```
ONE TRUE FILE FOR BOTH MAKE

Makefiles are the standard way to build and compile GNU programs on Linux & derivatives
Many Makefiles are very complex, so we'll try to introduce the basic parts of a
Makefile.

Makefiles are also useful in creating a development and testing setup system that can
automate a lot of the drudgery involved in testing and debugging code. For example,
suppose you're writing a program that uses a library of routines. Suppose further that the
routines in the library are changing frequently. Without a Makefile, you'll have to
remember which files you've changed and which files depend on them. With a Makefile, you
can just type "make" and be confident that all the right files will be recompiled.

Makefiles are also useful for installing software. Many programs come with Makefiles that
can be used to install the software in a standard way. Using a Makefile to install your
software can make it easier for other people to build your software.

The simplest Makefile you can write is a file which lists only the targets of the files in
the program. If the program is called "prog" and is made from "main.o" and "subs.o", which
are compiled from "main.c" and "subs.c" and "subs.h", then the following Makefile will
compile the program:

    prog: main.o subs.o
        cc -o prog main.o subs.o

    main.o: main.c subs.h
        cc -c main.c

    subs.o: subs.c subs.h
        cc -c subs.c

    clean:
        rm -f prog main.o subs.o

This Makefile contains four targets, "prog", "main.o", "subs.o", and "clean". The first
target is the one that will be built when you just type "make". It depends on the two ".o"
files, and tells how to link them together. The second target tells how to compile "main.o".
It depends on "main.c" and "subs.h", and tells how to compile it. The third target tells how
to compile "subs.o". The last target, "clean", doesn't build anything. It just removes the
program and the ".o" files, so that everything can be rebuilt from scratch.

Each target can be followed by a list of commands to execute. Before each command is
executed, the variables and functions are expanded, and then the command is passed to a
subshell for execution. Each command must be preceded by a tab. This is a requirement of
the Makefile syntax; it's not optional.

Variables are like macros. They're defined by writing

    VARIABLE = value

Later, the value can be substituted by writing $(VARIABLE) or ${VARIABLE}. Here's a version
of the Makefile which uses variables:

    OBJS = main.o subs.o
    CC = cc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

Here's a version that will compile the program using gcc instead of cc, and that will
install it in /usr/local/bin:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/local/bin/prog

Here's a version that will install the program in /usr/bin instead:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists, and will also install the program in /usr/local/bin if the directory
/usr/local/bin exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi
        if [ -d /usr/local/bin ]; then cp prog /usr/local/bin/prog; fi

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists, and will also install the program in /usr/local/bin if the directory
/usr/local/bin exists, and will also install the program in /usr/local/bin/X11 if the
directory /usr/local/bin/X11 exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi
        if [ -d /usr/local/bin ]; then cp prog /usr/local/bin/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists, and will also install the program in /usr/local/bin if the directory
/usr/local/bin exists, and will also install the program in /usr/local/bin/X11 if the
directory /usr/local/bin/X11 exists, and will also install the program in /usr/local/bin/X11
if the directory /usr/local/bin/X11 exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi
        if [ -d /usr/local/bin ]; then cp prog /usr/local/bin/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists, and will also install the program in /usr/local/bin if the directory
/usr/local/bin exists, and will also install the program in /usr/local/bin/X11 if the
directory /usr/local/bin/X11 exists, and will also install the program in /usr/local/bin/X11
if the directory /usr/local/bin/X11 exists, and will also install the program in
/usr/local/bin/X11 if the directory /usr/local/bin/X11 exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi
        if [ -d /usr/local/bin ]; then cp prog /usr/local/bin/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi

Here's a version that will install the program in /usr/bin and also install the manual page
in /usr/man/man1, and will also install the program in /usr/bin/X11 if the directory
/usr/bin/X11 exists, and will also install the program in /usr/local/bin if the directory
/usr/local/bin exists, and will also install the program in /usr/local/bin/X11 if the
directory /usr/local/bin/X11 exists, and will also install the program in /usr/local/bin/X11
if the directory /usr/local/bin/X11 exists, and will also install the program in
/usr/local/bin/X11 if the directory /usr/local/bin/X11 exists, and will also install the
program in /usr/local/bin/X11 if the directory /usr/local/bin/X11 exists:

    OBJS = main.o subs.o
    CC = gcc
    prog: $(OBJS)
        $(CC) -o prog $(OBJS)

    main.o: main.c subs.h
        $(CC) -c main.c

    subs.o: subs.c subs.h
        $(CC) -c subs.c

    clean:
        rm -f prog $(OBJS)

    install: prog
        cp prog /usr/bin/prog
        cp prog.1 /usr/man/man1/prog.1
        if [ -d /usr/bin/X11 ]; then cp prog /usr/bin/X11/prog; fi
        if [ -d /usr/local/bin ]; then cp prog /usr/local/bin/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi
        if [ -d /usr/local/bin/X11 ]; then cp prog /usr/local/bin/X11/prog; fi

This is getting a bit ridiculous. Suppose we want to add another directory to the list of
directories that the program is installed in? We'd have to add another line to the
Makefile, and we'd have to add another "if" statement to the "install" target. There's a
better way to do this. Here's a version that will install the program in all of the above
directories, and will also install the program in /usr/local/bin/X11 if the directory
/usr/local/bin/X11 
```

```
1 = 'I YES'
2 = 'I NO'
/

VALUE INAFFLIM
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE REVHERE
-1 = 'I INAPPLICABLE'
1 = 'I KEY'
2 = 'I NOT KEY'
/

VALUE LFTSFIP
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'I SOME DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'I UNABLE TO DO'
/

VALUE MARRY118
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I MARRIED'
2 = 'I WIDOWED'
3 = 'I DIVORCED'
4 = 'A SEPARATED'
5 = 'I NEVER MARRIED'
6 = 'UNDER 14 INAPPLICABLE'
7 = 'I MARRIED IN BOOKS'
8 = 'I WIDOWED IN BOOKS'
9 = 'I DIVORCED IN BOOKS'
10 = 'IS SEPARATED IN BOOKS'
/

VALUE MCA113F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE MCA131R
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE MCA6E11F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE MCA6I11F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'A LOT OF DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'I UNABLE TO DO'
/

VALUE MARLIP1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I VERY HIGH'
2 = 'I HIGH'
3 = 'I LOW'
4 = 'I VERY LOW'
/

VALUE MDA1F
-1 = 'I INAPPLICABLE'
1 = 'I MDA'
2 = 'I NON-MDA'
/

VALUE MARMOT2F
-9 = 'NOT ASCERTAINED'
-1 = 'I INAPPLICABLE'
1 = 530 = 'I = 500 BOMBER OF ENFLOUSES'
/

VALUE OFFRIS18
-9 = 'NOT ASCERTAINED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE OFFRAL1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE OFFMOS1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PANEL
1 = 'I PANEL 11 NO 1'
12 = '12 PANEL 12 NO 1'
/

VALUE PID
10 = 501 = '10 = 501 PERSON BOMBER'
/

VALUE PRIMIP1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PRIMIP2F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PRIMIP3F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PRIMIP4F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PRIVIP1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PROMIP1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE PROMIP2F
-1 = 'I INAPPLICABLE'
1 = 'I RESPONDENT IS NO MEMBER'
2 = 'I RESPONDENT IS A PRIEST'
/

VALUE PSTAT1F
-1 = 'I NOT FIELDED/NON-RESPONSE NP'
10 = 'I INAPPLICABLE - NOT IN SAMPLE/NON-RES DELT
11 = 'I PERSON IN HOUSEHOLD, NOT PT ACTIVE HOU
12 = 'IN HOUSEHOLD, PT HOU DUTY-DUTY-HOUSE'
13 = 'I PERSON IN HOUSEHOLD, PT HOUSE'
14 = 'IN AREA NH, PT ACTIVE DUTY-INSIDE'
21 = 'I INSTITUTIONALIZED WHOLE, HI ONLY
22 = 'I INSTITUTIONALIZED HOUSE, HI ONLY
23 = 'I LEAVES INSTITUTION W DIED - 63 ONLY
24 = 'DIED IN INST DURING NO - 63 ONLY
31 = 'I
32 = 'INSTITUTIONALIZED IN HEALTH CARE FAC'
33 = 'I INSTITUTIONALIZED NON-HEALTH CARE FAC'
34 = 'I INSTITUTIONALIZED, PT ACTIVE DUTY
35 = 'I PATIENT, PT ACTIVE DUTY'
36 = 'I INSTITUTIONALIZED (TYPE UNKNOWN)'
41 = 'I ARMED FORCES
42 = 'I PERSON WHO JOINED NO'
43 = 'I REASON DEN IN MOVED TO DEN LOCATION'
44 = 'I PERSON WHO DIED IN NO - 63 ONLY
51 = 'I NONWORK IN REF PERIOD'
61 = 'I INCHARGE PRIN TO REF PERIOD-83/1'
62 = 'I INCHARGE PRIN IN REF PERIOD'
63 = 'I LEFT U.S. PRIOR TO REF PER-83/1'
64 = 'I PT HOU, ACL PAC MOVE PRIOR REF PER-83/1'
71 = 'I INMATE, PT ACTIVE DUTY-INSIDE'
72 = 'I INMATE NON-HOU/INSIDEPT HOU, PR ONLY'
73 = 'I NONRES, NOT PT HOU, MOVED W/OUT KEY'
74 = 'I NONRES, NOT PT HOU, MOVED W/ KEY'
81 = 'I NOVEL, PT STOU, NO HOUS, DATA INPUTED'
/

VALUE POB118
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE RACE6
-1 = 'I INAPPLICABLE'
1 = 'I ASIAN-NO OTHER RACE REPORTED'
2 = 'I ASIAN-OTHER RACE(S) REPORTED'
3 = 'I ALL OTHER RACE ASSIGNMENTS'
/

VALUE RACE8
-1 = 'I INAPPLICABLE'
1 = 'I BLACK-NO OTHER RACE REPORTED'
2 = 'I BLACK-OTHER RACE(S) REPORTED'
3 = 'I ALL OTHER RACE ASSIGNMENTS'
/

VALUE RACE9MR
-1 = 'I INAPPLICABLE'
1 = 'I PERSON IS BLACK-NO OTH RACE/NOT HISPANIC'
2 = 'I PERSON IS BLACK-OTH RACE(S)/NOT HISPANIC'
3 = 'I ALL OTHER RACE/NOT HISPANIC'
4 = 'I OTHER RACE/NOT HISPANIC'
/

VALUE RACE9
-1 = 'I INAPPLICABLE'
1 = 'I WHITE-NO OTHER RACE REPORTED'
2 = 'I BLACK-NO OTHER RACE REPORTED'
3 = 'AMER INDIAN/ALASKA NATIVE - NO OTH RACE'
4 = 'ASIAN - NO OTHER RACE REPORTED'
5 = 'HAWAIIAN/PACIFIC ISLANDER-NO OTHER'
6 = 'I MULTIPLE RACES REPORTED'
/

VALUE RCHGP1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'A LOT OF DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'I UNABLE TO DO'
/

VALUE RELG11F
-1 = 'I INAPPLICABLE'
1 = 'I PROTESTANT'
2 = 'I ROMANIST'
3 = 'I HEBTH'
4 = 'I NONE'
/

VALUE REMOVE1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = '11 = 7 = 11 DAY'
/

VALUE REMOVE1P
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I JANUARY'
2 = 'I FEBRUARY'
3 = 'I MARCH'
4 = 'I APRIL'
5 = 'I MAY'
6 = 'I JUNE'
7 = 'I JULY'
8 = 'I AUGUST'
9 = 'I SEPTEMBER'
10 = 'I OCTOBER'
11 = 'I NOVEMBER'
12 = 'I DECEMBER'
/

VALUE REMOVI1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
2047 = 2057 YEAR
/

VALUE RESPHIP
-1 = 'I INAPPLICABLE'
1 = 'I YES, FIRST RESPONDENT'
2 = 'I NO, NOT FIRST RESPONDENT'
3 = 'I NO, NOT RESPONSIBLE'
/

VALUE RFMAR118
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
0 = 'SELF'
1 = 'MOTHER'
2 = 'FATHER'
3 = 'SISTER/SPTER/HALF'
4 = 'BROTHER/SPTER/HALF'
5 = 'I DAUGHTER/ADOPTED DAUGHTER'
6 = 'SON/ADOPTED SON'
7 = 'AUNT/UNCLE'
8 = 'NIECE/NEPHEW'
9 = 'MOTHER-IN-LAW'
10 = 'FATHER-IN-LAW'
11 = 'SISTER-IN-LAW'
12 = 'BROTHER-IN-LAW'
13 = 'DAUGHTER-IN-LAW'
14 = 'SON-IN-LAW'
15 = 'STEPMOTHER'
16 = 'STEPFATHER'
17 = 'STEPDAUGHTER'
18 = 'STEPSON'
19 = 'GRANDMOTHER'
20 = 'GRANDFATHER'
21 = 'AUNT'
22 = 'UNCLE'
23 = 'NIECE'
24 = 'NEPHEW'
25 = 'GRANDSON'
26 = 'GRANDSON'
27 = 'GRANDDAUGHTER'
28 = 'GRANDDAUGHTER'
29 = 'HALF PARTNER'
30 = 'MOTHER'S PARTNER'
31 = 'FATHER'S PARTNER'
32 = 'DAUGHTER OF PARTNER'
33 = 'SON OF PARTNER'
34 = 'GRANDMOTHER PARTNER'
35 = 'GRANDFATHERPARTNER'
36 = 'GRANDDAUGHTER'
37 = 'GRANDDAUGHTER'
38 = 'GRANDSON'
39 = 'GRANDSON'
40 = 'AUNT'
41 = 'AUNT'
42 = 'UNCLE'
43 = 'UNCLE'
44 = 'GREAT NEPHEW'
45 = 'GREAT NIECE'
46 = 'GREAT NIECE'
47 = 'GREAT NEPHEW'
48 = 'FOSTER BROTHER'
49 = 'FOSTER SISTER'
50 = 'I NOT RELATED'
/

VALUE RLIP11F
-1 = 'I INAPPLICABLE'
'A' = 'A'
'B' = 'B'
'C' = 'C'
'D' = 'D'
'E' = 'E'
'F' = 'F'
'G' = 'G'
'H' = 'H'
/

VALUE RNUM11F
-1 = 'I INAPPLICABLE'
11 = 99 = 11 = 99
/

VALUE SPOLIP1F
-1 = 'I INAPPLICABLE'
'63' = '63 COMPLETE WITH NO MEMBER'
'64' = '64 PERSON IN NO - 63 COMPLETEDNESS'
'65' = '65 PERSON-ALL NO MEMB CAPITUS/INDIANS'
'67' = '67 COMPLETE WITH PRIEST, OTHER'
'68' = '68 PERSON IN NO - 63 COMPLETEDNESS NO'
'71' = '71 DR COMPL NO ALIA NO MEMBERS, INDIAN'
'72' = '72 NO INSTUTE IN PRISON NO-STILL INSTITU'
'73' = '73 NO INSTUTE IN PRISON NO-STILL INSTITU'
'81' = 'I ENTIRE NO INCHARGE BEFORE 1/1/67'
'82' = '82 ENTIRE NO IN NOL BEFORE 1/1/67'
'83' = '83 ENTIRE NO LEFT NOL BEFORE 1/1/67'
'84' = '84 ENTIRE NO LEFT NOL AFTER 1/1/67'
'85' = '85 NO INELIG BEFORE 1/1/67, NOLIC REASON'
'86' = '86 NO INELIG BEFORE 1/1/67, OTHER'
'87' = '87 REASON CONEL, NO ELIG NO NOW, INELIG'
'88' = '88 UNKNOWINLABLE DURING FIELD PERIOD'
'89' = '89 NO INELIG AFT 1/1/67'
'90' = '90 PHYSICAL/MENTAL INCAPACITATED, NO PROXY'
'91' = '91 FINAL REFUSAL'
'92' = '92 FINAL NONCONTACT'
'93' = '93 UNABLE TO LOCATE'
'94' = '94 ENTIRE NO NIL OR LEFT US AFTER 1/1/67'
'95' = '95 NO INELIGIBLE AFT 1/1/67, NO REASON'
'96' = '96 NO ENGLNGED AFT 1/1/67, NO REASON'
'97' = '97 REASON COMPLETE, NO NO MEM, NON-RESPN'
'98' = '98 FINAL REFUSAL TOO FAR TO INTERVIEW'
'99' = '99 FINAL OTHER NONRESPONSE'
/

VALUE RTRLIP1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I EXCELLENT'
2 = 'I VERY GOOD'
3 = 'GOOD'
4 = 'I FAIR'
5 = 'I POOR'
/

VALUE RUCLA11F
-1 = 'I INAPPLICABLE'
1 = 'I STANDARD MC'
2 = 'SEM MC'
3 = 'STUDENT MC'
/

VALUE RUS111F
-1 = 'I INAPPLICABLE'
1 = 'I PERSON IN MC'
2 = 'I PERSON IN NC'
3 = 'I PERSON IN NC'
4 = 'I PERSON IN NC'
5 = 'I PERSON IN NC'
6 = 'I PERSON IN NC'
7 = 'I PERSON IN NC'
8 = 'I PERSON IN NC'
9 = 'I PERSON IN NC'
10 = '10 PERSONS IN NC'
11 = '11 PERSONS IN NC'
12 = '12 PERSONS IN NC'
13 = '13 PERSONS IN NC'
14 = '14 PERSONS IN NC'
15 = '15 PERSONS IN NC'
16 = '16 PERSONS IN NC'
17 = '17 PERSONS IN NC'
18 = '18 PERSONS IN NC'
/

VALUE RUSLA11F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE SEX
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I FEMALE'
/

VALUE SLFCPE1F
-9 = 'NOT ASCERTAINED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'I WORKS FOR SOMEONE ELSE'
/

VALUE SOCLAS1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE SPIG1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
15 = 994 = '201 = 994 VALID EPOCH(S)'
995 = '995 NO EPOCHS IN HOUSE'
996 = '996 ALL EPOCHS IN HOUSE/UNKNOWN'
997 = '997 LESS THAN 10 YRS OLD'
/

VALUE SPIG1P
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I ENGAGED IN THE HOUSE'
2 = 'I ENGAGED OUTSIDE THE HOUSE'
3 = 'I UNDER 16 INAPPLICABLE'
/

VALUE STRMIP1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'A LOT OF DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'A UNABLE TO DO'
/

VALUE STRMIP1P
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'A LOT OF DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'I UNABLE TO DO IT'
5 = 'I COMPLETELY UNABLE TO WALK'
/

VALUE STPMOI1F
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE TRIMAI18
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE UNALA1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE VARPSU
1.2 = '1.2'
/

VALUE VARSTR
1 = 124 = '1 = 124'
/

VALUE WGTFIL1F
0 = 0.000000
417.066738 = 57102.141401 = '417.066738 = 57102.141401 WEIGHT'
/

VALUE WGTFIL1P
0 = 0.000000
328.996146 = 43811.124297 = '328.996146 = 43811.124297 WEIGHT'
/

VALUE WLKPIP1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I NO DIFFICULTY'
2 = 'I SOME DIFFICULTY'
3 = 'A LOT OF DIFFICULTY'
4 = 'I UNABLE TO DO'
/

VALUE WLKLIM1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/

VALUE WRKLIM1F
-9 = 'NOT ASCERTAINED'
-8 = 'DK'
-7 = 'I REFUSED'
-1 = 'I INAPPLICABLE'
1 = 'I YES'
2 = 'I NO'
/
```