0 and e102052[i]=2 and e98429l[i]>0 and e102142[i]>=0 then
         othpay2[i]=(e102142[i]*e98429l[i])/e98429[i];
  if e98429[i]>0 and e102052[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102142[i]>=0
         then othpay2[i]=e102142[i]/e98429[i];
  if e98429[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e98429[i]);
  if e98429[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e98429[i]);
  if e98429[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
         othpay2[i]=e102142[i]/(e99500[i]*e98429[i]);
  if e98429[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e98429[i]);
  /*missing value*/;   if e102142[i]>=0 and -4<e98429[i]<0 then othpay2[i]=e98429[i];
  if -4<e102142[i]<0 then othpay2[i]=e102142[i];
  if e102142[i]>=0 and e98429[i]=0 then othpay2[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e98402[i]>0 and e102052[i]=1 and e102142[i]>=0 then othpay2[i]=e102142[i];
  if e98402[i]>0 and e102052[i]=2 and e98402d[i]>0 and e102142[i]>=0 then
         othpay2[i]=(e102142[i]*e98402d[i])/e98402[i];
  if e98402[i]>0 and e102052[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102142[i]>=0
         then othpay2[i]=e102142[i]/e98402[i];
  if e98402[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e98402[i]);
  if e98402[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e98402[i]);
  if e98402[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
         othpay2[i]=e102142[i]/(e99500[i]*e98402[i]);
  if e98402[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e98402[i]);
  if e102052[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf2[i]=othpf2[i]+1;
  if e102052[i] in (9,14) then othpay2[i]=0;
  /*missing value*/;  if e102142[i]>=0 and -4<e98402[i]<0 then othpay2[i]=e98402[i];
  if e102142[i]>=0 and e98402[i]=0 then othpay2[i]=-3;
  if e102052[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay2[i]=-3;
  if e102052[i]=2 and -4<e98402d[i]<0 then othpay2[i]=e98402d[i];
  if e102052[i]=2 and -4<e98429l[i]<0 then othpay2[i]=e98429l[i];
  if e102052[i]=6 and e99500[i] le 0 then othpay2[i]=-3;
  if e102052[i]=6 and -4<e99500[i]<0 then othpay2[i]=e99500[i];
end;
/* if commissions is corrected in the later part */
do i=1 to 9;
  if e102256[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e100235[i]=1 and e100251[i]>=0 then othpay2[i]=e100251[i];
    if e100235[i]=2 and e100251[i]>=0 and e98429l[i]>0 and e88000[i]>0 then
         othpay2[i]=e100251[i]*e98429l[i]/e88000[i];
    if e100235[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100251[i]>=0 and e88000[i]>0
         then othpay2[i]=e100251[i]/e88000[i];
    if e100235[i]=4 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(2*e88000[i]);
    if e100235[i]=5 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(4.3*e88000[i]);
```

```
      if e100235[i]=6 and e100251[i]>=0 and e88000[i]>0 and e99500[i]>0 then
            othpay2[i]=e100251[i]/(e99500[i]*e88000[i]);
      if e100235[i]=8 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(2.15*e88000[i]);
      /*missing value*/  if -4<e88000[i]<0 then othpay2[i]=e88000[i];
      if e88000[i]=0 then othpay2[i]=-3;
   /* non-hourly wage at the beginning, with overtime*/
      if e100235[i]=1 and e100251[i]>=0 then othpay2[i]=e100251[i];
      if e100235[i]=2 and e100251[i]>=0 and e98429l[i]>0 and e98429[i]>0 then
            othpay2[i]=e100251[i]*e98429l[i]/e98429[i];
      if e100235[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100251[i]>=0 and e98429[i]>0
            then othpay2[i]=e100251[i]/e98429[i];
      if e100235[i]=4 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(2*e98429[i]);
      if e100235[i]=5 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(4.3*e98429[i]);
      if e100235[i]=6 and e100251[i]>=0 and e98429[i]>0 and e99500[i]>0 then
            othpay2[i]=e100251[i]/(e99500[i]*e98429[i]);
      if e100235[i]=8 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(2.15*e98429[i]);
      /*missing value*/  if -4<e98429[i]<0 then othpay2[i]=e98429[i];
      if -4<e100251[i]<0 then othpay2[i]=e100251[i];
      if e98429[i]=0 then othpay2[i]=-3;
   /* non-hourly wage at the beginning, without overtime*/
      if e100235[i]=1 and e100251[i]>=0 then othpay2[i]=e100251[i];
      if e100235[i]=2 and e100251[i]>=0 and e98402d[i]>0 and e98402[i]>0 then
            othpay2[i]=e100251[i]*e98402d[i]/e98402[i];
      if e100235[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100251[i]>=0 and e98402[i]>0
            then othpay2[i]=e100251[i]/e98402[i];
      if e100235[i]=4 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(2*e98402[i]);
      if e100235[i]=5 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(4.3*e98402[i]);
      if e100235[i]=6 and e100251[i]>=0 and e98402[i]>0 and e99500[i]>0 then
            othpay2[i]=e100251[i]/(e99500[i]*e98402[i]);
      if e100235[i]=8 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(2.15*e98402[i]);
      if e100235[i] in (9,14) then othpay2[i]=0;
      /*missing value*/  if -4<e98402[i]<0 then othpay2[i]=e98402[i];
      if e98402[i]=0 then othpay2[i]=-3;
      if e100235[i]=2 and e98402d[i] le 0 and e98429l[i] le 0  then othpay2[i]=-3;
      if e100235[i]=2 and -4<e98402d[i]<0 then othpay2[i]=e98402d[i];
      if e100235[i]=2 and -4<e98429l[i]<0 then othpay2[i]=e98429l[i];
      if e100235[i]=6 and e99500[i] le 0 then othpay2[i]=-3;
      if e100235[i]=6 and -4<e99500[i]<0 then othpay2[i]=e99500[i];
   end;
end;

/******** for bonuses ********/   /*report hourly wage at the beginning*/
do i=1 to 9;
   if e88000[i]>0 and e102053[i]=1 and e102143[i]>=0 then othpay3[i]=e102143[i];
   if e88000[i]>0 and e102053[i]=2 and e98429l[i]>0 and e102143[i]>=0 then
            othpay3[i]=(e102143[i]*e98429l[i])/e88000[i];
   if e88000[i]>0 and e102053[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102143[i]>=0
            then othpay3[i]=e102143[i]/e88000[i];
   if e88000[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e88000[i]);
   if e88000[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e88000[i]);
   if e88000[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
            othpay3[i]=e102143[i]/(e99500[i]*e88000[i]);
   if e88000[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e88000[i]);
   /*missing value*/;   if e102143[i]>=0 and -4<e88000[i]<0 then othpay3[i]=e88000[i];
   if e102143[i]>=0 and e88000[i]=0 then othpay3[i]=-3;
end;
```

```
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e98429[i]>0 and e102053[i]=1 and e102143[i]>=0 then othpay3[i]=e102143[i];
  if e98429[i]>0 and e102053[i]=2 and e98429l[i]>0 and e102143[i]>=0 then
         othpay3[i]=(e102143[i]*e98429l[i])/e98429[i];
  if e98429[i]>0 and e102053[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102143[i]>=0
         then othpay3[i]=e102143[i]/e98429[i];
  if e98429[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e98429[i]);
  if e98429[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e98429[i]);
  if e98429[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
         othpay3[i]=e102143[i]/(e99500[i]*e98429[i]);
  if e98429[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e98429[i]);
  /*missing value*/;   if e102143[i]>=0 and -4<e98429[i]<0 then othpay3[i]=e98429[i];
  if -4<e102143[i]<0 then othpay3[i]=e102143[i];
  if e102143[i]>=0 and e98429[i]=0 then othpay3[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e98402[i]>0 and e102053[i]=1 and e102143[i]>=0 then othpay3[i]=e102143[i];
  if e98402[i]>0 and e102053[i]=2 and e98402d[i]>0 and e102143[i]>=0 then
         othpay3[i]=(e102143[i]*e98402d[i])/e98402[i];
  if e98402[i]>0 and e102053[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102143[i]>=0
         then othpay3[i]=e102143[i]/e98402[i];
  if e98402[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e98402[i]);
  if e98402[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e98402[i]);
  if e98402[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
         othpay3[i]=e102143[i]/(e99500[i]*e98402[i]);
  if e98402[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e98402[i]);
  if e102053[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf3[i]=othpf3[i]+1;
  if e102053[i] in (9,14) then othpay3[i]=0;
  /*missing value*/;    if e102143[i]>=0 and -4<e98402[i]<0 then othpay3[i]=e98402[i];
  if e102143[i]>=0 and e98402[i]=0 then othpay3[i]=-3;
  if e102053[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay3[i]=-3;
  if e102053[i]=2 and -4<e98402d[i]<0 then othpay3[i]=e98402d[i];
  if e102053[i]=2 and -4<e98429l[i]<0 then othpay3[i]=e98429l[i];
  if e102053[i]=6 and e99500[i] le 0 then othpay3[i]=-3;
  if e102053[i]=6 and -4<e99500[i]<0 then othpay3[i]=e99500[i];
end;

/* if bonus is corrected in the later part */
do i=1 to 9;
  if e102257[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e100236[i]=1 and e100252[i]>=0 then othpay3[i]=e100252[i];
    if e100236[i]=2 and e100252[i]>=0 and e98429l[i]>0 and e88000[i]>0 then
         othpay3[i]=e100252[i]*e98429l[i]/e88000[i];
    if e100236[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100252[i]>=0 and e88000[i]>0
         then othpay3[i]=e100252[i]/e88000[i];
    if e100236[i]=4 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(2*e88000[i]);
    if e100236[i]=5 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(4.3*e88000[i]);
    if e100236[i]=6 and e100252[i]>=0 and e88000[i]>0 and e99500[i]>0 then
         othpay3[i]=e100252[i]/(e99500[i]*e88000[i]);
    if e100236[i]=8 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(2.15*e88000[i]);
    /*missing value*/   if -4<e88000[i]<0 then othpay3[i]=e88000[i];
    if e98429[i]=0 then othpay3[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
```

```
    if e100236[i]=1 and e100252[i]>=0 then othpay3[i]=e100252[i];
    if e100236[i]=2 and e100252[i]>=0 and e98429l[i]>0 and e98429[i]>0 then
          othpay3[i]=e100252[i]*e98429l[i]/e98429[i];
    if e100236[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100252[i]>=0 and e98429[i]>0
          then othpay3[i]=e100252[i]/e98429[i];
    if e100236[i]=4 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(2*e98429[i]);
    if e100236[i]=5 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(4.3*e98429[i]);
    if e100236[i]=6 and e100252[i]>=0 and e98429[i]>0 and e99500[i]>0 then
          othpay3[i]=e100252[i]/(e99500[i]*e98429[i]);
    if e100236[i]=8 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(2.15*e98429[i]);
    /*missing value*/   if -4<e98429[i]<0 then othpay3[i]=e98429[i];
    if -4<e100252[i]<0 then othpay3[i]=e100252[i];
    if e98429[i]=0 then othpay3[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e100236[i]=1 and e100252[i]>=0 then othpay3[i]=e100252[i];
    if e100236[i]=2 and e100252[i]>=0 and e98402d[i]>0 and e98402[i]>0 then
          othpay3[i]=e100252[i]*e98402d[i]/e98402[i];
    if e100236[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100252[i]>=0 and e98402[i]>0
          then othpay3[i]=e100252[i]/e98402[i];
    if e100236[i]=4 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(2*e98402[i]);
    if e100236[i]=5 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(4.3*e98402[i]);
    if e100236[i]=6 and e100252[i]>=0 and e98402[i]>0 and e99500[i]>0 then
          othpay3[i]=e100252[i]/(e99500[i]*e98402[i]);
    if e100236[i]=8 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(2.15*e98402[i]);
    if e100236[i] in (9,14) then othpay3[i]=0;
    /*missing value*/   if -4<e98402[i]<0 then othpay3[i]=e98402[i];
    if e98402[i]=0 then othpay3[i]=-3;
    if e100236[i]=2 and e98402d[i] le 0  and e98429l[i] le 0 then othpay3[i]=-3;
    if e100236[i]=2 and -4<e98402d[i]<0 then othpay3[i]=e98402d[i];
    if e100236[i]=2 and -4<e98429l[i]<0 then othpay3[i]=e98429l[i];
    if e100236[i]=6 and e99500[i] le 0 then othpay3[i]=-3;
    if e100236[i]=6 and -4<e99500[i]<0 then othpay3[i]=e99500[i];
  end;
end;

/********* for incentive pay *********/   /*report hourly wage at the beginning*/
do i=1 to 9;
  if e88000[i]>0 and e102054[i]=1 and e102144[i]>=0 then othpay4[i]=e102144[i];
  if e88000[i]>0 and e102054[i]=2 and e98429l[i]>0 and e102144[i]>=0 then
          othpay4[i]=(e102144[i]*e98429l[i])/e88000[i];
  if e88000[i]>0 and e102054[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102144[i]>=0
          then othpay4[i]=e102144[i]/e88000[i];
  if e88000[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e88000[i]);
  if e88000[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e88000[i]);
  if e88000[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
          othpay4[i]=e102144[i]/(e99500[i]*e88000[i]);
  if e88000[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e88000[i]);
  /*missing value*/;    if e102144[i]>=0 and -4<e88000[i]<0 then othpay4[i]=e88000[i];
  if e102144[i]>=0 and e88000[i]=0 then othpay4[i]=-3;
end;
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e98429[i]>0 and e102054[i]=1 and e102144[i]>=0 then othpay4[i]=e102144[i];
  if e98429[i]>0 and e102054[i]=2 and e98429l[i]>0 and e102144[i]>=0 then
          othpay4[i]=(e102144[i]*e98429l[i])/e98429[i];
  if e98429[i]>0 and e102054[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102144[i]>=0
          then othpay4[i]=e102144[i]/e98429[i];
  if e98429[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e98429[i]);
  if e98429[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e98429[i]);
  if e98429[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
          othpay4[i]=e102144[i]/(e99500[i]*e98429[i]);
  if e98429[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e98429[i]);
  /*missing value*/;    if e102144[i]>=0 and -4<e98429[i]<0 then othpay4[i]=e98429[i];
  if -4<e102144[i]<0 then othpay4[i]=e102144[i];
  if e102144[i]>=0 and e98429[i]=0 then othpay4[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e98402[i]>0 and e102054[i]=1 and e102144[i]>=0 then othpay4[i]=e102144[i];
  if e98402[i]>0 and e102054[i]=2 and e98402d[i]>0 and e102144[i]>=0 then
          othpay4[i]=(e102144[i]*e98402d[i])/e98402[i];
  if e98402[i]>0 and e102054[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102144[i]>=0
          then othpay4[i]=e102144[i]/e98402[i];
  if e98402[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e98402[i]);
  if e98402[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e98402[i]);
  if e98402[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
          othpay4[i]=e102144[i]/(e99500[i]*e98402[i]);
  if e98402[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e98402[i]);
  if e102054[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf4[i]=othpf4[i]+1;
  if e102054[i] in (9,14) then othpay4[i]=0;
  /*missing value*/;   if e102144[i]>=0 and -4<e98402[i]<0 then othpay4[i]=e98402[i];
  if e102144[i]>=0 and e98402[i]=0 then othpay4[i]=-3;
  if e102054[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay4[i]=-3;
  if e102054[i]=2 and -4<e98402d[i]<0 then othpay4[i]=e98402d[i];
  if e102054[i]=2 and -4<e98429l[i]<0 then othpay4[i]=e98429l[i];
  if e102054[i]=6 and e99500[i] le 0 then othpay4[i]=-3;
  if e102054[i]=6 and -4<e99500[i]<0 then othpay4[i]=e99500[i];
end;
/* if incentive pay is corrected in the later part */
do i=1 to 9;
  if e102258[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e100237[i]=1 and e100253[i]>=0 then othpay4[i]=e100253[i];
    if e100237[i]=2 and e100253[i]>=0 and e98429l[i]>0 and e88000[i]>0 then
          othpay4[i]=e100253[i]*e98429l[i]/e88000[i];
    if e100237[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100253[i]>=0 and e88000[i]>0
          then othpay4[i]=e100253[i]/e88000[i];
    if e100237[i]=4 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(2*e88000[i]);
    if e100237[i]=5 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(4.3*e88000[i]);
    if e100237[i]=6 and e100253[i]>=0 and e88000[i]>0 and e99500[i]>0 then
          othpay4[i]=e100253[i]/(e99500[i]*e88000[i]);
    if e100237[i]=8 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(2.15*e88000[i]);
    /*missing value*/   if -4<e88000[i]<0 then othpay4[i]=e88000[i];
    if -4<e100253[i]<0 then othpay4[i]=e100253[i];
    if e88000[i]=0 then othpay4[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
    if e100237[i]=1 and e100253[i]>=0 then othpay4[i]=e100253[i];
    if e100237[i]=2 and e100253[i]>=0 and e98429l[i]>0 and e98429[i]>0 then
          othpay4[i]=e100253[i]*e98429l[i]/e98429[i];
    if e100237[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100253[i]>=0 and e98429[i]>0
          then othpay4[i]=e100253[i]/e98429[i];
    if e100237[i]=4 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(2*e98429[i]);
    if e100237[i]=5 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(4.3*e98429[i]);
    if e100237[i]=6 and e100253[i]>=0 and e98429[i]>0 and e99500[i]>0 then
         othpay4[i]=e100253[i]/(e99500[i]*e98429[i]);
    if e100237[i]=8 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(2.15*e98429[i]);
    /*missing value*/   if -4<e98429[i]<0 then othpay4[i]=e98429[i];
    if -4<e100253[i]<0 then othpay4[i]=e100253[i];
    if e98429[i]=0 then othpay4[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e100237[i]=1 and e100253[i]>=0 then othpay4[i]=e100253[i];
    if e100237[i]=2 and e100253[i]>=0 and e98402d[i]>0 and e98402[i]>0 then
         othpay4[i]=e100253[i]*e98402d[i]/e98402[i];
    if e100237[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100253[i]>=0 and e98402[i]>0
         then othpay4[i]=e100253[i]/e98402[i];
    if e100237[i]=4 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(2*e98402[i]);
    if e100237[i]=5 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(4.3*e98402[i]);
    if e100237[i]=6 and e100253[i]>=0 and e98402[i]>0 and e99500[i]>0 then
         othpay4[i]=e100253[i]/(e99500[i]*e98402[i]);
    if e100237[i]=8 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(2.15*e98402[i]);
    if e100237[i] in (9,14) then othpay4[i]=0;
    /*missing value*/   if -4<e98402[i]<0 then othpay4[i]=e98402[i];
    if e98402[i]=0 then othpay4[i]=-3;
    if e100237[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay4[i]=-3;
    if e100237[i]=2 and -4<e98402d[i]<0 then othpay4[i]=e98402d[i];
    if e100237[i]=2 and -4<e98429l[i]<0 then othpay4[i]=e98429l[i];
    if e100237[i]=6 and e99500[i] le 0 then othpay4[i]=-3;
    if e100237[i]=6 and -4<e99500[i]<0 then othpay4[i]=e99500[i];
  end;
end;

/******** for others ********/   /*report hourly wage at the beginning*/
do i=1 to 9;
  if e88000[i]>0 and e102055[i]=1 and e102145[i]>=0 then othpay5[i]=e102145[i];
  if e88000[i]>0 and e102055[i]=2 and e98429l[i]>0 and e102145[i]>=0 then
         othpay5[i]=(e102145[i]*e98429l[i])/e88000[i];
  if e88000[i]>0 and e102055[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102145[i]>=0
         then othpay5[i]=e102145[i]/e88000[i];
  if e88000[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e88000[i]);
  if e88000[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e88000[i]);
  if e88000[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
         othpay5[i]=e102145[i]/(e99500[i]*e88000[i]);
  if e88000[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e88000[i]);
  /*missing value*/;   if e102145[i]>=0 and -4<e88000[i]<0 then othpay5[i]=e88000[i];
  if e102145[i]>=0 and e88000[i]=0 then othpay5[i]=-3;
end;
  /* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
  if e98429[i]>0 and e102055[i]=1 and e102145[i]>=0 then othpay5[i]=e102145[i];
  if e98429[i]>0 and e102055[i]=2 and e98429l[i]>0 and e102145[i]>=0 then
         othpay5[i]=(e102145[i]*e98429l[i])/e98429[i];
  if e98429[i]>0 and e102055[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102145[i]>=0
         then othpay5[i]=e102145[i]/e98429[i];
  if e98429[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e98429[i]);
  if e98429[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e98429[i]);
  if e98429[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
         othpay5[i]=e102145[i]/(e99500[i]*e98429[i]);
  if e98429[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e98429[i]);
  /*missing value*/;    if e102145[i]>=0 and -4<e98429[i]<0 then othpay5[i]=e98429[i];
  if -4<e102145[i]<0 then othpay1[i]=e102145[i];
  if e102145[i]>=0 and e98429[i]=0 then othpay5[i]=-3;
end;
  /* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
  if e98402[i]>0 and e102055[i]=1 and e102145[i]>=0 then othpay5[i]=e102145[i];
  if e98402[i]>0 and e102055[i]=2 and e98402d[i]>0 and e102145[i]>=0 then
          othpay5[i]=(e102145[i]*e98402d[i])/e98402[i];
  if e98402[i]>0 and e102055[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e102145[i]>=0
          then othpay5[i]=e102145[i]/e98402[i];
  if e98402[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e98402[i]);
  if e98402[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e98402[i]);
  if e98402[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
          othpay5[i]=e102145[i]/(e99500[i]*e98402[i]);
  if e98402[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e98402[i]);
  if e102055[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf5[i]=othpf5[i]+1;
  if e102055[i] in (9,14) then othpay5[i]=0;
  *missing value*/;    if e102145[i]>=0 and -4<e98402[i]<0 then othpay5[i]=e98402[i];
  if e102145[i]>=0 and e98402[i]=0 then othpay5[i]=-3;
  if e102055[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay5[i]=-3;
  if e102055[i]=2 and -4<e98402d[i]<0 then othpay5[i]=e98402d[i];
  if e102055[i]=2 and -4<e98429l[i]<0 then othpay5[i]=e98429l[i];
  if e102055[i]=6 and e99500[i] le 0 then othpay5[i]=-3;
  if e102055[i]=6 and -4<e99500[i]<0 then othpay5[i]=e99500[i];
end;
/* if other compensation is corrected in the later part */
do i=1 to 9;
  if e102259[i]=1 then do;
  /*report hourly wage at the beginning*/
    if e100239[i]=1 and e100254[i]>=0 then othpay5[i]=e100254[i];
    if e100239[i]=2 and e100254[i]>=0 and e98429l[i]>0 and e88000[i]>0 then
          othpay5[i]=e100254[i]*e98429l[i]/e88000[i];
    if e100239[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100254[i]>=0 and e88000[i]>0
          then othpay5[i]=e100254[i]/e88000[i];
    if e100239[i]=4 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(2*e88000[i]);
    if e100239[i]=5 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(4.3*e88000[i]);
    if e100239[i]=6 and e100254[i]>=0 and e88000[i]>0 and e99500[i]>0 then
          othpay5[i]=e100254[i]/(e99500[i]*e88000[i]);
    if e100239[i]=8 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(2.15*e88000[i]);
    /*missing value*/  if -4<e88000[i]<0 then othpay5[i]=e88000[i];
    if e88000[i]=0 then othpay5[i]=-3;
  /* non-hourly wage at the beginning, with overtime*/
    if e100239[i]=1 and e100254[i]>=0 then othpay5[i]=e100254[i];
    if e100239[i]=2 and e100254[i]>=0 and e98429l[i]>0 and e98429[i]>0 then
          othpay5[i]=e100254[i]*e98429l[i]/e98429[i];
    if e100239[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100254[i]>=0 and e98429[i]>0
          then othpay5[i]=e100254[i]/e98429[i];
    if e100239[i]=4 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(2*e98429[i]);
    if e100239[i]=5 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(4.3*e98429[i]);
    if e100239[i]=6 and e100254[i]>=0 and e98429[i]>0 and e99500[i]>0 then
          othpay5[i]=e100254[i]/(e99500[i]*e98429[i]);
    if e100239[i]=8 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(2.15*e98429[i]);
    /*missing value*/  if -4<e98429[i]<0 then othpay5[i]=e98429[i];
    if -4<e100254[i]<0 then othpay5[i]=e100254[i];
    if e98429[i]=0 then othpay5[i]=-3;
  /* non-hourly wage at the beginning, without overtime*/
    if e100239[i]=1 and e100254[i]>=0 then othpay5[i]=e100254[i];
    if e100239[i]=2 and e100254[i]>=0 and e98402d[i]>0 and e98402[i]>0 then
          othpay5[i]=e100254[i]*e98402d[i]/e98402[i];
    if e100239[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e100254[i]>=0 and e98402[i]>0
          then othpay5[i]=e100254[i]/e98402[i];
    if e100239[i]=4 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(2*e98402[i]);
    if e100239[i]=5 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(4.3*e98402[i]);
    if e100239[i]=6 and e100254[i]>=0 and e98402[i]>0 and e99500[i]>0 then
          othpay5[i]=e100254[i]/(e99500[i]*e98402[i]);
    if e100239[i]=8 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(2.15*e98402[i]);
    if e100239[i] in (9,14) then othpay5[i]=0;
    /*missing value*/   if -4<e98402[i]<0 then othpay5[i]=e98402[i];
    if e98402[i]=0 then othpay5[i]=-3;
    if e100239[i]=2 and e98402d[i] le 0 and e98429l[i] le 0 then othpay5[i]=-3;
    if e100239[i]=2 and -4<e98402d[i]<0 then othpay5[i]=e98402d[i];
    if e100239[i]=2 and -4<e98429l[i]<0 then othpay5[i]=e98429l[i];
    if e100239[i]=6 and e99500[i] le 0 then othpay5[i]=-3;
    if e100239[i]=6 and -4<e99500[i]<0 then othpay5[i]=e99500[i];
  end;
end;

/*********************************** overall start hourly compensation
          ****************************************/
array hrcomp hrcomp01 hrcomp02 hrcomp03 hrcomp04 hrcomp05 hrcomp06 hrcomp07 hrcomp08 hrcomp09;
do i=1 to 9;
  hrcomp[i]=0;
  if hrwg[i] ge 0 then hrcomp[i]=hrcomp[i]+hrwg[i];
  if ot[i] ge 0 then hrcomp[i]=hrcomp[i]+ot[i];
  if othpay1[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay1[i];
  if othpay2[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay2[i];
  if othpay3[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay3[i];
  if othpay4[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay4[i];
  if othpay5[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay5[i];
  if -4<hrwg[i]<0 or -4<ot[i]<0 or -4<othpay1[i]<0 or -4<othpay2[i]<0 or -4<othpay3[i]<0 or -4<othpay4[i]<0 or -
          4<othpay5[i]<0 then hrcomp[i]=-3;
  if hrwg[i]=-4 then hrcomp[i]=-4;
end;


/*********************************Section 2:  End wages for
          youth***************************************/

/*set up the end hourly wage for youths who report an hourly wage*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* without overtime at the beginning*/
    if (e38013[i]=1 and e38014[i]=1 ) then do;
      if e38023[i]>=0 then hrwage[i]=e38023[i];
      /*missing value*/   if -4<e38023[i]<0 then hrwage[i]=e38023[i];
    end;
  /* with overtime at the beginning */
    if (e38106[i]=1 and e38107[i]=1) then do;
      if e38116[i]>=0 then hrwage[i]=e38116[i];
      /*missing value*/      if -4<e38106[i]<0 then hrwage[i]=e38106[i];
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in daily units.*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /*without overtime at the beginning*/
    if (e38013[i]=1 and e38014[i]=2) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e38027[i]>0 and e34402[i]>0 then
          daily[i]=e38023[i]*e38027[i]/e34402[i];
      if e38023[i]>=0 and e3800f[i]>0 and e38027[i]>0 then daily[i]=e38023[i]*e38027[i]/e3800f[i];
      /*missing value*/      if e38023[i]>=0 and -4<e38027[i]<0 then daily[i]=e38027[i];
      if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then daily[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then daily[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then daily[i]=-3;
    end;
  /*with overtime at the beginning*/
    if (e38106[i]=1 and e38107[i]=2) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e38116b[i]>0 and e34428[i]>0 then
          daily[i]=e38116b[i]*e38116[i]/e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 and e38116b[i]>0 then
          daily[i]=e38116b[i]*e38116[i]/e38103[i];
      /*missing value*/     if e38116[i]>=0 and -4<e38116b[i]<0 then daily[i]=e38116b[i];
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then daily[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then daily[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then daily[i]=-3;
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in weekly units.*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /*without overtime at the beginning */
    if (e38013[i]=1 and e38014[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then weekly[i]=e38023[i]/e34402[i];
      if e38023[i]>=0 and e3800f[i]>0 then weekly[i]=e38023[i]/e3800f[i];
      /*missing value*/      if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then weekly[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then weekly[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then weekly[i]=-3;
    end;
  /* with overtime at the beginning */
    if (e38106[i]=1 and e38107[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then weekly[i]=e38116[i]/e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then weekly[i]=e38116[i]/e38103[i];
      /*missing value*/    if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then
          weekly[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then weekly[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then weekly[i]=-3;
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in biwkly units.*/
do i=1 to 9;
```

```
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* without overtime at the beginning */
    if (e38013[i]=1 and e38014[i]=4) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then biwkly[i]=e38023[i]/(2*e34402[i]);
      if e38023[i]>=0 and e3800f[i]>0 then biwkly[i]=e38023[i]/(2*e3800f[i]);
      /*missing value*/    if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then biwkly[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then biwkly[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then biwkly[i]=-3;
    end;
  /* with overtime at the beginning*/
    if (e38106[i]=1 and e38107[i]=4) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then
          biwkly[i]=e38116[i]/(2*e34428[i]);
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then biwkly[i]=e38116[i]/(2*e38103[i]);
      /*missing value*/    if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then
          biwkly[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then biwkly[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then biwkly[i]=-3;
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in month units.*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* without overtime at the beginning */
    if (e38013[i]=1 and e38014[i]=5) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then month[i]=e38023[i]/(4.3*e34402[i]);
      if e38023[i]>=0 and e3800f[i]>0 then month[i]=e38023[i]/(4.3*e3800f[i]);
      /*missing value*/    if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then month[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then month[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then month[i]=-3;
    end;
  /* with overtime at the beginning */
    if (e38106[i]=1 and e38107[i]=5) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then
          month[i]=e38116[i]/(4.3*e34428[i]);
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then month[i]=e38116[i]/(4.3*e38103[i]);
      /*missing value*/  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then
          month[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then month[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then month[i]=-3;
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in annual units.*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* without overtime at the beginning */
    if (e38013[i]=1 and e38014[i]=6) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 and e35600[i]>0 then
          annual[i]=e38023[i]/(e35600[i]*e34402[i]);
      if e38023[i]>=0 and e3800f[i]>0 and e35600[i]>0 then annual[i]=e38023[i]/(e35600[i]*e3800f[i]);
      /*missing value*/ if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then annual[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then annual[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then annual[i]=-3;
      if e35600[i] le 0 then annual[i]=-3;
      if -4<e35600[i]<0 then annual[i]=e35600[i];
    end;
  /* with overtime at the beginning */
    if (e38106[i]=1 and e38107[i]=6) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 and e35600[i]>0 then
         annual[i]=e38116[i]/(e35600[i]*e34428[i]);
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 and e35600[i]>0 then
         annual[i]=e38116[i]/(e35600[i]*e38103[i]);
      /*missing value*/  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then
         annual[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then annual[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then annual[i]=-3;
      if e35600[i] le 0 then annual[i]=-3;
      if -4<e35600[i]<0 then annual[i]=e35600[i];
    end;
  end;
end;

/*set up the end hourly wage for youths who report their wage in semim units.*/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* without overtime at the beginning*/
    if (e38013[i]=1 and e38014[i]=8) then do;
      if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then semim[i]=e38023[i]/(2.15*e34402[i]);
      if e38023[i]>=0 and e3800f[i]>0 then semim[i]=e38023[i]/(2.15*e3800f[i]);
      /*missing value*/    if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then semim[i]=e34402[i];
      if e38023[i]>=0 and -4<e3800f[i]<0 then semim[i]=e3800f[i];
      if e38023[i]>=0 and (e34402[i]=0 or e3800f[i]=0) then semim[i]=-3;
    end;
  /* with overtime at the beginning */
    if (e38106[i]=1 and e38107[i]=8) then do;
      if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then
         semim[i]=e38116[i]/(2.15*e34428[i]);
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then semim[i]=e38116[i]/(2.15*e38103[i]);
      /*missing value*/ if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then
         semim[i]=e34428[i];
      if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then semim[i]=e38103[i];
      if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then semim[i]=-3;
    end;
  end;
end;

/*******************create the hourly rate of pay combining the information from end
         wage*******************/

do i=1 to 9;
  if e37901b[i]=1 or e59900[i]=1 then do;
    if annual[i] ge 0 then hrwg[i]=annual[i]; if month[i] ge 0 then hrwg[i]=month[i];
    if biwkly[i] ge 0 then hrwg[i]=biwkly[i]; if weekly[i] ge 0 then hrwg[i]=weekly[i];
    if daily[i] ge 0 then hrwg[i]=daily[i];       if hrwage[i] ge 0 then hrwg[i]=hrwage[i];
    if semim[i] ge 0 then hrwg[i]=semim[i];
    if hrwage[i] eq -1 or daily[i]=-1 or weekly[i] eq -1 or biwkly[i] eq -1 or month[i] eq -1 or annual[i] eq -1 or
         semim[i]=-1 then hrwg[i]=-1;
    if hrwage[i] eq -2 or daily[i]=-2 or weekly[i] eq -2 or biwkly[i] eq -2 or month[i] eq -2 or annual[i] eq -2 or
         semim[i]=-2 then hrwg[i]=-2;
    if hrwage[i] eq -3 or daily[i]=-3 or weekly[i] eq -3 or biwkly[i] eq -3 or month[i] eq -3 or annual[i] eq -3 or
         semim[i]=-3 then hrwg[i]=-3;
  end;
end;

/*set up the hourly wage for youths who report their wage in other manners */
do i=1 to 9.;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /* if job was not offering any comp at the end */
    if (e38013[i]=1 and e38014[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28)) then otherf[i]=otherf[i]+1;
    if (e38013[i]=1 and e38014[i] in (9,14)) then hrwg[i]=0;
    if e38013[i]=1 and -4<e38023[i]<0 then hrwg[i]=e38023[i];
  /* if job was offering a comp at the end */
    if (e38106[i]=1 and e38107[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28)) then otherf[i]=otherf[i]+1;
    if (e38106[i]=1 and e38107[i] in (9,14)) then hrwg[i]=0;
    if e38106[i]=1 and -4<e38116[i]<0 then hrwg[i]=e38116[i];
  end;
end;

/* missing values */  do i=1 to 9;
  if e37901b[i]=1 or e59900[i]=1 then do;
    if -4<e38023[i]<0 then hrwg[i]=e38023[i];
    if -4<e38116[i]<0 then hrwg[i]=e38116[i];
  end;
end;

/* the end wage if job lasts for >=13 weeks and report the same amount but diff. hours*/
/* note: for this case, we only change hourly rate hrwg[i] without changing hrwage[i], daily[i], weeklw[i] biwkly[i],
         month[i], annual[i] or semim[i]*/
do i=1 to 9;
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and hrwg[i]>=0 and e34402[i]>0 and
         e3800f[i]>0 then hrwg[i]=hrwg[i]*e34402[i]/e3800f[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and e3800f[i]=0 then hrwg[i]=-3;
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and -4<e3800f[i]<0 then hrwg[i]=e3800f[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3)) and hrwg[i]>=0 and
         e34428[i]>0 and e38103[i]>0 then hrwg[i]=hrwg[i]*e34428[i]/e38103[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3)) and hrwg[i]>=0 and
         e34402[i]>0 and e38103[i]>0 then hrwg[i]=hrwg[i]*e34402[i]/e38103[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3)) and e38103[i]=0 then
         hrwg[i]=-3;
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3)) and -4<e38103[i]<0 then
         hrwg[i]=e38103[i];
end;

/******************add the end compensation*****************/

/************************ amount paid of overtime ****************************/
do i=1 to 9;
  if (e37901b[i]=1 or e59900[i]=1) then do;
  /**** without compensation at the beginning *****/
    if e38001[i]=0 then ot[i]=0;
    else do;
      if e38002[i]>0 and e38003[i]=1 and e38012[i]>=0 then ot[i]=e38012[i];
    if e38002[i]>0 and e38003[i]=2 and e38012[i]>=0 and e38012b[i]>0 then
       ot[i]=e38012[i]*e38012b[i]/e38002[i];
    if e38002[i]>0 and e38003[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e38012[i]>=0
       then ot[i]=e38012[i]/e38002[i];
    if e38002[i]>0 and e38003[i]=4 and e38012[i]>=0 then ot[i]=e38012[i]/(2*e38002[i]);
    if e38002[i]>0 and e38003[i]=5 and e38012[i]>=0 then ot[i]=e38012[i]/(4.3*e38002[i]);
    if e38002[i]>0 and e38003[i]=6 and e38012[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
    if e38002[i]>0 and e38003[i]=8 and e38012[i]>=0 then ot[i]=e38012[i]/(2.15*e38002[i]);
    if e38003[i] in (9,14) then ot[i]=0;
    if e38002[i]>0 and e38003[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then otf[i]=otf[i]+1;
    /*missing value*/  if -4<e38002[i]<0 then ot[i]=e38002[i];
    if e38012[i]>=0 and e38002[i]=0 then ot[i]=-3;
    if -4<e38012[i]<0 then ot[i]=e38012[i];
  end;
/****  with compensation at the beginning  *****/
  if e38201[i]=1 then do;
    if e38101[i]=1 then do;       /*same no. of hours as at the beginning */
      if e19200[i]=1 then do;       /* report hourly rate of pay at the beginning*/
        if e24501[i]>0 and e38202[i]=1  and e38211[i]>=0 then ot[i]=e38211[i];
        if e24501[i]>0 and e38202[i]=2  and e38211b[i]>0 and e38211[i]>=0 then
            ot[i]=e38211[i]*e38211b[i]/e24501[i];
        if e24501[i]>0 and e38202[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and
            e38211[i]>=0 then ot[i]=e38211[i]/e24501[i];
        if e24501[i]>0 and e38202[i]=4  and e38211[i]>=0 then ot[i]=e38211[i]/(2*e24501[i]);
        if e24501[i]>0 and e38202[i]=5  and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e24501[i]);
        if e24501[i]>0 and e38202[i]=6  and e38211[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
        if e24501[i]>0 and e38202[i]=8  and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e24501[i]);
        if e38202[i] in (9,14) then ot[i]=0;
        if e24501[i]>0 and e38202[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then otf[i]=otf[i]+1;
        /*missing value*/;  if -4<e24501[i]<0 then ot[i]=e24501[i];
        if e38211[i]>=0 and e24501[i]=0 then ot[i]=-3;
        if -4<e38211[i]<0 then ot[i]=e38211[i];
      end;
      if e19200[i] ne 1 then do;    /*report payment in other units at the begining*/
        if e34403[i]>0 and e38202[i]=1  and e38211[i]>=0 then ot[i]=e38211[i];
        if e34403[i]>0 and e38202[i]=2  and e38211b[i]>0 and e38211[i]>=0 then
            ot[i]=e38211[i]*e38211b[i]/e34403[i];
        if e34403[i]>0 and e38202[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and
            e38211[i]>=0 then ot[i]=e38211[i]/e34403[i];
        if e34403[i]>0 and e38202[i]=4  and e38211[i]>=0 then ot[i]=e38211[i]/(2*e34403[i]);
        if e34403[i]>0 and e38202[i]=5  and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e34403[i]);
        if e34403[i]>0 and e38202[i]=6  and e38211[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
        if e34403[i]>0 and e38202[i]=8  and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e34403[i]);
        if e38202[i] in (9,14) then ot[i]=0;
        if e34403[i]>0 and e38202[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then otf[i]=otf[i]+1;
        /*missing value*/;  if -4<e34403[i]<0 then ot[i]=e34403[i];
        if e38211[i]>=0 and e34403[i]=0 then ot[i]=-3;
        if -4<e38211[i]<0 then ot[i]=e38211[i];
      end;
    end;
    else do;     /* different no. of hours from the beginning*/
      if e38105[i]>0 and e38202[i]=1  and e38211[i]>=0 then ot[i]=e38211[i];
      if e38105[i]>0 and e38202[i]=2  and e38211b[i]>0 and e38211[i]>=0 then
            ot[i]=e38211[i]*e38211b[i]/e38105[i];
      if e38105[i]>0 and e38202[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and e38211[i]>=0
            then ot[i]=e38211[i]/e38105[i];
```

```
          if e38105[i]>0 and e38202[i]=4  and e38211[i]>=0 then ot[i]=e38211[i]/(2*e38105[i]);
          if e38105[i]>0 and e38202[i]=5  and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e38105[i]);
          if e38105[i]>0 and e38202[i]=6  and e38211[i]>=0 then ot[i]=-3; /* snice no weeks per year available. */
          if e38105[i]>0 and e38202[i]=8  and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e38105[i]);
          if e38202[i] in (9,14) then ot[i]=0;
          if e38105[i]>0 and e38202[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then otf[i]=otf[i]+1;
          /*missing value*/;   if -4<e38105[i]<0 then ot[i]=e38105[i];
          if e38211[i]>=0 and e38105[i]=0 then ot[i]=-3;
          if -4<e38211[i]<0 then ot[i]=e38211[i];
        end;
      end;
   end;
end;

/******************** non-overtime payment  ****************************/
do i=1 to 9;
   if (e37901b[i]=1 or e59900[i]=1) then do;
   /*** case i. without overtime at the beginning, same no. of hours. ***/
      if e3800b[i]=1 and e212001[i] ne 1 then do;
       /* report hourly wage at the beginning*/   /* for tips*/
        if e23901[i]>0 and e384071[i]=1  and e384161[i]>=0 then othpay1[i]=e384161[i];
        if e23901[i]>0 and e384071[i]=2  and e34402b[i]>0 and e384161[i]>=0 then
                 othpay1[i]=(e384161[i]*e34402b[i])/e23901[i];
        if e23901[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and e384161[i]>=0
                 then othpay1[i]=e384161[i]/e23901[i];
        if e23901[i]>0 and e384071[i]=4  and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
        if e23901[i]>0 and e384071[i]=5  and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e384071[i]=6  and e384161[i]>=0 and e35600[i]>0 then
                 othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e384071[i]=8  and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
        if e23901[i]>0 and e384071[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                 othpf1[i]=othpf1[i]+1;
        if e384071[i] in (9,14) then othpay1[i]=0;
        /*missing value*/;     if e384161[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
        if e384161[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
        if -4<e384161[i]<0 then othpay1[i]=e384161[i];
        if e384071[i]=2 and e34402b[i] le 0 then othpay1[i]=-3;
        if e384071[i]=2 and -4<e34402b[i]<0 then othpay1[i]=e34402b[i];
        if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
        if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
       /*for commissions*/
        if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
        if e23901[i]>0 and e384072[i]=2 and e34402b[i]>0 and e384162[i]>=0 then
                 othpay2[i]=(e384162[i]*e34402b[i])/e23901[i];
        if e23901[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384162[i]>=0
                 then othpay2[i]=e384162[i]/e23901[i];
        if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
        if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
                 othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
        if e23901[i]>0 and e384072[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                 othpf2[i]=othpf2[i]+1;
        if e384072[i] in (9,14) then othpay2[i]=0;
        /*missing value*/;    if e384162[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
        if e384162[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
```

```
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
  if e384072[i]=2 and e34402b[i] le 0 then othpay2[i]=-3;
  if e384072[i]=2 and -4<e34402b[i]<0 then othpay2[i]=e34402b[i];
  if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
/*for bonuses*/
  if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e23901[i]>0 and e384073[i]=2 and e34402b[i]>0 and e384163[i]>=0 then
          othpay3[i]=(e384163[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384163[i]>=0
          then othpay3[i]=e384163[i]/e23901[i];
  if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
  if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
          othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384073[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
  if e384073[i] in (9,14) then othpay3[i]=0;
 /*missing value*/;   if e384163[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
  if e384163[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay3[i]=e384163[i];
  if e384073[i]=2 and e34402b[i] le 0 then othpay3[i]=-3;
  if e384073[i]=2 and -4<e34402b[i]<0 then othpay3[i]=e34402b[i];
  if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
/*for incentive pay*/
  if e23901[i]>0 and e384074[i]=1  and e384164[i]>=0 then othpay4[i]=e384164[i];
  if e23901[i]>0 and e384074[i]=2  and e34402b[i]>0 and e384164[i]>=0 then
          othpay4[i]=(e384164[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and e384164[i]>=0
          then othpay4[i]=e384164[i]/e23901[i];
  if e23901[i]>0 and e384074[i]=4  and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
  if e23901[i]>0 and e384074[i]=5  and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384074[i]=6  and e384164[i]>=0 and e35600[i]>0 then
          othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384074[i]=8  and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384074[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf4[i]=othpf4[i]+1;
  if e384074[i] in (9,14) then othpay4[i]=0;
  /*missing value*/;    if e384164[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
  if e384164[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
  if -4<e384164[i]<0 then othpay4[i]=e384164[i];
  if e384074[i]=2 and e34402b[i] le 0 then othpay4[i]=-3;
  if e384074[i]=2 and -4<e34402b[i]<0 then othpay4[i]=e34402b[i];
  if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
  if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
/*for others*/
  if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
  if e23901[i]>0 and e384075[i]=2 and e34402b[i]>0 and e384165[i]>=0 then
          othpay5[i]=(e384165[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384165[i]>=0
          then othpay5[i]=e384165[i]/e23901[i];
  if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);
  if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
```

```
    if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
            othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
    if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
    if e23901[i]>0 and e384075[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf5[i]=othpf5[i]+1;
    if e384075[i] in (9,14) then othpay5[i]=0;
    /*missing value*/;    if e384165[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
    if e384165[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
    if -4<e384165[i]<0 then othpay5[i]=e384165[i];
    if e384075[i]=2 and e34402b[i] le 0 then othpay5[i]=-3;
    if e384075[i]=2 and -4<e34402b[i]<0 then othpay5[i]=e34402b[i];
    if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
    if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/* report nonhourly wage at the beginning*/  /* for tips*/
    if e34402[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
    if e34402[i]>0 and e384071[i]=2 and e34402b[i]>0 and e384161[i]>=0 then
            othpay1[i]=(e384161[i]*e34402b[i])/e34402[i];
    if e34402[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384161[i]>=0
            then othpay1[i]=e384161[i]/e34402[i];
    if e34402[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34402[i]);
    if e34402[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34402[i]);
    if e34402[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
            othpay1[i]=e384161[i]/(e35600[i]*e34402[i]);
    if e34402[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34402[i]);
    if e34402[i]>0 and e384071[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf1[i]=othpf1[i]+1;
    /*missing value*/;    if e384161[i]>=0 and -4<e34402[i]<0 then othpay1[i]=e34402[i];
    if e384161[i]>=0 and e34402[i]=0 then othpay1[i]=-3;
    if -4<e384161[i]<0 then othpay1[i]=e384161[i];
  /*for commissions*/
    if e34402[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
    if e34402[i]>0 and e384072[i]=2 and e34402b[i]>0 and e384162[i]>=0 then
            othpay2[i]=(e384162[i]*e34402b[i])/e34402[i];
    if e34402[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384162[i]>=0
            then othpay2[i]=e384162[i]/e34402[i];
    if e34402[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34402[i]);
    if e34402[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34402[i]);
    if e34402[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
            othpay2[i]=e384162[i]/(e35600[i]*e34402[i]);
    if e34402[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34402[i]);
    if e34402[i]>0 and e384072[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf2[i]=othpf2[i]+1;
    /*missing value*/;    if e384162[i]>=0 and -4<e34402[i]<0 then othpay2[i]=e34402[i];
    if e384162[i]>=0 and e34402[i]=0 then othpay2[i]=-3;
    if -4<e384162[i]<0 then othpay2[i]=e384162[i];
  /*for bonuses*/
    if e34402[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
    if e34402[i]>0 and e384073[i]=2 and e34402b[i]>0 and e384163[i]>=0 then
            othpay3[i]=(e384163[i]*e34402b[i])/e34402[i];
    if e34402[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384163[i]>=0
            then othpay3[i]=e384163[i]/e34402[i];
    if e34402[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34402[i]);
    if e34402[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34402[i]);
    if e34402[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
            othpay3[i]=e384163[i]/(e35600[i]*e34402[i]);
```

```
      if e34402[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34402[i]);
      if e34402[i]>0 and e384073[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
              othpf3[i]=othpf3[i]+1;
      /*missing value*/;    if e384163[i]>=0 and -4<e34402[i]<0 then othpay3[i]=e34402[i];
      if e384163[i]>=0 and e34402[i]=0 then othpay3[i]=-3;
      if -4<e384163[i]<0 then othpay3[i]=e384163[i];
    /*for incentive pay*/
      if e34402[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
      if e34402[i]>0 and e384074[i]=2 and e34402b[i]>0 and e384164[i]>=0 then
              othpay4[i]=(e384164[i]*e34402b[i])/e34402[i];
      if e34402[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384164[i]>=0
              then othpay4[i]=e384164[i]/e34402[i];
      if e34402[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34402[i]);
      if e34402[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34402[i]);
      if e34402[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
              othpay4[i]=e384164[i]/(e35600[i]*e34402[i]);
      if e34402[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34402[i]);
      if e34402[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
              othpf4[i]=othpf4[i]+1;
      /*missing value*/;    if e384164[i]>=0 and -4<e34402[i]<0 then othpay4[i]=e34402[i];
      if e384164[i]>=0 and e34402[i]=0 then othpay4[i]=-3;
      if -4<e384164[i]<0 then othpay4[i]=e384164[i];
    /*for others*/
      if e34402[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
      if e34402[i]>0 and e384075[i]=2 and e34402b[i]>0 and e384165[i]>=0 then
              othpay5[i]=(e384165[i]*e34402b[i])/e34402[i];
      if e34402[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384165[i]>=0
              then othpay5[i]=e384165[i]/e34402[i];
      if e34402[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34402[i]);
      if e34402[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34402[i]);
      if e34402[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
              othpay5[i]=e384165[i]/(e35600[i]*e34402[i]);
      if e34402[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34402[i]);
      if e34402[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
              othpf5[i]=othpf5[i]+1;
      /*missing value*/;   if e384165[i]>=0 and -4<e34402[i]<0 then othpay5[i]=e34402[i];
      if e384165[i]>=0 and e34402[i]=0 then othpay5[i]=-3;
      if -4<e384165[i]<0 then othpay5[i]=e384165[i];
    end;

/*** case ii. without overtime at the beginning, diff no. of hours.***/
    if e3800b[i]=0 and e212001[i] ne 1 then do;
    /* for tips*/
      if e3800f[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
      if e3800f[i]>0 and e384071[i]=2 and e38027[i]>0 and e384161[i]>=0 then
              othpay1[i]=(e384161[i]*e38027[i])/e3800f[i];
      if e3800f[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384161[i]>=0
              then othpay1[i]=e384161[i]/e3800f[i];
      if e3800f[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e3800f[i]);
      if e3800f[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e3800f[i]);
      if e3800f[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
              othpay1[i]=e384161[i]/(e35600[i]*e3800f[i]);
      if e3800f[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e3800f[i]);
      if e3800f[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
              othpf1[i]=othpf1[i]+1;
      if e384071[i] in (9,14) then othpay1[i]=0;
```

```
  /*missing value*/;   if e384161[i]>=0 and -4<e3800f[i]<0 then othpay1[i]=e3800f[i];
  if e384161[i]>=0 and e3800f[i]=0 then othpay1[i]=-3;
  if -4<e384161[i]<0 then othpay1[i]=e384161[i];
  if e384071[i]=2 and e38027[i]=0 then othpay1[i]=-3;
  if e384071[i]=2 and -4<e38027[i]<0 then othpay1[i]=e38027[i];
  if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
  if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
/*for commissions*/
  if e3800f[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e3800f[i]>0 and e384072[i]=2 and e38027[i]>0 and e384162[i]>=0 then
          othpay2[i]=(e384162[i]*e38027[i])/e3800f[i];
  if e3800f[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384162[i]>=0
          then othpay2[i]=e384162[i]/e3800f[i];
  if e3800f[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e3800f[i]);
  if e3800f[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e3800f[i]);
  if e3800f[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
          othpay2[i]=e384162[i]/(e35600[i]*e3800f[i]);
  if e3800f[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e3800f[i]);
  if e3800f[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf2[i]=othpf2[i]+1;
  if e384072[i] in (9,14) then othpay2[i]=0;
  /*missing value*/;    if e384162[i]>=0 and -4<e3800f[i]<0 then othpay2[i]=e3800f[i];
  if e384162[i]>=0 and e3800f[i]=0 then othpay2[i]=-3;
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
  if e384072[i]=2 and e38027[i]=0 then othpay2[i]=-3;
  if e384072[i]=2 and -4<e38027[i]<0 then othpay2[i]=e38027[i];
  if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
/*for bonuses*/
  if e3800f[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e3800f[i]>0 and e384073[i]=2 and e38027[i]>0 and e384163[i]>=0 then
          othpay3[i]=(e384163[i]*e38027[i])/e3800f[i];
  if e3800f[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384163[i]>=0
          then othpay3[i]=e384163[i]/e3800f[i];
  if e3800f[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e3800f[i]);
  if e3800f[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e3800f[i]);
  if e3800f[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
          othpay3[i]=e384163[i]/(e35600[i]*e3800f[i]);
  if e3800f[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e3800f[i]);
  if e3800f[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
  if e384073[i] in (9,14) then othpay3[i]=0;
  /*missing value*/;    if e384163[i]>=0 and -4<e3800f[i]<0 then othpay3[i]=e3800f[i];
  if e384163[i]>=0 and e3800f[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay1[i]=e384163[i];
  if e384073[i]=2 and e38027[i]=0 then othpay3[i]=-3;
  if e384073[i]=2 and -4<e38027[i]<0 then othpay3[i]=e38027[i];
  if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
/*for incentive pay*/
  if e3800f[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
  if e3800f[i]>0 and e384074[i]=2 and e38027[i]>0 and e384164[i]>=0 then
          othpay4[i]=(e384164[i]*e38027[i])/e3800f[i];
  if e3800f[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384164[i]>=0
          then othpay4[i]=e384164[i]/e3800f[i];
  if e3800f[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e3800f[i]);
```

```
      if e3800f[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e3800f[i]);
      if e3800f[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
                othpay4[i]=e384164[i]/(e35600[i]*e3800f[i]);
      if e3800f[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e3800f[i]);
      if e3800f[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                othpf4[i]=othpf4[i]+1;
      if e384074[i] in (9,14) then othpay4[i]=0;
      /*missing value*/;    if e384164[i]>=0 and -4<e3800f[i]<0 then othpay4[i]=e3800f[i];
      if e384164[i]>=0 and e3800f[i]=0 then othpay4[i]=-3;
      if -4<e384164[i]<0 then othpay4[i]=e384164[i];
      if e384074[i]=2 and e38027[i]=0 then othpay4[i]=-3;
      if e384074[i]=2 and -4<e38027[i]<0 then othpay4[i]=e38027[i];
      if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
      if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
    /*for others*/
      if e3800f[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
      if e3800f[i]>0 and e384075[i]=2 and e38027[i]>0 and e384165[i]>=0 then
                othpay5[i]=(e384165[i]*e38027[i])/e3800f[i];
      if e3800f[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384165[i]>=0
                then othpay5[i]=e384165[i]/e3800f[i];
      if e3800f[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e3800f[i]);
      if e3800f[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e3800f[i]);
      if e3800f[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
                othpay5[i]=e384165[i]/(e35600[i]*e3800f[i]);
      if e3800f[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e3800f[i]);
      if e3800f[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                othpf5[i]=othpf5[i]+1;
      if e384075[i] in (9,14) then othpay5[i]=0;
      /*missing value*/;    if e384165[i]>=0 and -4<e3800f[i]<0 then othpay5[i]=e3800f[i];
      if e384165[i]>=0 and e3800f[i]=0 then othpay5[i]=-3;
      if -4<e384165[i]<0 then othpay5[i]=e384165[i];
      if e384075[i]=2 and e38027[i]=0 then othpay5[i]=-3;
      if e384075[i]=2 and -4<e38027[i]<0 then othpay5[i]=e38027[i];
      if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
      if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
    end;

/*** case iii. only one compensation at the beginning,same no. of hours. ***/
    if e38329[i]=1 and e20700[i]=1 then do;
    /* for tips*/
      if e212002[i]=1 then do;
    /*with overtime at the beginning*/
        if e38102[i] ne 1 and e38102[i] ne 3 then do;
        /*report hourly wage at the beginning*/
          if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
          if e23901[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
                othpay1[i]=(e38329d[i]*e34430[i])/e23901[i];
          if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
                e38329d[i]>=0 then othpay1[i]=e38329d[i]/e23901[i];
          if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e23901[i]);
          if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e23901[i]);
          if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
                othpay1[i]=e38329d[i]/(e35600[i]*e23901[i]);
          if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e23901[i]);
          if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                othpf1[i]=othpf1[i]+1;
```

```
  /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
  if e38329d[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
  if -4<e38329d[i]<0 then othpay1[i]=e38329d[i];
/*report nonhourly wage*/
  if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
  if e34428[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
       othpay1[i]=(e38329d[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay1[i]=e38329d[i]/e34428[i];
  if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e34428[i]);
  if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
       othpay1[i]=e38329d[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e34428[i]);
  if e34428[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
       othpf1[i]=othpf1[i]+1;
  if e38329b[i] in (9,14) then othpay1[i]=0;
  /*missing value*/;    if e38329d[i]>=0 and -4<e34428[i]<0 then othpay1[i]=e34428[i];
  if e38329d[i]>=0 and e34428[i]=0 then othpay1[i]=-3;
  if -4<e38329d[i]<0 then othpay1[i]=e38329d[i];
  if e38329b[i]=2 and e34430[i] le 0 then othpay1[i]=-3;
  if e38329b[i]=2 and -4<e34430[i]<0 then othpay1[i]=e34430[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/*report hourly wage at the beginning*/
  if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
  if e23901[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
       othpay1[i]=(e38329d[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay1[i]=e38329d[i]/e23901[i];
  if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e23901[i]);
  if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
       othpay1[i]=e38329d[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
       othpf1[i]=othpf1[i]+1;
  /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
  if e38329d[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
  if -4<e38329d[i]<0 then othpay1[i]=e38329d[i];
/*report non-hourly wage at the beginning*/
  if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
  if e34402[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
       othpay1[i]=(e38329d[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay1[i]=e38329d[i]/e34402[i];
  if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e34402[i]);
  if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
       othpay1[i]=e38329d[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
       othpf1[i]=othpf1[i]+1;
```

```
      if e38329b[i] in (9,14) then othpay1[i]=0;
      /*missing value*/;    if e38329d[i]>=0 and -4<e34402[i]<0 then othpay1[i]=e34402[i];
      if e38329d[i]>=0 and e34402[i]=0 then othpay1[i]=-3;
      if e38329b[i]=2 and e34402b[i] le 0 then othpay1[i]=-3;
      if e38329b[i]=2 and -4<e34402b[i]<0 then othpay1[i]=e34402b[i];
      if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
      if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
    end;
  end;
/*for commissions*/
  if e212003[i]=1 then do;
  /*with overtime at the beginning*/
    if e38102[i] ne 1 and e38102[i] ne 3 then do;
    /*report hourly wage at the beginning*/
      if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
      if e23901[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
           othpay2[i]=(e38329d[i]*e34430[i])/e23901[i];
      if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
           e38329d[i]>=0 then othpay2[i]=e38329d[i]/e23901[i];
      if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e23901[i]);
      if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e23901[i]);
      if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
           othpay2[i]=e38329d[i]/(e35600[i]*e23901[i]);
      if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e23901[i]);
      if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
           othpf1[i]=othpf1[i]+1;
      /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
      if e38329d[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
      if -4<e38329d[i]<0 then othpay2[i]=e38329d[i];
    /*report nonhourly wage*/
      if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
      if e34428[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
           othpay2[i]=(e38329d[i]*e34430[i])/e34428[i];
      if e34428[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
           e38329d[i]>=0 then othpay2[i]=e38329d[i]/e34428[i];
      if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e34428[i]);
      if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e34428[i]);
      if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
           othpay2[i]=e38329d[i]/(e35600[i]*e34428[i]);
      if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e34428[i]);
      if e34428[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
           othpf2[i]=othpf2[i]+1;
      if e38329b[i] in (9,14) then othpay2[i]=0;
      /*missing value*/;  if e38329d[i]>=0 and -4<e34428[i]<0 then othpay2[i]=e34428[i];
      if e38329d[i]>=0 and e34428[i]=0 then othpay2[i]=-3;
      if -4<e38329d[i]<0 then othpay2[i]=e38329d[i];
      if e38329b[i]=2 and e34430[i] le 0 then othpay2[i]=-3;
      if e38329b[i]=2 and -4<e34430[i]<0 then othpay2[i]=e34430[i];
      if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
      if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
    end;
  /*without overtime at the beginning*/
    if e3800b[i]=1 then do;
    /*report hourly wage at the beginning*/
      if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
      if e23901[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
          othpay2[i]=(e38329d[i]*e34402b[i])/e23901[i];
      if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay2[i]=e38329d[i]/e23901[i];
      if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e23901[i]);
      if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e23901[i]);
      if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay2[i]=e38329d[i]/(e35600[i]*e23901[i]);
      if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e23901[i]);
      if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf1[i]=othpf1[i]+1;
      /*missing value*/;   if e38329d[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
      if e38329d[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
      if -4<e38329d[i]<0 then othpay2[i]=e38329d[i];
    /*report nonhourly wage*/
      if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
      if e34402[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
          othpay2[i]=(e38329d[i]*e34402b[i])/e34402[i];
      if e34402[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay2[i]=e38329d[i]/e34402[i];
      if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e34402[i]);
      if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e34402[i]);
      if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay2[i]=e38329d[i]/(e35600[i]*e34402[i]);
      if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e34402[i]);
      if e34402[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf2[i]=othpf2[i]+1;
      if e38329b[i] in (9,14) then othpay2[i]=0;
      /*missing value*/;    if e38329d[i]>=0 and -4<e34402[i]<0 then othpay2[i]=e34402[i];
      if e38329d[i]>=0 and e34402[i]=0 then othpay2[i]=-3;
      if e38329b[i]=2 and e34402b[i] le 0 then othpay2[i]=-3;
      if e38329b[i]=2 and -4<e34402b[i]<0 then othpay2[i]=e34402b[i];
      if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
      if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
    end;
  end;
/*for bonuses*/
  if e212004[i]=1 then do;
    /*with overtime at the beginning*/
    if e38102[i] ne 1 and e38102[i] ne 3 then do;
    /* report hourly wage at the beginning*/
      if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
      if e23901[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
          othpay3[i]=(e38329d[i]*e34430[i])/e23901[i];
      if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay3[i]=e38329d[i]/e23901[i];
      if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e23901[i]);
      if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e23901[i]);
      if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay3[i]=e38329d[i]/(e35600[i]*e23901[i]);
      if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e23901[i]);
      if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
      */missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
      if e38329d[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
      if -4<e38329d[i]<0 then othpay3[i]=e38329d[i];
```

```
/*report non-hourly wage*/
   if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
   if e34428[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
          othpay3[i]=(e38329d[i]*e34430[i])/e34428[i];
   if e34428[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay3[i]=e38329d[i]/e34428[i];
   if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e34428[i]);
   if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e34428[i]);
   if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay3[i]=e38329d[i]/(e35600[i]*e34428[i]);
   if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e34428[i]);
   if e34428[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
   if e38329b[i] in (9,14) then othpay3[i]=0;
   /*missing value*/;   if e38329d[i]>=0 and -4<e34428[i]<0 then othpay3[i]=e34428[i];
   if e38329d[i]>=0 and e34428[i]=0 then othpay3[i]=-3;
   if -4<e38329d[i]<0 then othpay3[i]=e38329d[i];
   if e38329b[i]=2 and e34430[i] le 0 then othpay3[i]=-3;
   if e38329b[i]=2 and -4<e34430[i]<0 then othpay3[i]=e34430[i];
   if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
   if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
   if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
   if e23901[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
          othpay3[i]=(e38329d[i]*e34402b[i])/e23901[i];
   if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay3[i]=e38329d[i]/e23901[i];
   if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e23901[i]);
   if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e23901[i]);
   if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay3[i]=e38329d[i]/(e35600[i]*e23901[i]);
   if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e23901[i]);
   if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
   /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
   if e38329d[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
   if -4<e38329d[i]<0 then othpay3[i]=e38329d[i];
/*report non-hourly wage*/
   if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
   if e34402[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
          othpay3[i]=(e38329d[i]*e34402b[i])/e34402[i];
   if e34402[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
          e38329d[i]>=0 then othpay3[i]=e38329d[i]/e34402[i];
   if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e34402[i]);
   if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e34402[i]);
   if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
          othpay3[i]=e38329d[i]/(e35600[i]*e34402[i]);
   if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e34402[i]);
   if e34402[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
   if e38329b[i] in (9,14) then othpay3[i]=0;
   /*missing value*/;    if e38329d[i]>=0 and -4<e34402[i]<0 then othpay3[i]=e34402[i];
   if e38329d[i]>=0 and e34402[i]=0 then othpay3[i]=-3;
```

```
    if e38329b[i]=2 and e34402b[i] le 0 then othpay3[i]=-3;
    if e38329b[i]=2 and -4<e34402b[i]<0 then othpay3[i]=e34402b[i];
    if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
    if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
  end;
end;
/*for incentive pay*/
if e212005[i]=1 then do;
/*with overtime at the beginning*/
  if e38102[i] ne 1 and e38102[i] ne 3 then do;
  /* report hourly wage at the beginning*/
    if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
    if e23901[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
         othpay4[i]=(e38329d[i]*e34430[i])/e23901[i];
    if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
         e38329d[i]>=0 then othpay4[i]=e38329d[i]/e23901[i];
    if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e23901[i]);
    if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e23901[i]);
    if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
         othpay4[i]=e38329d[i]/(e35600[i]*e23901[i]);
    if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e23901[i]);
    if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
         othpf4[i]=othpf4[i]+1;
    /*missing value*/;  if e38329d[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
    if e38329d[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
    if -4<e38329d[i]<0 then othpay4[i]=e38329d[i];
  /* report nonhourly wage*/
    if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
    if e34428[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
         othpay4[i]=(e38329d[i]*e34430[i])/e34428[i];
    if e34428[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
         e38329d[i]>=0 then othpay4[i]=e38329d[i]/e34428[i];
    if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e34428[i]);
    if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e34428[i]);
    if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
         othpay4[i]=e38329d[i]/(e35600[i]*e34428[i]);
    if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e34428[i]);
    if e34428[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
         othpf4[i]=othpf4[i]+1;
    if e38329b[i] in (9,14) then othpay4[i]=0;
    /*missing value*/;    if e38329d[i]>=0 and -4<e34428[i]<0 then othpay4[i]=e34428[i];
    if e38329d[i]>=0 and e34428[i]=0 then othpay4[i]=-3;
    if -4<e38329d[i]<0 then othpay4[i]=e38329d[i];
    if e38329b[i]=2 and e34430[i] le 0 then othpay4[i]=-3;
    if e38329b[i]=2 and -4<e34430[i]<0 then othpay4[i]=e34430[i];
    if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
    if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
  end;
  /*without overtime at the beginning*/
  if e3800b[i]=1 then do;
  /* report hourly wage at the beginning*/
    if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
    if e23901[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
         othpay4[i]=(e38329d[i]*e34402b[i])/e23901[i];
    if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
         e38329d[i]>=0 then othpay4[i]=e38329d[i]/e23901[i];
```

```
        if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e23901[i]);
        if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay4[i]=e38329d[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e23901[i]);
        if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
             othpf4[i]=othpf4[i]+1;
        /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
        if e38329d[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
        if -4<e38329d[i]<0 then othpay4[i]=e38329d[i];
      /* report nonhourly wage*/
        if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
        if e34402[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
             othpay4[i]=(e38329d[i]*e34402b[i])/e34402[i];
        if e34402[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
             e38329d[i]>=0 then othpay4[i]=e38329d[i]/e34402[i];
        if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e34402[i]);
        if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e34402[i]);
        if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay4[i]=e38329d[i]/(e35600[i]*e34402[i]);
        if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e34402[i]);
        if e34402[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
             othpf4[i]=othpf4[i]+1;
        if e38329b[i] in (9,14) then othpay4[i]=0;
        /*missing value*/;   if e38329d[i]>=0 and -4<e34402[i]<0 then othpay4[i]=e34402[i];
        if e38329d[i]>=0 and e34402[i]=0 then othpay4[i]=-3;
        if e38329b[i]=2 and e34402b[i] le 0 then othpay4[i]=-3;
        if e38329b[i]=2 and -4<e34402b[i]<0 then othpay4[i]=e34402b[i];
        if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
        if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
      end;
    end;
    /*for others*/
    if e212006[i]=1 then do;
    /*with overtime at the beginning*/
      if e38102[i] ne 1 and e38102[i] ne 3 then do;
      /* report hourly wage at the beginning*/
        if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
        if e23901[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
             othpay5[i]=(e38329d[i]*e34430[i])/e23901[i];
        if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
             e38329d[i]>=0 then othpay5[i]=e38329d[i]/e23901[i];
        if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e23901[i]);
        if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay5[i]=e38329d[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e23901[i]);
        if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24.25,26,28) then
             othpf5[i]=othpf5[i]+1;
        /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
        if e38329d[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
        if -4<e38329d[i]<0 then othpay5[i]=e38329d[i];
      /*report non-hourly wage*/
        if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
        if e34428[i]>0 and e38329b[i]=2 and e34430[i]>0 and e38329d[i]>=0 then
             othpay5[i]=(e38329d[i]*e34430[i])/e34428[i];
```

```
  if e34428[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay5[i]=e38329d[i]/e34428[i];
  if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e34428[i]);
  if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
       othpay5[i]=e38329d[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e34428[i]);
  if e34428[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24.25,26,28) then
       othpf5[i]=othpf5[i]+1;
  if e38329b[i] in (9,14) then othpay5[i]=0;
  /*missing value*/;    if e38329d[i]>=0 and -4<e34428[i]<0 then othpay5[i]=e34428[i];
  if e38329d[i]>=0 and e34428[i]=0 then othpay5[i]=-3;
  if -4<e38329d[i]<0 then othpay5[i]=e38329d[i];
  if e38329b[i]=2 and e34430[i] le 0 then othpay5[i]=-3;
  if e38329b[i]=2 and -4<e34430[i]<0 then othpay5[i]=e34430[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
  if e23901[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
       othpay5[i]=(e38329d[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay5[i]=e38329d[i]/e23901[i];
  if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e23901[i]);
  if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
       othpay5[i]=e38329d[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,1718,19,21,22,23,24.25,26,28) then
       othpf5[i]=othpf5[i]+1;
  /*missing value*/;    if e38329d[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
  if e38329d[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
  if -4<e38329d[i]<0 then othpay5[i]=e38329d[i];
/*report non-hourly wage*/
  if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
  if e34402[i]>0 and e38329b[i]=2 and e34402b[i]>0 and e38329d[i]>=0 then
       othpay5[i]=(e38329d[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
       e38329d[i]>=0 then othpay5[i]=e38329d[i]/e34402[i];
  if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e34402[i]);
  if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(52*e34402[i]);
  if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
       othpf5[i]=othpf5[i]+1;
  if e38329b[i] in (9,14) then othpay5[i]=0;
  /*missing value*/;   if e38329d[i]>=0 and -4<e34402[i]<0 then othpay5[i]=e34402[i];
  if e38329d[i]>=0 and e34402[i]=0 then othpay5[i]=-3;
  if e38329b[i]=2 and e34402b[i] le 0 then othpay5[i]=-3;
  if e38329b[i]=2 and -4<e34402b[i]<0 then othpay5[i]=e34402b[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
end;
```

```
      end;
    end;

/*** case iv. one compensation at the beginning, diff no. of hours. ***/
    if e38329[i]=1 and e20700[i]=1 then do;
    /* for tips*/
      if e212002[i]=1 then do;
      /* without overtime at the beginning*/
        if e3800b[i]=0 then do;
          if e3800f[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
          if e3800f[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
               othpay1[i]=(e38329d[i]*e38116b[i])/e3800f[i];
          if e3800f[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
               e38329d[i]>=0 then othpay1[i]=e38329d[i]/e3800f[i];
          if e3800f[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e3800f[i]);
          if e3800f[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e3800f[i]);
          if e3800f[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
               othpay1[i]=e38329d[i]/(e35600[i]*e3800f[i]);
          if e3800f[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e3800f[i]);
          if e3800f[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
               othpf1[i]=othpf1[i]+1;
          /*missing value*/;    if e38329d[i]>=0 and -4<e3800f[i]<0 then othpay1[i]=e3800f[i];
          if e38329d[i]>=0 and e3800f[i]=0 then othpay1[i]=-3;
          if -4<e38329d[i]<0 then othpay1[i]=e38329d[i];
          if e38329b[i]=2 and e38116b[i]=0 then othpay1[i]=-3;
          if e38329b[i]=2 and -4<e38116b[i]<0 then othpay1[i]=e38116b[i];
          if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
          if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
        end;
        /* with overtime at the beginning*/
        if e38102[i]=1 or e38102[i]=3 then do;
          if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
          if e38103[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
               othpay1[i]=(e38329d[i]*e38116b[i])/e38103[i];
          if e38103[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
               e38329d[i]>=0 then othpay1[i]=e38329d[i]/e38103[i];
          if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e38103[i]);
          if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e38103[i]);
          if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
               othpay1[i]=e38329d[i]/(e35600[i]*e38103[i]);
          if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e38103[i]);
          if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
               othpf1[i]=othpf1[i]+1;
          if e38329b[i] in (9,14) then othpay1[i]=0;
          /*missing value*/;    if e38329d[i]>=0 and -4<e38103[i]<0 then othpay1[i]=e38103[i];
          if e38329d[i]>=0 and e38103[i]=0 then othpay1[i]=-3;
          if -4<e38329d[i]<0 then othpay1[i]=e38329d[i];
          if e38329b[i]=2 and e38116b[i]=0 then othpay1[i]=-3;
          if e38329b[i]=2 and -4<e38116b[i]<0 then othpay1[i]=e38116b[i];
          if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
          if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
        end;
     end;
     /*for commissions*/
     if e212003[i]=1 then do;
     /*without overtime at the beginning*/
```

```
  if e3800f[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
  if e3800f[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
        othpay2[i]=(e38329d[i]*e38116b[i])/e3800f[i];
  if e3800f[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e38329d[i]>=0 then othpay2[i]=e38329d[i]/e3800f[i];
  if e3800f[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay2[i]=e38329d[i]/(e35600[i]*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e3800f[i]);
  if e3800f[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf2[i]=othpf2[i]+1;
  /*missing value*/;    if e38329d[i]>=0 and -4<e3800f[i]<0 then othpay2[i]=e3800f[i];
  if e38329d[i]>=0 and e3800f[i]=0 then othpay2[i]=-3;
  if -4<e38329d[i]<0 then othpay2[i]=e38329d[i];
  if e38329b[i]=2 and e38116b[i]=0 then othpay2[i]=-3;
  if e38329b[i]=2 and -4<e38116b[i]<0 then othpay2[i]=e38116b[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
/* with overtime at the beginning*/
  if e38102[i]=1 or e38102[i]=3 then do;
    if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
    if e38103[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
        othpay2[i]=(e38329d[i]*e38116b[i])/e38103[i];
    if e38103[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e38329d[i]>=0 then othpay2[i]=e38329d[i]/e38103[i];
    if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e38103[i]);
    if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e38103[i]);
    if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay2[i]=e38329d[i]/(e35600[i]*e38103[i]);
    if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e38103[i]);
    if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf2[i]=othpf2[i]+1;
    if e38329b[i] in (9,14) then othpay2[i]=0;
    /*missing value*/;   if e38329d[i]>=0 and -4<e38103[i]<0 then othpay2[i]=e38103[i];
    if e38329d[i]>=0 and e38103[i]=0 then othpay2[i]=-3;
    if -4<e38329d[i]<0 then othpay2[i]=e38329d[i];
    if e38329b[i]=2 and e38116b[i]=0 then othpay2[i]=-3;
    if e38329b[i]=2 and -4<e38116b[i]<0 then othpay2[i]=e38116b[i];
    if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
    if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
  end;
end;
/*for bonuses*/
if e212004[i]=1 then do;
/* without overtime at the beginning*/
  if e3800f[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
  if e3800f[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
        othpay3[i]=(e38329d[i]*e38116b[i])/e3800f[i];
  if e3800f[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e38329d[i]>=0 then othpay3[i]=e38329d[i]/e3800f[i];
  if e3800f[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay3[i]=e38329d[i]/(e35600[i]*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e3800f[i]);
  if e3800f[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
  /*missing value*/;    if e38329d[i]>=0 and -4<e3800f[i]<0 then othpay3[i]=e3800f[i];
  if e38329d[i]>=0 and e3800f[i]=0 then othpay3[i]=-3;
  if -4<e38329d[i]<0 then othpay3[i]=e38329d[i];
  if e38329b[i]=2 and e38116b[i]=0 then othpay3[i]=-3;
  if e38329b[i]=2 and -4<e38116b[i]<0 then othpay3[i]=e38116b[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
/*with overtime at the beginning*/
  if e38102[i]=1 or e38102[i]=3 then do;
    if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
    if e38103[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
        othpay3[i]=(e38329d[i]*e38116b[i])/e38103[i];
    if e38103[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e38329d[i]>=0 then othpay3[i]=e38329d[i]/e38103[i];
    if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e38103[i]);
    if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e38103[i]);
    if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay3[i]=e38329d[i]/(e35600[i]*e38103[i]);
    if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e38103[i]);
    if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
    if e38329b[i] in (9,14) then othpay3[i]=0;
    /*missing value*/;   if e38329d[i]>=0 and -4<e38103[i]<0 then othpay3[i]=e38103[i];
    if e38329d[i]>=0 and e38103[i]=0 then othpay3[i]=-3;
    if -4<e38329d[i]<0 then othpay3[i]=e38329d[i];
    if e38329b[i]=2 and e38116b[i]=0 then othpay3[i]=-3;
    if e38329b[i]=2 and -4<e38116b[i]<0 then othpay3[i]=e38116b[i];
    if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
    if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
  end;
end;
/*for incentive pay*/
if e212005[i]=1 then do;
/*without overtime at the beginning*/
  if e3800f[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
  if e3800f[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
        othpay4[i]=(e38329d[i]*e38116b[i])/e3800f[i];
  if e3800f[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e38329d[i]>=0 then othpay4[i]=e38329d[i]/e3800f[i];
  if e3800f[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay4[i]=e38329d[i]/(e35600[i]*e3800f[i]);
  if e3800f[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e3800f[i]);
  if e3800f[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf4[i]=othpf4[i]+1;
  /*missing value*/;    if e38329d[i]>0 and -4<e3800f[i]<0 then othpay4[i]=e3800f[i];
  if e38329d[i]>=0 and e3800f[i]=0 then othpay4[i]=-3;
  if -4<e38329d[i]<0 then othpay4[i]=e38329d[i];
  if e38329b[i]=2 and e38116b[i]=0 then othpay4[i]=-3;
  if e38329b[i]=2 and -4<e38116b[i]<0 then othpay4[i]=e38116b[i];
  if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
  if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
/* with overtime at the beginning*/
```

```
    if e38102[i]=1 or e38102[i]=3 then do;
      if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
      if e38103[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
             othpay4[i]=(e38329d[i]*e38116b[i])/e38103[i];
      if e38103[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
             e38329d[i]>=0 then othpay4[i]=e38329d[i]/e38103[i];
      if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e38103[i]);
      if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e38103[i]);
      if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay4[i]=e38329d[i]/(e35600[i]*e38103[i]);
      if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e38103[i]);
      if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
             othpf4[i]=othpf4[i]+1;
      if e38329b[i] in (9,14) then othpay4[i]=0;
      /*missing value*/;    if e38329d[i]>0 and -4<e38103[i]<0 then othpay4[i]=e38103[i];
      if e38329d[i]>=0 and e38103[i]=0 then othpay4[i]=-3;
      if -4<e38329d[i]<0 then othpay4[i]=e38329d[i];
      if e38329b[i]=2 and e38116b[i]=0 then othpay4[i]=-3;
      if e38329b[i]=2 and -4<e38116b[i]<0 then othpay4[i]=e38116b[i];
      if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
      if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
    end;
  end;
  /*for others*/
  if e212006[i]=1 then do;
  /* without overtime at the beginning*/
    if e3800f[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
    if e3800f[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
             othpay5[i]=(e38329d[i]*e38116b[i])/e3800f[i];
    if e3800f[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
             e38329d[i]>=0 then othpay5[i]=e38329d[i]/e3800f[i];
    if e3800f[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e3800f[i]);
    if e3800f[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e3800f[i]);
    if e3800f[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay5[i]=e38329d[i]/(e35600[i]*e3800f[i]);
    if e3800f[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e3800f[i]);
    if e3800f[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
             othpf5[i]=othpf5[i]+1;
    /*missing value*/;   if e38329d[i]>=0 and -4<e3800f[i]<0 then othpay5[i]=e3800f[i];
    if e38329d[i]>=0 and e3800f[i]=0 then othpay5[i]=-3;
    if -4<e38329d[i]<0 then othpay5[i]=e38329d[i];
    if e38329b[i]=2 and e38116b[i]=0 then othpay5[i]=-3;
    if e38329b[i]=2 and -4<e38116b[i]<0 then othpay5[i]=e38116b[i];
    if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
    if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
  /* with overtime at the beginning*/
    if e38102[i]=1 or e38102[i]=3 then do;
      if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
      if e38103[i]>0 and e38329b[i]=2 and e38116b[i]>0 and e38329d[i]>=0 then
             othpay5[i]=(e38329d[i]*e38116b[i])/e38103[i];
      if e38103[i]>0 and e38329b[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
             e38329d[i]>=0 then othpay5[i]=e38329d[i]/e38103[i];
      if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e38103[i]);
      if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e38103[i]);
      if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
             othpay5[i]=e38329d[i]/(e35600[i]*e38103[i]);
```

```
          if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e38103[i]);
          if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
               othpf5[i]=othpf5[i]+1;
          if e38329b[i] in (9,14) then othpay5[i]=0;
          /*missing value*/;    if e38329d[i]>=0 and -4<e38103[i]<0 then othpay5[i]=e38103[i];
          if e38329d[i]>=0 and e38103[i]=0 then othpay5[i]=-3;
          if -4<e38329d[i]<0 then othpay5[i]=e38329d[i];
          if e38329b[i]=2 and e38116b[i]=0 then othpay5[i]=-3;
          if e38329b[i]=2 and -4<e38116b[i]<0 then othpay5[i]=e38116b[i];
          if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
          if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
        end;
      end;
    end;

/*** case v. more then one compensation at the beginning, same no. of hours. ***/
    if e38330[i]=1 and e20700[i]=1 then do;
    /* for tips*/   /*with overtime at the beginning*/
      if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;
      /*report hourly wage at the beginning*/
        if e23901[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
        if e23901[i]>0 and e384071[i]=2 and e34430[i]>0 and e384161[i]>=0 then
               othpay1[i]=(e384161[i]*e34430[i])/e23901[i];
        if e23901[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
               e384161[i]>=0 then othpay1[i]=e384161[i]/e23901[i];
        if e23901[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
        if e23901[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
               othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
        if e23901[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
               othpf1[i]=othpf1[i]+1;
        /*missing value*/;    if e384161[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
        if e384161[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
        if -4<e384161[i]<0 then othpay1[i]=e384161[i];
      /*report non-hourly wage*/
        if e34428[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
        if e34428[i]>0 and e384071[i]=2 and e34430[i]>0 and e384161[i]>=0 then
               othpay1[i]=(e384161[i]*e34430[i])/e34428[i];
        if e34428[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
               e384161[i]>=0 then othpay1[i]=e384161[i]/e34428[i];
        if e34428[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34428[i]);
        if e34428[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34428[i]);
        if e34428[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
               othpay1[i]=e384161[i]/(e35600[i]*e34428[i]);
        if e34428[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34428[i]);
        if e34428[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
               othpf1[i]=othpf1[i]+1;
        if e384071[i] in (9,14) then othpay1[i]=0;
        /*missing value*/;    if e384161[i]>=0 and -4<e34428[i]<0 then othpay1[i]=e34428[i];
        if e384161[i]>=0 and e34428[i]=0 then othpay1[i]=-3;
        if -4<e384161[i]<0 then othpay1[i]=e384161[i];
        if e384071[i]=2 and e34430[i] le 0 then othpay1[i]=-3;
        if e384071[i]=2 and -4<e34430[i]<0 thne othpay1[i]=e34430[i];
        if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
        if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
```

```
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/*report hourly wage at the beginning*/
  if e23901[i]>0 and e384071[i]=1  and e384161[i]>=0 then othpay1[i]=e384161[i];
  if e23901[i]>0 and e384071[i]=2  and e34402b[i]>0 and e384161[i]>=0 then
         othpay1[i]=(e384161[i]*e34402b[i])/e23901[i];
   if e23901[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and
         e384161[i]>=0 then othpay1[i]=e384161[i]/e23901[i];
  if e23901[i]>0 and e384071[i]=4  and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
  if e23901[i]>0 and e384071[i]=5  and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384071[i]=6  and e384161[i]>=0 and e35600[i]>0 then
         othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384071[i]=8  and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
         othpf1[i]=othpf1[i]+1;
  /*missing value*/;    if e384161[i]>=0 and -4<e23901[i]<0 then othpay1[i]=e23901[i];
  if e384161[i]>=0 and e23901[i]=0 then othpay1[i]=-3;
  if -4<e384161[i]<0 then othpay1[i]=e384161[i];
/*report non-hourly wage*/
  if e34402[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
  if e34402[i]>0 and e384071[i]=2 and e34402b[i]>0 and e384161[i]>=0 then
         othpay1[i]=(e384161[i]*e34402b[i])/e34402[i];
   if e34402[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
         e384161[i]>=0 then othpay1[i]=e384161[i]/e34402[i];
  if e34402[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34402[i]);
  if e34402[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
         othpay1[i]=e384161[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
         othpf1[i]=othpf1[i]+1;
  if e384071[i] in (9,14) then othpay1[i]=0;
  /*missing value*/;
  if e384161[i]>=0 and -4<e34402[i]<0 then othpay1[i]=e34402[i];
  if e384161[i]>=0 and e34402[i]=0 then othpay1[i]=-3;
  if e384071[i]=2 and e34402b[i] le 0 then othpay1[i]=-3;
  if e384071[i]=2 and -4<e34402b[i]<0 thne othpay1[i]=e34402b[i];
   if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
  if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
end;
/*for commissions*/   /*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;
 /* report hourly wage at the beginning*/
  if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e23901[i]>0 and e384072[i]=2 and e34430[i]>0 and e384162[i]>=0 then
         othpay2[i]=(e384162[i]*e34430[i])/e23901[i];
   if e23901[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
         e384162[i]>=0 then othpay2[i]=e384162[i]/e23901[i];
  if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
  if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
         othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
         othpf2[i]=othpf2[i]+1;
  /*missing value*/;   if e384162[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
  if e384162[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
/* report non-hourly wage*/
  if e34428[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e34428[i]>0 and e384072[i]=2 and e34430[i]>0 and e384162[i]>=0 then
        othpay2[i]=(e384162[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384162[i]>=0 then othpay2[i]=e384162[i]/e34428[i];
  if e34428[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34428[i]);
  if e34428[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
        othpay2[i]=e384162[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34428[i]);
  if e34428[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf2[i]=othpf2[i]+1;
  if e384072[i] in (9,14) then othpay2[i]=0;
  /*missing value*/;   if e384162[i]>=0 and -4<e34428[i]<0 then othpay2[i]=e34428[i];
  if e384162[i]>=0 and e34428[i]=0 then othpay2[i]=-3;
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
  if e384072[i]=2 and e34430[i] le 0 then othpay2[i]=-3;
  if e384072[i]=2 and -4<e34430[i]<0 thne othpay2[i]=e34430[i];
  if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e23901[i]>0 and e384072[i]=2 and e34402b[i]>0 and e384162[i]>=0 then
        othpay2[i]=(e384162[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384162[i]>=0 then othpay2[i]=e384162[i]/e23901[i];
  if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
  if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
        othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf2[i]=othpf2[i]+1;
  /*missing value*/;   if e384162[i]>=0 and -4<e23901[i]<0 then othpay2[i]=e23901[i];
  if e384162[i]>=0 and e23901[i]=0 then othpay2[i]=-3;
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
/* report non-hourly wage*/
  if e34402[i]>0 and e384072[i]=1  and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e34402[i]>0 and e384072[i]=2  and e34402b[i]>0 and e384162[i]>=0 then
        othpay2[i]=(e384162[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28)  and
        e384162[i]>=0 then othpay2[i]=e384162[i]/e34402[i];
  if e34402[i]>0 and e384072[i]=4  and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34402[i]);
  if e34402[i]>0 and e384072[i]=5  and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e384072[i]=6  and e384162[i]>=0 and e35600[i]>0 then
        othpay2[i]=e384162[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e384072[i]=8  and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf2[i]=othpf2[i]+1;
  if e384072[i] in (9,14) then othpay2[i]=0;
  /*missing value*/;   if e384162[i]>=0 and -4<e34402[i]<0 then othpay2[i]=e34402[i];
  if e384162[i]>=0 and e34402[i]=0 then othpay2[i]=-3;
  if e384072[i]=2 and e34402b[i] le 0 then othpay2[i]=-3;
  if e384072[i]=2 and -4<e34402b[i]<0 thne othpay2[i]=e34402b[i];
  if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
end;
/*for bonuses*/  /*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e23901[i]>0 and e384073[i]=2 and e34430[i]>0 and e384163[i]>=0 then
        othpay3[i]=(e384163[i]*e34430[i])/e23901[i];
  if e23901[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384163[i]>=0 then othpay3[i]=e384163[i]/e23901[i];
  if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
  if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
        othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
  /*missing value*/;   if e384163[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
  if e384163[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay3[i]=e384163[i];
/*report non-hourly wage*/
  if e34428[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e34428[i]>0 and e384073[i]=2 and e34430[i]>0 and e384163[i]>=0 then
        othpay3[i]=(e384163[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384163[i]>=0 then othpay3[i]=e384163[i]/e34428[i];
  if e34428[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34428[i]);
  if e34428[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
        othpay3[i]=e384163[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34428[i]);
  if e34428[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
  if e384073[i] in (9,14) then othpay3[i]=0;
  /*missing value*/;   if e384163[i]>=0 and -4<e34428[i]<0 then othpay3[i]=e34428[i];
  if e384163[i]>=0 and e34428[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay3[i]=e384163[i];
  if e384073[i]=2 and e34430[i] le 0 then othpay3[i]=-3;
  if e384073[i]=2 and -4<e34430[i]<0 thne othpay3[i]=e34430[i];
  if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e23901[i]>0 and e384073[i]=2 and e34402b[i]>0 and e384163[i]>=0 then
        othpay3[i]=(e384163[i]*e34402b[i])/e23901[i];
  if e23901[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384163[i]>=0 then othpay3[i]=e384163[i]/e23901[i];
  if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
  if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
        othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
  /*missing value*/;    if e384163[i]>=0 and -4<e23901[i]<0 then othpay3[i]=e23901[i];
  if e384163[i]>=0 and e23901[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay3[i]=e384163[i];
/*report non-hourly wage*/
  if e34402[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e34402[i]>0 and e384073[i]=2 and e34402b[i]>0 and e384163[i]>=0 then
        othpay3[i]=(e384163[i]*e34402b[i])/e34402[i];
  if e34402[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384163[i]>=0 then othpay3[i]=e384163[i]/e34402[i];
  if e34402[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34402[i]);
  if e34402[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34402[i]);
  if e34402[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
        othpay3[i]=e384163[i]/(e35600[i]*e34402[i]);
  if e34402[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34402[i]);
  if e34402[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf3[i]=othpf3[i]+1;
  if e384073[i] in (9,14) then othpay3[i]=0;
  /*missing value*/;    if e384163[i]>=0 and -4<e34402[i]<0 then othpay3[i]=e34402[i];
  if e384163[i]>=0 and e34402[i]=0 then othpay3[i]=-3;
  if e384073[i]=2 and e34402b[i] le 0 then othpay3[i]=-3;
  if e384073[i]=2 and -4<e34402b[i]<0 thne othpay3[i]=e34402b[i];
  if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;
/*for incentive pay*/   /*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;
/*report hourly wage at the beginning*/
  if e23901[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
  if e23901[i]>0 and e384074[i]=2 and e34430[i]>0 and e384164[i]>=0 then
        othpay4[i]=(e384164[i]*e34430[i])/e23901[i];
  if e23901[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384164[i]>=0 then othpay4[i]=e384164[i]/e23901[i];
  if e23901[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
  if e23901[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
        othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf4[i]=othpf4[i]+1;
  /*missing value*/;    if e384164[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
  if e384164[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
  if -4<e384164[i]<0 then othpay4[i]=e384164[i];
/*report non-hourly wage*/
  if e34428[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
  if e34428[i]>0 and e384074[i]=2 and e34430[i]>0 and e384164[i]>=0 then
        othpay4[i]=(e384164[i]*e34430[i])/e34428[i];
  if e34428[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384164[i]>=0 then othpay4[i]=e384164[i]/e34428[i];
  if e34428[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34428[i]);
```

```
   if e34428[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34428[i]);
   if e34428[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
            othpay4[i]=e384164[i]/(e35600[i]*e34428[i]);
   if e34428[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34428[i]);
   if e34428[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf4[i]=othpf4[i]+1;
   if e384074[i] in (9,14) then othpay4[i]=0;
   /*missing value*/;    if e384164[i]>=0 and -4<e34428[i]<0 then othpay4[i]=e34428[i];
   if e384164[i]>=0 and e34428[i]=0 then othpay4[i]=-3;
   if -4<e384164[i]<0 then othpay4[i]=e384164[i];
   if e384074[i]=2 and e34430[i] le 0 then othpay4[i]=-3;
   if e384074[i]=2 and -4<e34430[i]<0 thne othpay4[i]=e34430[i];
   if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
   if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/*report hourly wage at the beginning*/
   if e23901[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
   if e23901[i]>0 and e384074[i]=2 and e34402b[i]>0 and e384164[i]>=0 then
            othpay4[i]=(e384164[i]*e34402b[i])/e23901[i];
   if e23901[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
            e384164[i]>=0 then othpay4[i]=e384164[i]/e23901[i];
   if e23901[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
   if e23901[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
   if e23901[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
            othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
   if e23901[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
   if e23901[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf4[i]=othpf4[i]+1;
   /*missing value*/;   if e384164[i]>=0 and -4<e23901[i]<0 then othpay4[i]=e23901[i];
   if e384164[i]>=0 and e23901[i]=0 then othpay4[i]=-3;
   if -4<e384164[i]<0 then othpay4[i]=e384164[i];
/*report non-hourly wage*/
   if e34402[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
   if e34402[i]>0 and e384074[i]=2 and e34402b[i]>0 and e384164[i]>=0 then
            othpay4[i]=(e384164[i]*e34402b[i])/e34402[i];
   if e34402[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
            e384164[i]>=0 then othpay4[i]=e384164[i]/e34402[i];
   if e34402[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34402[i]);
   if e34402[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34402[i]);
   if e34402[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
            othpay4[i]=e384164[i]/(e35600[i]*e34402[i]);
   if e34402[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34402[i]);
   if e34402[i]>0 and e384074[i] in (9,7,12,13,114,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf4[i]=othpf4[i]+1;
   if e384074[i] in (9,14) then othpay4[i]=0;
   /*missing value*/;    if e384164[i]>=0 and -4<e34402[i]<0 then othpay4[i]=e34402[i];
   if e384164[i]>=0 and e34402[i]=0 then othpay4[i]=-3;
   if e384074[i]=2 and e34402b[i] le 0 then othpay4[i]=-3;
   if e384074[i]=2 and -4<e34402b[i]<0 thne othpay4[i]=e34402b[i];
   if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
   if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
end;
/*for others*/   /*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;
```

```
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
  if e23901[i]>0 and e384075[i]=2 and e34430[i]>0 and e384165[i]>=0 then
        othpay5[i]=(e384165[i]*e34430[i])/e23901[i];
   if e23901[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384165[i]>=0 then othpay5[i]=e384165[i]/e23901[i];
  if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);
  if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
        othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf5[i]=othpf5[i]+1;
  /*missing value*/;    if e384165[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
  if e384165[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
  if -4<e384165[i]<0 then othpay5[i]=e384165[i];
/*report non-hourly wage*/
  if e34428[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
  if e34428[i]>0 and e384075[i]=2 and e34430[i]>0 and e384165[i]>=0 then
        othpay5[i]=(e384165[i]*e34430[i])/e34428[i];
   if e34428[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384165[i]>=0 then othpay5[i]=e384165[i]/e34428[i];
  if e34428[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34428[i]);
  if e34428[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34428[i]);
  if e34428[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
        othpay5[i]=e384165[i]/(e35600[i]*e34428[i]);
  if e34428[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34428[i]);
  if e34428[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf5[i]=othpf5[i]+1;
  if e384075[i] in (9,14) then othpay5[i]=0;
  /*missing value*/;    if e384165[i]>=0 and -4<e34428[i]<0 then othpay5[i]=e34428[i];
  if e384165[i]>=0 and e34428[i]=0 then othpay5[i]=-3;
  if -4<e384165[i]<0 then othpay5[i]=e384165[i];
  if e384075[i]=2 and e34430[i] le 0 then othpay5[i]=-3;
  if e384075[i]=2 and -4<e34430[i]<0 thne othpay5[i]=e34430[i];
  if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
  if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
end;
/*without overtime at the beginning*/
if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
  if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
  if e23901[i]>0 and e384075[i]=2 and e34402b[i]>0 and e384165[i]>=0 then
        othpay5[i]=(e384165[i]*e34402b[i])/e23901[i];
   if e23901[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
        e384165[i]>=0 then othpay5[i]=e384165[i]/e23901[i];
  if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);
  if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
  if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
        othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
  if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
  if e23901[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
        othpf5[i]=othpf5[i]+1;
  /*missing value*/;    if e384165[i]>=0 and -4<e23901[i]<0 then othpay5[i]=e23901[i];
  if e384165[i]>=0 and e23901[i]=0 then othpay5[i]=-3;
  if -4<e384165[i]<0 then othpay5[i]=e384165[i];
```

```
      /*report non-hourly wage*/
        if e34402[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
        if e34402[i]>0 and e384075[i]=2 and e34402b[i]>0 and e384165[i]>=0 then
                othpay5[i]=(e384165[i]*e34402b[i])/e34402[i];
        if e34402[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and
                e384165[i]>=0 then othpay5[i]=e384165[i]/e34402[i];
        if e34402[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34402[i]);
        if e34402[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34402[i]);
        if e34402[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
                othpay5[i]=e384165[i]/(e35600[i]*e34402[i]);
        if e34402[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34402[i]);
        if e34402[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                othpf5[i]=othpf5[i]+1;
        if e384075[i] in (9,14) then othpay5[i]=0;
        /*missing value*/;    if e384165[i]>=0 and -4<e34402[i]<0 then othpay5[i]=e34402[i];
        if e384165[i]>=0 and e34402[i]=0 then othpay5[i]=-3;
        if e384075[i]=2 and e34402b[i] le 0 then othpay5[i]=-3;
        if e384075[i]=2 and -4<e34402b[i]<0 thne othpay5[i]=e34402b[i];
        if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
        if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
      end;
    end;

/*** case vi. more then one compensation at the beginning, diff no. of hours. ***/
    if e38330[i]=1 and e20700[i]=1 and (e38102[i]=1 or e38102[i]=3) then do;
    /* for tips*/
      if e38103[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
      if e38103[i]>0 and e384071[i]=2 and e38116b[i]>0 and e384161[i]>=0 then
                othpay1[i]=(e384161[i]*e38116b[i])/e38103[i];
      if e38103[i]>0 and e384071[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384161[i]>=0
                then othpay1[i]=e384161[i]/e38103[i];
      if e38103[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e38103[i]);
      if e38103[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e38103[i]);
      if e38103[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
                othpay1[i]=e384161[i]/(e35600[i]*e38103[i]);
      if e38103[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e38103[i]);
      if e38103[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
                othpf1[i]=othpf1[i]+1;
      if e384071[i] in (9,14) then othpay1[i]=0;
      /*missing value*/;    if e384161[i]>=0 and -4<e38103[i]<0 then othpay1[i]=e38103[i];
      if e384161[i]>=0 and e38103[i]=0 then othpay1[i]=-3;
      if -4<e384161[i]<0 then othpay1[i]=e384161[i];
      if e384071[i]=2 and e38116b[i]=0 then othpay1[i]=-3;
      if e384071[i]=2 and -4<e38116b[i]<0 then othpay1[i]=e38116b[i];
      if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
      if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
    /*for commissions*/
      if e38103[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
      if e38103[i]>0 and e384072[i]=2 and e38116b[i]>0 and e384162[i]>=0 then
                othpay2[i]=(e384162[i]*e38116b[i])/e38103[i];
      if e38103[i]>0 and e384072[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384162[i]>=0
                then othpay2[i]=e384162[i]/e38103[i];
      if e38103[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e38103[i]);
      if e38103[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e38103[i]);
      if e38103[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
                othpay2[i]=e384162[i]/(e35600[i]*e38103[i]);
```

```
  if e38103[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e38103[i]);
  if e38103[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf2[i]=othpf2[i]+1;
  if e384072[i] in (9,14) then othpay2[i]=0;
  /*missing value*/;    if e384162[i]>=0 and -4<e38103[i]<0 then othpay2[i]=e38103[i];
  if e384162[i]>=0 and e38103[i]=0 then othpay2[i]=-3;
  if -4<e384162[i]<0 then othpay2[i]=e384162[i];
  if e384072[i]=2 and e38116b[i]=0 then othpay2[i]=-3;
  if e384072[i]=2 and -4<e38116b[i]<0 then othpay2[i]=e38116b[i];
  if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
  if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
/*for bonuses*/
  if e38103[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
  if e38103[i]>0 and e384073[i]=2 and e38116b[i]>0 and e384163[i]>=0 then
          othpay3[i]=(e384163[i]*e38116b[i])/e38103[i];
  if e38103[i]>0 and e384073[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384163[i]>=0
          then othpay3[i]=e384163[i]/e38103[i];
  if e38103[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e38103[i]);
  if e38103[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e38103[i]);
  if e38103[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
          othpay3[i]=e384163[i]/(e35600[i]*e38103[i]);
  if e38103[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e38103[i]);
  if e38103[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf3[i]=othpf3[i]+1;
  if e384073[i] in (9,14) then othpay3[i]=0;
  /*missing value*/;  if e384163[i]>=0 and -4<e38103[i]<0 then othpay3[i]=e38103[i];
  if e384163[i]>=0 and e38103[i]=0 then othpay3[i]=-3;
  if -4<e384163[i]<0 then othpay3[i]=e384163[i];
  if e384073[i]=2 and e38116b[i]=0 then othpay3[i]=-3;
  if e384073[i]=2 and -4<e38116b[i]<0 then othpay3[i]=e38116b[i];
  if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
  if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
/*for incentive pay*/
  if e38103[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
  if e38103[i]>0 and e384074[i]=2 and e38116b[i]>0 and e384164[i]>=0 then
          othpay4[i]=(e384164[i]*e38116b[i])/e38103[i];
  if e38103[i]>0 and e384074[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384164[i]>=0
          then othpay4[i]=e384164[i]/e38103[i];
  if e38103[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e38103[i]);
  if e38103[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e38103[i]);
  if e38103[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
          othpay4[i]=e384164[i]/(e35600[i]*e38103[i]);
  if e38103[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e38103[i]);
  if e38103[i]>0 and e384074[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
          othpf4[i]=othpf4[i]+1;
  if e384074[i] in (9,14) then othpay4[i]=0;
  /*missing value*/;    if e384164[i]>=0 and -4<e38103[i]<0 then othpay4[i]=e38103[i];
  if e384164[i]>=0 and e38103[i]=0 then othpay4[i]=-3;
  if -4<e384164[i]<0 then othpay4[i]=e384164[i];
  if e384074[i]=2 and e38116b[i]=0 then othpay4[i]=-3;
  if e384074[i]=2 and -4<e38116b[i]<0 then othpay4[i]=e38116b[i];
  if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
  if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
/*for others*/
  if e38103[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
```

```
      if e38103[i]>0 and e384075[i]=2 and e38116b[i]>0 and e384165[i]>=0 then
            othpay5[i]=(e384165[i]*e38116b[i])/e38103[i];
      if e38103[i]>0 and e384075[i] in (3, 7, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) and e384165[i]>=0
            then othpay5[i]=e384165[i]/e38103[i];
      if e38103[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e38103[i]);
      if e38103[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e38103[i]);
      if e38103[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
            othpay5[i]=e384165[i]/(e35600[i]*e38103[i]);
      if e38103[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e38103[i]);
      if e38103[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then
            othpf5[i]=othpf5[i]+1;
      if e384075[i] in (9,14) then othpay5[i]=0;
      /*missing value*/;    if e384165[i]>=0 and -4<e38103[i]<0 then othpay5[i]=e38103[i];
      if e384165[i]>=0 and e38103[i]=0 then othpay5[i]=-3;
      if -4<e384165[i]<0 then othpay5[i]=e384165[i];
      if e384075[i]=2 and e38116b[i]=0 then othpay5[i]=-3;
      if e384075[i]=2 and -4<e38116b[i]<0 then othpay5[i]=e38116b[i];
      if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
      if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
    end;
  end;
end;

/*************************************overall end hourly compensation
      **************************************/
do i=1 to 9;
  if e37901b[i]=1 or e59900[i]=1 then do;
    hrcomp[i]=0;
    if hrwg[i] ge 0 then hrcomp[i]=hrcomp[i]+hrwg[i];
    if ot[i] ge 0 then hrcomp[i]=hrcomp[i]+ot[i];
    if othpay1[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay1[i];
    if othpay2[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay2[i];
    if othpay3[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay3[i];
    if othpay4[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay4[i];
    if othpay5[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay5[i];
    if -4<hrwg[i]<0 or -4<ot[i]<0 or -4<othpay1[i]<0 or -4<othpay2[i]<0 or -4<othpay3[i]<0 or -4<othpay4[i]<0 or -
        4<othpay5[i]<0 then hrcomp[i]=-3;
    if hrwg[i]=-4 then hrcomp[i]=-4;
  end;
end;

/* fix the hourly compensation because the respondents may understand the overtime question differently. 1) if
      overtime pay is less than or equal to the hourly wage, we assume the resp. consider the overtime as
      something above the hourly wage. so the hourly comp. should include both hourly wage and overtime pay.
      2) if overtime pay is greater than the hourly wage, we assume the respondents consider the overtime pay as a
      pay including hourly wage. so the hourly comp. only include the overtime pay without the hourly wage. */
do i=1 to 9; if hrwg[i] gt 0 and ot[i] gt hrwg[i] and hrcomp[i] gt 0 then hrcomp[i]=hrcomp[i]-hrwg[i]; end;

/************************************Section 3 :  Job lasts 13 weeks or
      less**************************************/
jleng01=-4;      jleng02=-4;      jleng03=-4;      jleng04=-4;      jleng05=-4;
jleng06=-4;      jleng07=-4;      jleng08=-4;      jleng09=-4;

array jleng jleng01 jleng02 jleng03 jleng04 jleng05 jleng06 jleng07 jleng08 jleng09 jleng10;
do i=1 to 9;
  if e37701[i]=0 then jleng[i]=0;
```

```
   if e58201[i]=0 then jleng[i]=0;
   if e37901b[i]>-1 or e59900[i]>-1 or e58401[i]>-1 then do;
      if e37901b[i]=1 or e59900[i]=1 or e58401[i]=0 then jleng[i]=1;
      else jleng[i]=0;
   end;
end;

array jleng jleng01 jleng02 jleng03 jleng04 jleng05 jleng06 jleng07 jleng08 jleng09;
array hrwg hrwg01 hrwg02 hrwg03 hrwg04 hrwg05 hrwg06  hrwg07 hrwg08 hrwg09 ;
array hrcomp hrcomp01 hrcomp02 hrcomp03 hrcomp04 hrcomp05 hrcomp06  hrcomp07 hrcomp08 hrcomp09;
do i=1 to 9;
   if jleng[i]>-4 and hrwg[i]=-4 then hrwg[i]=-3;
   if jleng[i]>-4 and hrcomp[i]=-4 then hrcomp[i]=-3;
end;

array hrwgr hrwgr01 hrwgr02 hrwgr03 hrwgr04 hrwgr05 hrwgr06 hrwgr07 hrwgr08 hrwgr09;
hrwgr01=-4;      hrwgr02=-4;      hrwgr03=-4;      hrwgr04=-4;      hrwgr05=-4;
hrwgr06=-4;      hrwgr07=-4;      hrwgr08=-4;      hrwgr09=-4;
do i=1 to 9;
   if hrwg[i] ne -4 then do; hrwgr[i]=round(hrwg[i],1); end;
    if e200a=-5 then hrwgr[i]=-5;
    if e200a=-5 then jleng[i]=-5;
end;

array hrcomr hrcomr01 hrcomr02 hrcomr03 hrcomr04 hrcomr05 hrcomr06 hrcomr07 hrcomr08 hrcomr09;
hrcomr01=-4;     hrcomr02=-4;     hrcomr03=-4;     hrcomr04=-4;     hrcomr05=-4;
hrcomr06=-4;     hrcomr07=-4;     hrcomr08=-4;     hrcomr09=-4;
do i=1 to 9;
    if hrcomp[i] ne -4 then do; hrcomr[i]=round(hrcomp[i],1); end;
    if e200a=-5 then hrcomr[i]=-5;
end;

endsas;
```