A Computer Program for Subset Selection in Regression Analysis (IBM Version)

by

G. P. McCabe, J. N. Arvesen, and
T. R. Wong

October 10, 1974

I. INTRODUCTION

This is intended as a computer supplement to "A Subset Selection Procedure for Regression Variables," by G. P. McCabe and J. N. Arvesen, appearing in the Journal of Statistical Computation and Simulation, 1974, Vol. 3, pp. 137-146. The original version of their computer program is described in "A Computer Program for Subset Selection in Regression Analysis," by G. P. McCabe, J. N. Arvesen and R. J. Pohl, Purdue University Statistics Department Mimeo Series No. 317, May 1973. The following writeup describes a revised program which was implemented on an IBM 360/91 at Columbia University. The original version was implemented on a CDC 6500.

The major changes to the original program are:

1. The ability to substitute the correlation matrix in lieu of the input data.

2. The ability to specify specific subsets of independent variables.

3. The ability to process different sets of data in the same run.

4. The use of the Box-Muller normal random number generator. The IBM SSP pseudo random number generator RANDU is used. Since there is a plethora of better pseudo random number generators, we leave it to the user to substitute if he desires.

II. SEQUENCE OF CARDS FOR PROCESSING

A. Data description card

B. Parameter card

C. Critical 'A' value output cards

D. Specific subsets cards (Optional)

E. Input format card

F. Data cards (Optional)

Repeat A-F as needed for different sets of data.

III. PREPARATION OF INPUT CARDS

A. Data description card

COL.	DESCRIPTION
1-8	DATADESC
9-80	Description used for labeling output

B. Parameter card

COL.	DESCRIPTION
1-5	Number of observations.
6-10	Number of independent variables (no more than 10).
11-15	Desired subset size.
16-20	Device number for input data. If data is on cards, enter 5; otherwise, enter 8.
21-25	Number of 'A' values to be calculated.

26-30	Output option: 0--Input data and correlation matrix are not printed out. 1--Input variables are printed. 2--Both input data and correlation matrix are printed.
31-39	Positive odd integer used as seed for random number generator.
41	If not zero, input data is correlation matrix.
43	If not zero, specific subsets of independent variables are read in.

C. Critical 'A' value output cards

COL.	DESCRIPTION
1-5	Number of pairs (no more than 10) of lower and upper limits for printing subsets of the sorted sequence of stochastically generated 'A' values.
6-10	Lower limit for 1st pair.
11-15	Upper limit for 1st pair.
16-20	Lower limit for 2nd pair.
21-25	Upper limit for 2nd pair.
.	
.	
.	
76-80	Lower limit for 9th pair.

Continue on 2nd card if necessary.

1-5	Upper limit for 9th pair.
6-10	Lower limit for 10th pair.
11-15	Upper limit for 10th pair.

D. Specific subsets cards (Optional)

COL.	DESCRIPTION
1-3	Number of specific subsets of independent variables (no more than 10). These are entered as <u>hexidecimal</u> (i.e. base 16) images. e.g. For 10 independent variables, if a specific subset consists of variables 1, 6 and 8, the bit pattern is ....10100001. Hence, the hexidecimal image is 000000A1.
9-18	1st subset
19-27	2nd subset
.	
.	
.	
73-80	9th subset.

Continue on 2nd card if necessary.

1-8	10th subset.

E. Input format card

COL.	DESCRIPTION
1-80	Fortran type format statement used to read input data.

F. Data cards (Optional)

1. If original observations are being used, all values of the independent variables for observation 1 are read first, followed by those for observation 2, and so forth. Each set of observations is punched beginning on a new card.

2. If the correlation matrix is being used, each row is punched beginning on a new card. Because of symmetry, only enter those values from the lower portion of the matrix.

e.g. To enter a 3x3 correlation matrix, the 3 data cards could appear as follows:

Card 1	XXX.X		
Card 2	XXX.X	XXX.X	
Card 3	XXX.X	XXX.X	XXX.X

Note: The observations or correlations, if necessary, can reside on a magnetic tape or a direct access storage device. Just be sure to denote the device number as 8.

IV. SAMPLE INPUT/OUTPUT AND PROGRAM LISTING

Input Description:

1. Card 1 describes the problem data set.

2. Card 2 tells the program:

.there are 16 observations

.there are 6 independent variables

.the subset size to be considered is 3

.the data is read from cards (i.e. device number 5)

.the number of 'A' values to be calculated is 100

.2 means that input matrix 'X' and the correlation matrix is to be printed

.the seed for generating random numbers is 123456789

.the original observations are to be read in as the input data

.the program is to find the 'best' subset of size 3

3. Card 3 asks for the printing of 3 groups of 'A' variables: 49-51, 89-91, 94-96.
4. Card 4 is the variable format card.
5. Card 5-Card 20 contain the input data. Note that the first number on these data cards is ignored.

Output Description:

1. The 6 independent variables are listed.
2. The standardized correlation matrix is printed out. Since the matrix is symmetric, only the lower half is printed out.
3. The best subset is displayed.
4. The 'A' values are listed.

V. AUXILIARY PROGRAMS

The following subroutines are from the IMSL Library:

.VSORTA
.LINV1P
.LETZ1P
.LUDECP
.LUELMP
.UERTST

For further information contact

International Mathematical and Statistical Libraries, Inc.
GNB Bldg.-Sixth Floor
7500 Bellaire
Houston, Texas 77036

```
CARD  1---> DATADESC LONGLEY DATA SET
CARD  2--->     16     6     3     5  100     2123456789
CARD  3--->      3    49    51    89    91    94    96
CARD  4---> (10X,6F10.3)
CARD  5---> 60323.     83.        234289.    2356.     1590.     107608.    1947.
CARD  6---> 61122.     88.5       259426.    2325.     1456.     108632.    1948.
CARD  7---> 60171.     88.2       258054.    3692.     1616.     109773.    1949.
CARD  8---> 61187.     89.5       284599.    3351.     1650.     110929.    1950.
CARD  9---> 63221.     96.2       328975.    2099.     3099.     112075.    1951.
CARD 10---> 63639.     98.1       346999.    1932.     3594.     113270.    1952.
CARD 11---> 64989.     99.        365385.    1870.     3547.     115094.    1953.
CARD 12---> 63761.     100.       363112.    3578.     3350.     116219.    1954.
CARD 13---> 66019.     101.2      397469.    2904.     3048.     117388.    1955.
CARD 14---> 67857.     104.6      419180.    2822.     2857.     118734.    1956.
CARD 15---> 68169.     108.4      442769.    2936.     2798.     120445.    1957.
CARD 16---> 66513.     110.8      444546.    4681.     2637.     121950.    1958.
CARD 17---> 68655.     112.6      482704.    3813.     2552.     123336.    1959.
CARD 18---> 69564.     114.2      502601.    3931.     2514.     125368.    1960.
CARD 19---> 69331.     115.7      518173.    4806.     2572.     127852.    1961.
CARD 20---> 70551.     116.9      554894.    4007.     2827.     130081.    1962.
```

```
DATA DESCRIPTION :  LONGLEY DATA SET

      NO. OF OBSERVATIONS ..............   16

      NO. OF INDEPENDENT VARIABLES
          (EXCLUDES INTERCEPT) .........    6

      SUBSET SIZE ......................    3

      NO. OF 'A' VARIABLES DESIRED .....  100
```

INPUT DATA MATRIX

```
    0.830000000D 02    0.234289000D 06    0.235600000D 04    0.159000000D 04    0.107608000D 06    0.194700000D 04
    0.885000000D 02    0.259426000D 06    0.232500000D 04    0.145600000D 04    0.108632000D 06    0.194900000D 04
    0.882000000D 02    0.258054000D 06    0.368200000D 04    0.161600000D 04    0.109773000D 06    0.194900000D 04
    0.895000000D 02    0.284599000D 06    0.335100000D 04    0.165000000D 04    0.110929000D 06    0.195000000D 04
    0.962000000D 02    0.328975000D 06    0.209900000D 04    0.309900000D 04    0.112075000D 06    0.195100000D 04
    0.981000000D 02    0.346999000D 06    0.193200000D 04    0.359400000D 04    0.113270000D 06    0.195200000D 04
    0.990000000D 02    0.365395000D 06    0.187000000D 04    0.354700000D 04    0.115094000D 06    0.195300000D 04
    0.100000000D 03    0.363112000D 06    0.357800000D 04    0.335000000D 04    0.116219000D 06    0.195400000D 04
    0.101200000D 03    0.397469000D 06    0.290400000D 04    0.304800000D 04    0.117389000D 06    0.195500000D 04
    0.104600000D 03    0.419183000D 06    0.282200000D 04    0.285700000D 04    0.118734000D 06    0.195600000D 04
    0.108400000D 03    0.442769000D 06    0.293 00000D 04    0.279800000D 04    0.120445000D 06    0.195700000D 04
    0.110800000D 03    0.444546000D 06    0.468100000D 04    0.263700000D 04    0.121950000D 06    0.195800000D 04
    0.112600000D 03    0.482704000D 06    0.381300000D 04    0.255200000D 04    0.123336000D 06    0.195900000D 04
    0.114200000D 03    0.502601000D 06    0.393100000D 04    0.251400000D 04    0.125368000D 06    0.196000000D 04
    0.115700000D 03    0.518173000D 06    0.480600000D 04    0.257200000D 04    0.127852000D 06    0.196100000D 04
    0.116900000D 03    0.554894000D 06    0.400700000D 04    0.282700000D 04    0.130081000D 06    0.196200000D 04
```

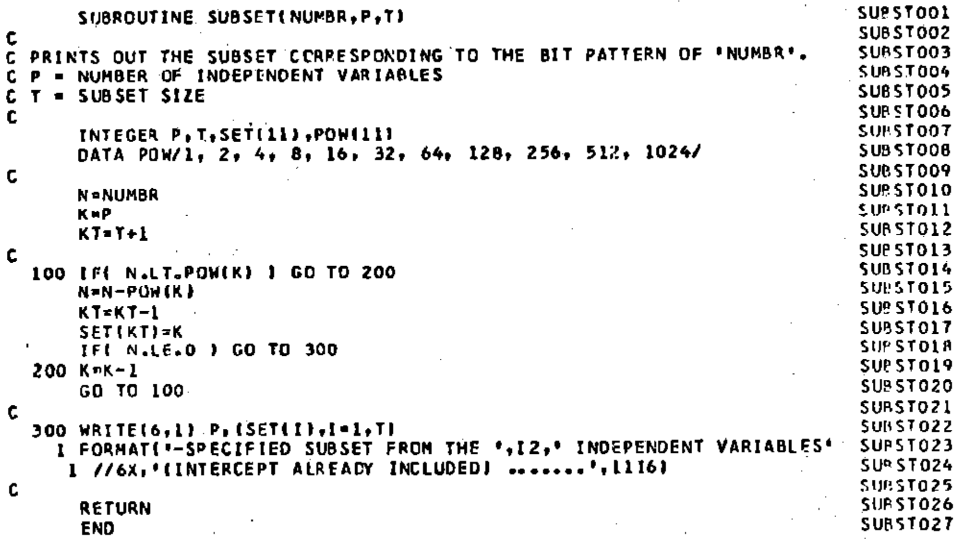
```
CORRELATION MATRIX OF STANDARDIZED DATA

  0.1600000000D 02
  0.0             0.1000000000D 01
  0.0             0.9915891780D 00  0.1000000000D 01
  0.0             0.6206333930D 00  0.6042609400D 00  0.1000000000D 01
  0.0             0.4647441080D 00  0.4464367927D 00 -0.1774206300D 00  0.1000000000D 01
  0.0             0.9791124580D 00  0.9910581080D 00  0.6865290840D 00  0.3645281890D 00  0.1000000000D 01
  0.0             0.9911491900D 00  0.9952734840D 00  0.6682566050D 00  0.4172451500D 00  0.9939246490D 00  0.1000000000D 01

OPTIMUM SUBSET SELECTED FROM THE  6 INDEPENDENT VARIABLES

    (INTERCEPT ALREADY INCLUDED) .......     1     3     4

X POINT          A VARIABLE

  49        0.1125516344D 01
  50        0.1126922761D 01
  51        0.1130968931D 01
  89        0.1438269193D 01
  90        0.1492157517D 01
  91        0.1495323862D 01
  94        0.1569926047D 01
  95        0.1648667594D 01
  96        0.1966478542D 01



NORMAL TERMINATION: NO MORE DATA SETS
```

```
C                                                                       R2SSP001
C***********************************************************************R2SSP002
C                                                                      *R2SSP003
C          THIS PROGRAM IS AN ALGORITHM FOR THE SUBSET SELECTION       *R2SSP004
C     PROCEDURE DESCRIBED IN THE PAPER                                 *R2SSP005
C     * A SUBSET SELECTION PROCEDURE FOR REGRESSION VARIABLES *        *R2SSP006
C          BY GEORGE MCCABE  AND  JAMES ARVESEN                        *R2SSP007
C                  DEPARTMENT OF STATISTICS                            *R2SSP008
C                    PURDUE UNIVERSITY                                 *R2SSP009
C                 WEST LAFAYETTE, IN 47907                             *R2SSP010
C                     FEBRUARY 1973                                    *R2SSP011
C                   MIMEO SERIES 316                                   *R2SSP012
C                                                                      *R2SSP013
C     REFERENCES ARE MADE IN THE DOCUMENTATION OF THIS PROGRAM TO      *R2SSP014
C     THE CORRESPONDING SECTIONS OF THE PAPER.                         *R2SSP015
C                                                                      *R2SSP016
C     PROGRAMER -- RICHARD POHL                                        *R2SSP017
C     DEPARTMENT OF STATISTICS                                         *R2SSP018
C     PURDUE UNIVERSITY                                                *R2SSP019
C     FEBRUARY 1973                                                    *R2SSP020
C                                                                      *R2SSP021
C***********************************************************************R2SSP022
C                                                                      *R2SSP023
C     *****INPUT INFORMATION*****                                      *R2SSP024
C                                                                      *R2SSP025
C          THE INFORMATIONAL INPUT IS FROM DEVICE 5.  THE DATA IS       R2SSP026
C     READ IN VIA DEVICE 5 OR DEVICE 8.  ALL OUTPUT IS ONTO DEVICE 6.   R2SSP027
C                                                                      *R2SSP028
C     N      IS THE NUMBER OF OBSERVATIONS.                            *R2SSP029
C     NP     IS THE NUMBER OF INDEPENDENT VARIABLES (MUST BE LESS      *R2SSP030
C            THAN OR EQUAL TO TEN).                                    *R2SSP031
C     IT     IS THE DESIRED SUBSET SIZE.                               *R2SSP032
C     IN     SPECIFIES THE TAPE FROM WHICH THE DATA IS TO BE READ      *R2SSP033
C                IF 'IN' IS NOT 8 THEN IT IS SET TO 5.                 *R2SSP034
C     M      IS THE NUMBER OF 'A' VALUES TO BE CALCULATED.             *R2SSP035
C             (CURRENTLY M MUST BE LESS THAN OR EQUA. TO 1000. IF      *R2SSP036
C             A GREATER NUMBER IS DESIRED THEN CHANGE THE              *R2SSP037
C             DIMENSION OF 'A')                                        *R2SSP038
C     IOPT   IS AN OUTPUT OPTION. IF IOPT=                             *R2SSP039
C               0 THEN THE INPUT DATA AND THE XTX MATRIX ARE NOT       *R2SSP040
C                 PRINTED OUT.                                         *R2SSP041
C               1 THEN THE INPUT VARIABLES ARE PRINTED.                *R2SSP042
C               2 THEN BOTH THE INPUT DATA AND THE XTX MATRIX ARE      *R2SSP043
C                 PRINTED.                                             *R2SSP044
C     ICOR   IF NON-ZERO, INPUT DATA IS CORRELATION MATRIX.             R2SSP045
C     IXNUMB IF NON-ZERO, USER SPECIFIES REGRESSION SUBSETS.            R2SSP046
C     FMT    IS THE FORMAT OF THE INDEPENDENT VARIABLES TO BE READ     *R2SSP047
C               IN.                                                    *R2SSP048
C     NLIST IS THE NUMBER OF BLOCKS OF THE 'A' VARIABLES TO BE         *R2SSP049
C               PRINTED. CURRENTLY NLIST MUST BE LESS THAN OR          *R2SSP050
```

```
C            EQUAL  TO TEN.
C      ILOW AND IUP    ARE THE LOWER AND UPPER LIMITS OF THE NLIST
C                      BLOCKS OF 'A' TO BE PRINTED.
C
C***********************************************************************
C
C      THE FOLLOWING SUBROUTINES ARE CALLED BY THE PROGRAM--
C
C      (A) PROGRAMS WRITTEN OR REVISED FOR THIS PROGRAM.
C      NUMB    IS A FUNCTION SUBPROGRAM WHICH RETURNS THE REFERENCE
C              TO AN R-SQUARED VALUE TO BE OBTAINED FROM SCRNA.
C      PRMUT   IS THE SUBROUTINE WHICH RETURNS THE PERMUTAIONS ON A
C              SET OF NUMBERS.
C      SCRNA   IS A SUBROUTINE WHICH CALCULATES ALL R-SQUARES FOR A
C              REGRESSION PROBLEM.
C      INDX    IS A FUNCTION WHICH RETURNS THE SYMMETRIC STORAGE
C              MODE POSITION OF AN ELEMENT IN A MATRIX.
C      SUBSET  PRINTS HEXADECIMAL IMAGE OF A SPECIFIC SUBSET OF
C              INDEPENDENT VARIABLES.
C
C      GAUSS IS A SUBROUTINE WHICH GENERATES NORMAL(0,1)
C            RANDOM NUMBERS VIA THE METHOD OF BOX-MULLER.
C      RANDU IS A SUBROUTINE WHICH GENERATES UNIFORM(0,1) NUMBERS
C            AND IS CALLED BY GAUSS.
C      **NOTE** SINCE RANDOM NUMBER GENERATORS ARE MACHINE DEPEND-
C               ENT, MODIFICATION WILL BE NEEDED TO RUN THIS PRO-
C               GRAM ON ANOTHER SYSTEM.
C               THE USER MAY WISH TO REPLACE THE CALL TO RANDU IN
C               THE GAUSS SUBROUTINE BY ANOTHER UNIFORM RANDOM
C               NUMBER GENERATOR OR THE USER MAY REPLACE GAUSS BY
C               ANOTHER NORMAL(0,1) RANDOM NUMBER GENERATOR.
C      (B) PUCC SUBPROGRAMS--
C      LINEQ1 IS A SUBROUTINE WHICH SOLVES THE SYSTEM OF LINEAR
C             EQUATIONS A*X=B FOR X,WHERE A,X,ANDB ARE ARRAYS.
C             LINEQ1 CALLS A FUNCION ARITH1.
C
C      (C) SUBPROGRAMS FROM THE IMSL LIBRARY.
C      VSORTA   IS A SUBROUTINE WHICH SORTS THE ELEMENTS OF A
C               VECTOR FROM LOW TO HIGH VALUE.
C      LINV1P   IS A SUBROUTINE WHICH CALCULATES THE INVERSE OF
C               A MATRIX IN SYMMETRIC STORAGE MODE.
C      THE FOLLOWING SUBROUTINES ARE CALLED BY LINV1P--
C         LETQ1P
C         LUDECP
C         LUELMP
C         UERTST
C
C***********************************************************************
C
C      MATRIX XTX IS STORED IN SYMMETRIC STORAGE MODE TO SAVE
C      SPACE.  THE ELEMENTS ARE ORDERED AS (A11,A21,A22,A31,A32,
C       A33,...)  THE ELEMENT I,J OF XTX IS IN THE POSITION
C       I*(I-1)/2+J WHERE I IS  LESS THAN OR EQUAL TO J.
C       SINCE MATRIX L IS LOWER TRIANGULAR IT IS ALSO STORED IN
C       SYMMETRIC STORAGE MODE.
C
C************************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
C IDUMMY FORCES PROPER ALIGNMENT IN COMMON AREA
      COMMON /BLOCK/ IDUMMY, IJK1(1023), RL1(1035), RL2(1331)
      COMMON /IWX/ IX
      REAL*8 L(78),LABEL(9)
      INTEGER XNUMBS(100)
      LOGICAL CORREL, XNUMB
      DIMENSION X(11),XTX(78),A(1000),XXTX(78)
      COMMON /SCRBLK/ JDUMMY,NP,XTX,IT,NUMBR,RSQ1,RSQ2
      DIMENSION FMT(10),ILOW(10),IUP(10)
      SQRT(D)=DSQRT(D)
C
  601 FORMAT(1H1)
12345 CONTINUE
      READ(5,499,END=54321) LABEL
  499 FORMAT(8X,9A8)
      READ(5,500) N,NP,IT,IN,M,IOPT,IX,ICOR,IXNUMB
  500 FORMAT(6I5,I9,1X,I1,1X,I1)
      CORREL=.FALSE.
      IF( ICOR.NE.0 ) CORREL=.TRUE.
      READ(5,501) NLIST,(ILOW(I),IUP(I),I=1,NLIST)
  501 FORMAT(16I5)
      XNUMB=.FALSE.
      IF( IXNUMB.NE.0 ) XNUMB=.TRUE.
      IF( .NOT.XNUMB ) GO TO 55055
      READ(5,503) NXNUMB,(XNUMBS(I),I=1,NXNUMB)
  503 FORMAT(I3,5X,9Z8/(10Z8))
55055 CONTINUE
      RFAD(5,502) FMT
  502 FORMAT(10A8)
      WRITE(6,50505) LABEL,N,NP,IT,M
50505 FORMAT('1DATA DESCRIPTION : ',9A8/'-',6X,
     1 'NO. OF OBSERVATIONS ...............',I5/'0',5X,
     2 'NO. OF INDEPENDENT VARIABLES'/11X,'(EXCLUDES INTERCEPT) ........
     3.',I5/'0',6X,'SUBSET SIZE .',7('...'),I5/'0',5X,
     4 'NO. OF ''A'' VARIABLES DESIRED .....',I5)
      IF(IN.NE.8) IN=5
      IF(NP.GT.10) STOP
C
C************************************************************************
C
C       INITIALIZE VALUES
C
C************************************************************************
```

```
C
      IP=NP+1
      NB=NP+2
      NBA=NB*(NB+1)/2
      NBB=NB*(NB-1)/2
      ANIP=FLOAT(N-IP)
      AA=DSQRT(2./9./ANIP)
      AB=2./9./ANIP+1
      XTX(1)=N
      DO 1 I=2,78
    1 XTX(I)=0.0
C
C***********************************************************************
C                                                                      *
C     READ DATA AND FORM XTX                                           *
C                                                                      *
C***********************************************************************
C
C IS INPUT CORRELATION MATRIX ?
      IF( .NOT.CORREL ) GO TO 99099
      WRITE(6,99001)
99001 FORMAT('1INPUT CORRELATION MATRIX'//)
      L1=2
      DO 99000 I=1,NP
      L1=L1 + I
      L2=L1 + I - 1
      READ(IN,FMT) (XTX(J),J=L1,L2)
      WRITE(6,99002) (XTX(J),J=L1,L2)
99002 FORMAT(1X,7E18.9)
99000 CONTINUE
      GO TO 99990
99099 WRITE(6,99199)
99199 FORMAT('1INPUT DATA MATRIX'//)
      DO 10 I=1,N
      READ(IN,FMT) (X(J),J=2,IP)
      DO 5 J=2,IP
      JSUB=J*(J-1)/2
      XTX(JSUB+1)=XTX(JSUB+1)+X(J)
      DO 5 JJ=J,IP
      JJSUB=(JJ-1)*JJ/2
    5 XTX(JJSUB+J)=XTX(JJSUB+J)+X(J)*X(JJ)
      IF(IOPT.GE.1) WRITE(6,600) (X(J),J=2,IP)
   10 CONTINUE
  600 FORMAT(1X,7E18.9)
      DO 105 J=2,IP
      JSUB=J*(J-1)/2
      DO 105 JJ=J,IP
      JJSUB=JJ*(JJ-1)/2
      JSUBJ=JSUB+J
      JJSUB1=JJSUB+1
      JJSUBJ=JJSUB+J
      JSUB1=JSUB+1
      JJSBJJ=JJSUB+JJ
      XXN=N
      XNUM=XTX(JJSUBJ)-XTX(JJSUB1)*XTX(JSUB1)/XXN
      DENOM=(XTX(JJSBJJ)-XTX(JJSUB1)**2/XXN)*(XTX(JSUBJ)-XTX(
     1JSUB1)**2/XXN)
  105 XXTX(JJSUB+J)=XNUM/(DSQRT(DENOM))
      DO 205 J=2,IP
      JSUB=J*(J-1)/2
  205 XTX(JSUB+1)=0.
      DO 251 J=2,IP
      DO 251 I=J,IP
      II=I*(I-1)/2
  251 XTX(II+J)=XXTX(II+J)
C
C***********************************************************************
C                                                                      *
C     PRINT XTX IF DESIRED.                                            *
C                                                                      *
C***********************************************************************
C
99990 CONTINUE
      IF(IOPT.NE.2) GO TO 50
      IF( CORREL ) GO TO 50
      WRITE(6,99888)
99888 FORMAT('1CORRELATION MATRIX OF STANDARDIZED DATA'//)
      DO 60 I=1,IP
      K=I*(I-1)/2
   60 WRITE(6,640) (XTX(K+J),J=1,I)
  640 FORMAT(1X,7E18.9)
C
C***********************************************************************
C                                                                      *
C     FIND THE BIT PATTERN CORRESPONDING TO THE R-SQUARED FOR          *
C     THE NUMERATOR OF A                                               *
C                                                                      *
C***********************************************************************
C
   50 CONTINUE
      IF( XNUMB ) GO TO 60000
      NUMBR=NUMB(XTX,NP,IT,N)
      L1=1
      L2=1
      XNUMBS(1)=NUMBR
      GO TO 66666
C
C***********************************************************************
C                                                                      *
C     FOR A PROCEDURE DESCRIPTION SEE                                  *
C     COMPUTATIONAL PROCEDURES                                         *
C     (B) ESTIMATION OF C INVERSE                                      *
```

```
C
C      FACTOR XTX INTO LLT SUCH THAT L IS LOWER TRIANGULAR.
C
C***********************************************************************
C
60000 CONTINUE
      WRITE(6,61116) NXNUMB
61116 FORMAT(///'-WILL PROCESS',I5,' SPECIFIED REGRESSION SUBSET(S)')
      L1=1
      L2=NXNUMB
C SAVE XTX FOR PROCESSING EXTRA NUMBRS
      NXXTX=IP*(IP+1)/2
      DO 16661 I=1,NXXTX
16661 XXTX(I)=XTX(I)
66666 DO 77777 KX=L1,L2
      NUMBR=XNUMBS(KX)
      IF( XNUMB ) CALL SUBSET(NUMBR,NP,IT)
C
      IF( KX.GT.1 ) GO TO 60066
      CALL LUDECP(XTX,L,IP,DET1,DET2,IERROR)
      IF(IERROR.NE.129) GO TO 20
      WRITE(6,610)
  610 FORMAT(' XTX IS SINGULAR')
      STOP
   20 CONTINUE
      DO 49 I=1,IP
      K=I*(I+1)/2
   49 L(K)=1./L(K)
C
C***********************************************************************
C
C      CALCULATE THE 'A' VARIABLES.
C
C      GENERATE THE INDEPENDENT VARIABLE COLUMNS FROM THE NORMAL
C      RANDOM NUMBER GENERATOR.
C
C***********************************************************************
C
60066 CONTINUE
      DO 100 LL=1,M
      ZAP=GAUSS(1.0,0.0)
      XTX(NBA)=ANIP*(ZAP*AA+AB)**3
      DO 70 I=1,IP
      X(I)=GAUSS(1.0,0.0)
      XTX(NBA)=XTX(NBA)+X(I)**2
      XTX(NBB+I)=0.0
      K= I*(I-1)/2
      DO 70 J=1,I
   70 XTX(NBB+I)=XTX(NBB+I)+L(K+J)*X(J)
      CALL SCRNA
      IF( NUMBR.EQ.845014520 ) GO TO 77177
      A(LL)=(1.-RSQ2)/(1.-RSQ1)
  100 CONTINUE
C
C***********************************************************************
C                                                                      *
C     SORT THE A VARIABLES AND OUTPUT THEM AS SPECIFIED.               *
C                                                                      *
C***********************************************************************
C
      CALL VSORTA(A,M)
      WRITE(6,20202)
20202 FORMAT('-% POINT',11X,'A VARIABLE'/)
      DO 200 I=1,NLIST
      IA=ILOW(I)
      IB=IUP(I)
      IF( (IA.LE.0).OR.(IA.GT.M).OR.(IB.LE.0).OR.(IB.GT.M)) GO TO 210
  200 WRITE(6,620) (J,A(J),J=IA,IB)
  620 FORMAT(1X,I5,5X,E20.10)
C RESTORE XTX
77177 IF( .NOT.XNUMB ) GO TO 77777
      DO 71117 I=1,NXXTX
71117 XTX(I)=XXTX(I)
C
77777 CONTINUE
C
      GO TO 12345
  210 WRITE(6,630) IA,IB
  630 FORMAT(///1X,2I5,' INVALID RANGE FOR OUTPUT OF A VARIABLES')
      GO TO 12345
54321 CONTINUE
      WRITE(6,55544)
55544 FORMAT('1NORMAL TERMINATION; NO MORE DATA SETS')
      STOP
      END
```

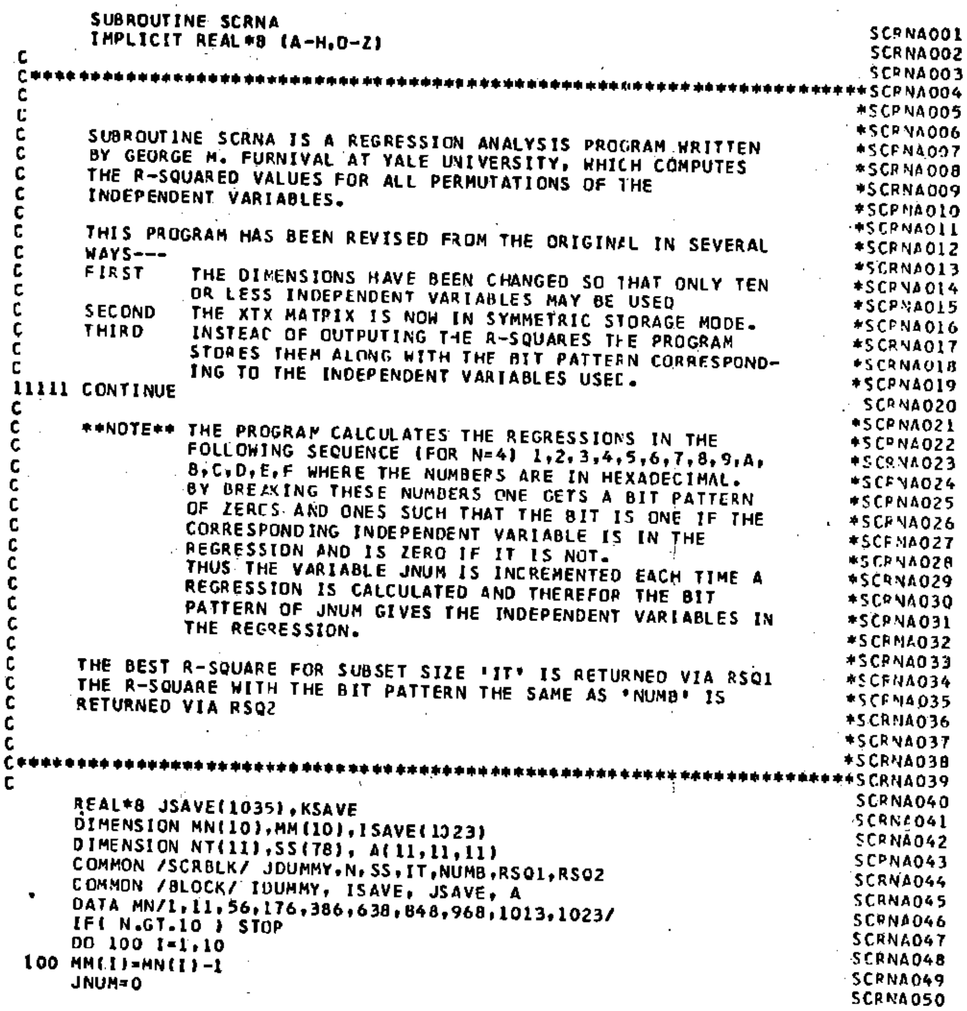


```
      SUBROUTINE SUBSET(NUMBR,P,T)
C
C PRINTS OUT THE SUBSET CCRRESPONDING TO THE BIT PATTERN OF 'NUMBR'.
C P = NUMBER OF INDEPENDENT VARIABLES
C T = SUBSET SIZE
C
      INTEGER P,T,SET(11),POW(11)
      DATA POW/1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024/
C
      N=NUMBR
      K=P
      KT=T+1
C
  100 IF( N.LT.POW(K) ) GO TO 200
      N=N-POW(K)
      KT=KT-1
      SET(KT)=K
      IF( N.LE.0 ) GO TO 300
  200 K=K-1
      GO TO 100
C
  300 WRITE(6,1) P,(SET(I),I=1,T)
    1 FORMAT('-SPECIFIED SUBSET FROM THE ',I2,' INDEPENDENT VARIABLES'
     1 //6X,'(INTERCEPT ALREADY INCLUDED) .......',11I6)
C
      RETURN
      END
      FUNCTION NUMB(XTX,P,T,N)
C
C***********************************************************************
C                                                                      *
C     FUNCTION NUMB FINDS THE INDEPENDENT VARIABLES OF THE             *
C     GIVEN SUBSET SIZE T TO BE USED IN FINDING                        *
C     A= (1.-RSQ(K))/(1.-RSQ(BEST)) IN THE MAINLINE                    *
C     PROGRAM.                                                         *
C                                                                      *
C     SEE COMPUTATIONAL PROCEDURES                                     *
C     (A) DETERMINATIUN OF K (PAGE 6-7)                                *
C                                                                      *
C     XTX   IS X TRANSPOSE X WHERE X IS THE INPUT DATA.                *
C           XTX IS IN SYMMFTRIC STORAGE MODE.                          *
C     P     IS THE NUMBER OF INDEPENDENT VARIABLES.                    *
C     T     IS THE GIVEN SUBSET SIZE.                                  *
C     N     IS THE NUMBER OF OBSERVATIONS.                             *
C                                                                      *
C     *****NOTE***** K*(K+1)/2 MUST NOT EXCEED THE DIMENSION           *
C     OF GAMMA AND GINV.   K IS THE NUMBER OF PERMUTATIONS             *
C     P THINGS TAKEN T AT A TIME.                                      *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XTX(1),X(11,11),Y(11,11),XY(11,11)
      DIMENSION ZONE(11,11), ZTWO(11,11)
      DIMENSION C(11,11), R(11), E(11)
      INTEGER PERM(93,11),P,T,PA,TA,SELECT(11)
C IDUMMY FORCES ALIGNMENT
      COMMON /BLOCK/ IDUMMY, PERM, GAMMA(1035), GINV(1331)
      EQUIVALENCE (Y(1),GINV(1)),(XY(1),GINV(122))
      EQUIVALENCE (ZONE(1),GINV(364)),(ZTWO(1),GINV(485))
      EQUIVALENCE (X(1),GINV(1210))
      EQUIVALENCE (C(1),GINV(606)),(R(1),GINV(727)),(E(1),GINV(738))
      PA=P+1
      TA=T+1
      AA=N-2*T-2
      AB=N-T-1
      CALL PRMUT(PA,TA,K)
      GAMMA(1)=1.0
C
C***********************************************************************
C                                                                      *
C     FOR EACH I=1,....,K AND J=1,....,K  THE PROGRAM                  *
C     DETERMINS X,Y,AND XY WHERE X REFERS TO THOSE ELEMENTS            *
C     OF XTX WHICH HAVE THE ROW COORDINATES PERM(I,1),....,            *
C     PERM(I,TA) AND THE COLUMN COORDINATES PERM(I,1),....,            *
C     PERM(I,TA).  Y IS FORMED THE SAME WAY EXCEPT THAT I IS           *
C     REPLACED BY J.                                                   *
```

```
C     XY IS FORMED BY THE (TA BY TA) SUBMATRIX OF XTX WITH THE          *
C     ROW COORDINATES PERM(I,1),....,PERM(I,TA) AND THE COLUMN          *
C     COORDINATES PERM(J,1),....,PERM(J,TA).                            *
C                                                                       *
C************************************************************************
C
      DO 200 I=2,K
      IM=I-1
      INDEX=I*(I+1)/2
      GAMMA(INDEX)=1.
C
C************************************************************************
C                                                                       *
C     DETERMINE X                                                       *
C                                                                       *
C************************************************************************
C
      DO 50 LOOP=1,TA
      INDEX=PERM(I,LOOP)
      DO 50 LOOPA=LOOP,TA
      INDEXA=PERM(I,LOOPA)
      ISUB=INOX(INDEX,INDEXA)
      X(LOOP,LOOPA)= XTX(ISUB)
      X(LOOPA,LOOP)= XTX(ISUB)
   50 CONTINUE
      DO 150 J=1,IM
C
C************************************************************************
C                                                                       *
C     DETERMINE Y AND XY                                                *
C                                                                       *
C************************************************************************
C
      DO 55 LOOPA=1,TA
      INDEXA=PERM(J,LOOPA)
      DO 54 LOOPB=LOOPA,TA
      INDEXB=PERM(J,LOOPB)
      ISUB=INOX(INDEXA,INDEXB)
      Y(LOOPA,LOOPB)=XTX(ISUB)
      Y(LOOPB,LOOPA)=XTX(ISUB)
   54 CONTINUE
      DO 55 LOOP=1,TA
      INDEX=PERM(I,LOOP)
      ISUB=INOX(INDEX,INDEXA)
   55 XY(LOOP,LOOPA)=XTX(ISUB)
C
C************************************************************************
C                                                                       *
C     FIND ZONE SUCH THAT X*ZONE=XY                                     *
C                                                                       *
C************************************************************************
```

```
C
      CALL LINEQ1(X,XY,ZONE,11,TA,TA,ISING,C,R,E)
      IF(ISING.EQ.0) GO TO 60
      WRITE(6,610) I,J
  610 FORMAT(1X,'MATRIX X IS SINGULAR',5X,2I5)
      STOP
C
C***********************************************************************
C                                                                      *
C     TAKE TRANSPOSE OF XY MATRIX.                                     *
C                                                                      *
C***********************************************************************
C
   60 DO 65 LOOP=1,T
      LOOPB=LOOP+1
      DO 65 LOOPA=LOOPB,TA
      SAVE=XY(LOOP,LOOPA)
      XY(LOOP ,LOOPA)=XY(LOOPA,LOOP)
   65 XY(LOOPA,LOOP)=SAVE
C
C***********************************************************************
C                                                                      *
C     FIND XTWO SUCH THAT Y*ZTWO=XY.                                   *
C                                                                      *
C***********************************************************************
C
      CALL LINEQ1(Y,XY,ZTWO,11,TA,TA,ISING,C,R,E)
      IF(ISING.EQ.0) GO TO 70
      WRITE(6,620) I,J
  620 FORMAT(1X,'MATRIX Y IS SINGULAR',5X,2I5)
      STOP
   70 CONTINUE
C
C***********************************************************************
C                                                                      *
C     FIND TRACE OF ZONE * ZTWO.                                       *
C     PERFORM OPERATION ON IT AND STORE IN GAMMA.                      *
C                                                                      *
C***********************************************************************
C
      SUM=0.0
      DO 75 LOOP=1,TA
      DO 75 LOOPB=1,TA
   75 SUM=SUM+ZONE(LOOP,LOOPB)*ZTWO(LOOPB,LOOP)
      INDEX=I*(I-1)/2+J
      GAMMA(INDEX)=(AA+SUM)/AB
  150 CONTINUE
  200 CONTINUE
C
C***********************************************************************
C                                                                      *
```

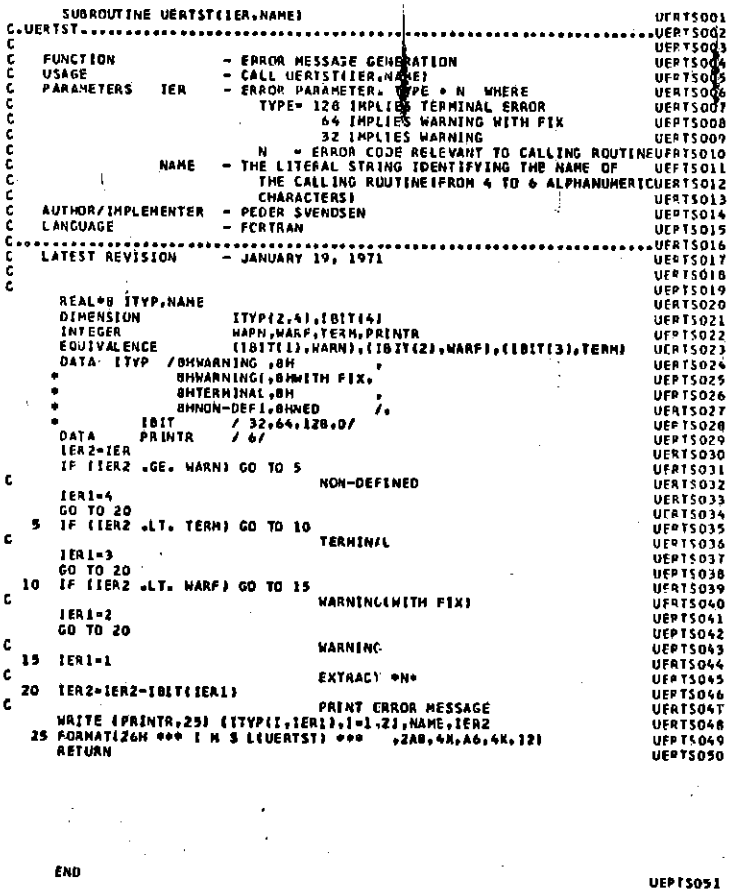
```
C     TAKE INVERSE OF GAMMA                                            *NUMB0153
C                                                                      *NUMB0154
C***********************************************************************NUMB0155
C                                                                       NUMB0156
      CALL LINV1P(GAMMA,K,GINV,7,X,IER)                                 NUMB0157
      IF(IER.NE.129) GO TO 210                                          NUMB0158
      WRITE(6,630)                                                      NUMB0159
  630 FORMAT(' GAMMA DID NOT HAVE AN INVERSE')                          NUMB0160
      STOP                                                              NUMB0161
C                                                                       NUMB0162
C***********************************************************************NUMB0163
C                                                                      *NUMB0164
C     FIND THE SMALLEST ELEMENT ON THE DIAGONAL OF GAMMA               *NUMB0165
C     INVERSE.                                                         *NUMB0166
C                                                                      *NUMB0167
C***********************************************************************NUMB0168
C                                                                       NUMB0169
  210 INDEX=1                                                           NUMB0170
      VALUE=GINV(1)                                                     NUMB0171
      DO 220 LOOP=2,K                                                   NUMB0172
      LOOPA=LOOP*(LOOP-1)/2+LOOP                                        NUMB0173
      IF(GINV(LOOPA).GE.VALUE) GO TO 220                                NUMB0174
      VALUE=GINV(LOOPA)                                                 NUMB0175
      INDEX=LOOP                                                        NUMB0176
  220 CONTINUE                                                          NUMB0177
C                                                                       NUMB0178
C***********************************************************************NUMB0179
C                                                                      *NUMB0180
C     THE  ELEMENTS(MINUS 1) OF THE (INDEX)TH PERMUTATION IN           *NUMB0181
C     PERM (2,....TA) REFER TO THE INDEPENDENT VARIABLES IN            *NUMB0182
C     THE DATA SET TO BE SEARCHED FOR IN CALCULATING THE               *NUMB0183
C     REGRESSION.                                                      *NUMB0184
C     THE INDEPENDENT VARIABLES ARE STORED IN NUMB ACCORDING           *NUMB0185
C     TO THEIR BIT PATTERN.                                            *NUMB0186
C     (I.E.  101110 REFERS TO THE INDEPENDENT VARIABLES                *NUMB0187
C     6,4,3,2.)                                                        *NUMB0188
C                                                                      *NUMB0189
C***********************************************************************NUMB0190
C                                                                       NUMB0191
      NUMB=0                                                            NUMB0192
      DO 230 I=2,TA                                                     NUMB0193
      J=PERM(INDEX,I)-2                                                 NUMB0194
      SELECT(I-1) = J+1                                                 NUMB0195
  230 NUMB=NUMB+2**J                                                    NUMB0196
C                                                                       NUMB0197
      WRITE(6,90000) P,(SELECT(I),I=1,T)                                NUMB0198
90000 FORMAT('-OPTIMUM SUBSET SELECTED FROM THE ',I2,                   NUMB0199
     1 ' INDEPENDENT VARIABLES'//6X,'(INTERCEPT ALREADY INCLUDED) ......NUMB0200
     2.',11I6)                                                          NUMB0201
      RETURN                                                            NUMB0202
      END                                                               NUMB0203
      FUNCTION INDX(I,J)
C
C***********************************************************************
C                                                                      *
C     FUNCTION INDX TAKES THE TWO INDICIES (I,J) OF A FULL             *
C     STORAGE MODE ARRAY AND COMPUTES THE CORRESPONDING                *
C     POSITION OF THAT ELEMENT IN AN ARRAY IN SYMMETRIC                *
C     STORAGE MODE.                                                    *
C                                                                      *
C***********************************************************************
C
      INDX=(I+J+IABS(I-J))/2
      INDX=INDX*(INDX-1)/2
      INDX=INDX+(I+J-IABS(I-J))/2
      RETURN
      END
```



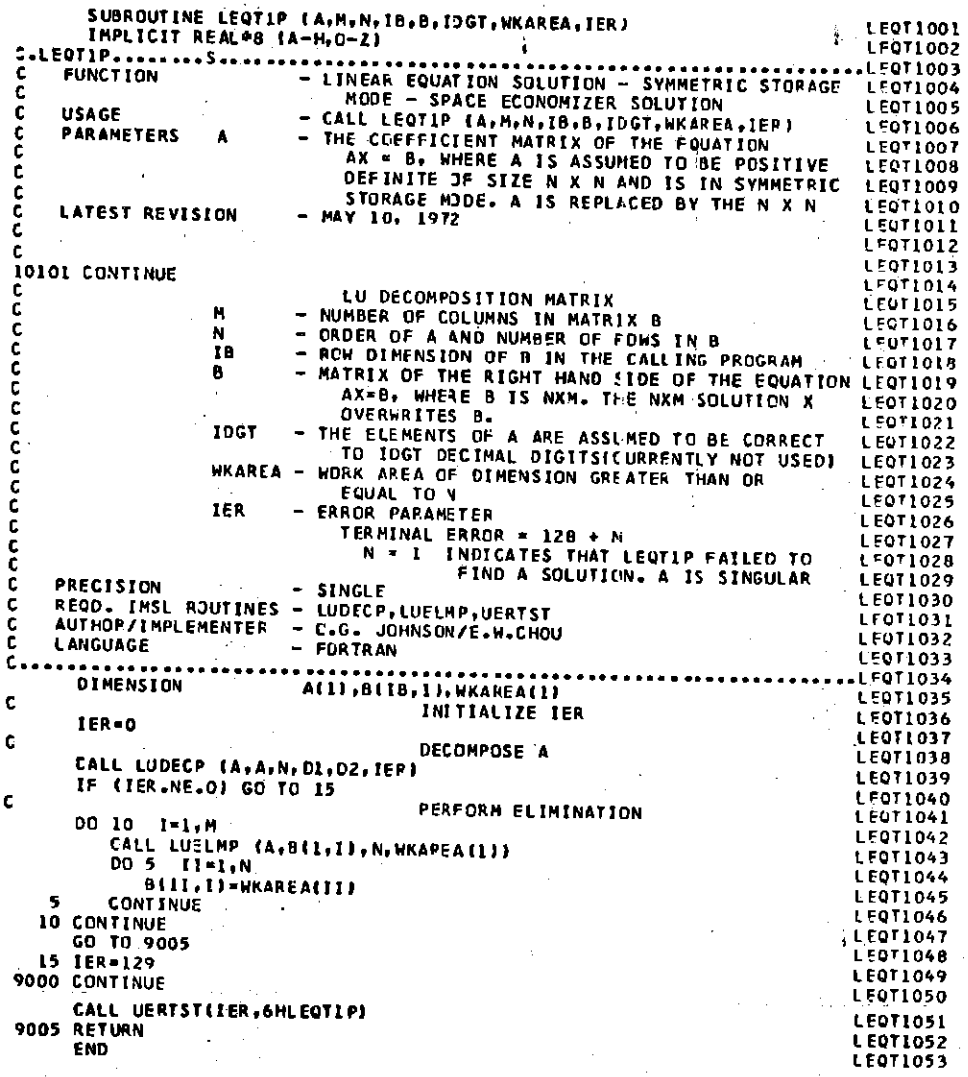
```
      SUBROUTINE SCRNA
      IMPLICIT REAL*8 (A-H,O-Z)
C
C*********************************************************************
C                                                                    *
C                                                                    *
C     SUBROUTINE SCRNA IS A REGRESSION ANALYSIS PROGRAM WRITTEN      *
C     BY GEORGE M. FURNIVAL AT YALE UNIVERSITY, WHICH COMPUTES       *
C     THE R-SQUARED VALUES FOR ALL PERMUTATIONS OF THE               *
C     INDEPENDENT VARIABLES.                                         *
C                                                                    *
C     THIS PROGRAM HAS BEEN REVISED FROM THE ORIGINAL IN SEVERAL     *
C     WAYS---                                                        *
C     FIRST     THE DIMENSIONS HAVE BEEN CHANGED SO THAT ONLY TEN    *
C               OR LESS INDEPENDENT VARIABLES MAY BE USED            *
C     SECOND    THE XTX MATRIX IS NOW IN SYMMETRIC STORAGE MODE.     *
C     THIRD     INSTEAD OF OUTPUTING THE R-SQUARES THE PROGRAM       *
C               STORES THEM ALONG WITH THE BIT PATTERN CORRESPOND-   *
C               ING TO THE INDEPENDENT VARIABLES USED.               *
11111 CONTINUE
C
C     **NOTE** THE PROGRAM CALCULATES THE REGRESSIONS IN THE         *
C              FOLLOWING SEQUENCE (FOR N=4) 1,2,3,4,5,6,7,8,9,A,     *
C              B,C,D,E,F WHERE THE NUMBERS ARE IN HEXADECIMAL.       *
C              BY BREAKING THESE NUMBERS ONE GETS A BIT PATTERN      *
C              OF ZEROS AND ONES SUCH THAT THE BIT IS ONE IF THE     *
C              CORRESPONDING INDEPENDENT VARIABLE IS IN THE          *
C              REGRESSION AND IS ZERO IF IT IS NOT.                  *
C              THUS THE VARIABLE JNUM IS INCREMENTED EACH TIME A     *
C              REGRESSION IS CALCULATED AND THEREFOR THE BIT         *
C              PATTERN OF JNUM GIVES THE INDEPENDENT VARIABLES IN    *
C              THE REGRESSION.                                       *
C                                                                    *
C     THE BEST R-SQUARE FOR SUBSET SIZE 'IT' IS RETURNED VIA RSQ1    *
C     THE R-SQUARE WITH THE BIT PATTERN THE SAME AS 'NUMB' IS        *
C     RETURNED VIA RSQ2                                              *
C                                                                    *
C                                                                    *
C                                                                    *
C*********************************************************************
C
      REAL*8 JSAVE(1035),KSAVE
      DIMENSION MN(10),MM(10),ISAVE(1023)
      DIMENSION NT(11),SS(78), A(11,11,11)
      COMMON /SCRBLK/ JDUMMY,N,SS,IT,NUMB,RSQ1,RSQ2
      COMMON /BLOCK/ IDUMMY, ISAVE, JSAVE, A
      DATA MN/1,11,56,176,386,638,848,968,1013,1023/
      IF( N.GT.10 ) STOP
      DO 100 I=1,10
  100 MM(I)=MN(I)-1
      JNUM=0
      NA=N+1
      NB=N+2
      NBA=NB*(NB-1)/2
      SQ=10./DSQRT(SS(NBA+NB)-SS(NBA+1)*SS(NBA+1)/SS(1))
      DO 1 I=2,NA
      ISUB=I*(I-1)/2
      NT(I-1)=0
      B=SS(ISUB+1)/SS(1)
      A(NA,1,I)=SQ*(B*SS(NBA+1)-SS(NBA+I))
      DO 1 J=I,NA
      JSUB=J*(J-1)/2
      A(NA,I,J)=B*SS(JSUB+1)-SS(JSUB+I)
    1 CONTINUE
      A(NA,1,1)=0.
      NT(NA)=1
      INUM=0
   11 DO 20 I=1,N
      IF( NT(I).EQ.0 ) GO TO 21
      INUM=INUM-1
      NT(I)=0
   20 CONTINUE
   10 LIMA=MN(IT)
      LIMB=MM(IT)
      RSQ1=JSAVE(LIMA)
      DO 60 LOOP=LIMA,LIMB
      IF( NUMB.EQ.ISAVE(LOOP) ) GO TO 61
   60 CONTINUE
      WRITE(6,600) NUMB
  600 FORMAT('0***** THE CORRECT BIT PATTERN COULD NOT BE FOUND : ',Z8)
      NUMB = 845014520
      RETURN
   61 RSQ2=JSAVE(LOOP)
      RETURN
   21 NT(I)=1
      JNUM=JNUM+1
      INUM=INUM+1
      K=I+1
      DO 22 J=K,NA
      IF( NT(J).EQ.1 ) GO TO 30
   22 CONTINUE
   30 DO 31 L=1,I
      B=A(J,L,K)/A(J,K,K)
      DO 31 M=L,I
      A(I,L,M)=A(J,L,M)-B*A(J,M,K)
   31 CONTINUE
      KSAVE=A(I,1,1)*.01
      MM(INUM)=MM(INUM)+1
      LIMA=MN(INUM)
      LIMB=MM(INUM)
      IF( LIMA.EQ.LIMB ) GO TO 50
      LIMA=LIMA+1
      DO 45 LOOPA=LIMA,LIMB                                             SCRNA102
      LOOP=LIMB+LIMA-LOOPA                                              SCRNA103
      IF( KSAVE.LE.JSAVE(LOOP-1) ) GO TO 55                             SCRNA104
      JSAVE(LOOP)=JSAVE(LOOP-1)                                         SCRNA105
   45 ISAVE(LOOP)=ISAVE(LOOP-1)                                         SCRNA106
   50 LOOP=MN(INUM)                                                     SCRNA107
   55 JSAVE(LOOP)=KSAVE                                                 SCRNA108
      ISAVE(LOOP)=JNUM                                                  SCRNA109
      GO TO 11                                                          SCRNA110
      END                                                               SCRNA111
      SUBROUTINE PRMUT(P,T,K)                                           PRMUT001
C                                                                       PRMUT002
C***********************************************************************PRMUT003
C                                                                      *PRMUT004
C     PRMUT IS A SUBROUTINE WHICH DETERMINES THE PERMUTAT-             *PRMUT005
C     IONS ON A SEQUENCE OF NUMBERS.                                   *PRMUT006
C     THE PERMUTATION IS TAKEN ON (P THINGS TAKEN T AT A               *PRMUT007
C     TIME).                                                           *PRMUT008
C     BECAUSE THE REGRESSION ANALYSIS PROGRAM SCRNA ALWAYS             *PRMUT009
C     INCLUDES AN INTERCEPT THE FIRST NUMBER IN EACH                   *PRMUT010
C     PERMUTATION WILL BE 1.  THUS THE PERMUTATIONS THAT               *PRMUT011
C     THE PROGRAM DETERMINES IS THE PERMUTATION (P-1)                  *PRMUT012
C     TAKEN (T-1) AT A TIME DONE ON THE INTEGERS 2,3,...,P.            *PRMUT013
C     THE USAGE OF THE SUBROUTINE IS                                   *PRMUT014
C             CALL PRMUT(P,T,K,PERM)                                   *PRMUT015
C     P        IS THE NUMBER OF INDEPENDENT VARIABLES PLUS 1.          *PRMUT016
C     T        IS THE SUBSET SIZE BEING CONSIDERED PLUS 1.             *PRMUT017
C     K        IS THE RESULTING NUMBER OF PERMUTATIONS.                *PRMUT018
C     PERM IS THE ARRAY WHICH STORES THE DETERMINED PERMU-             *PRMUT019
C             TATIONS.                                                 *PRMUT020
C     ALL ARGUMENTS ARE INTEGERS.                                      *PRMUT021
C                                                                      *PRMUT022
C***********************************************************************PRMUT023
C                                                                       PRMUT024
C IDUMMY FORCES ALIGNMENT                                               PRMUT025
      COMMON /BLOCK/ IDUMMY, PERM                                       PRMUT026
      INTEGER P,T,PERM(93,11),PM,TM                                     PRMUT027
      PM=P-1                                                            PRMUT028
C                                                                       PRMUT029
C***********************************************************************PRMUT030
C                                                                      *PRMUT031
C     CALCULATE K                                                      *PRMUT032
C                                                                      *PRMUT033
C***********************************************************************PRMUT034
C                                                                       PRMUT035
      K=1                                                               PRMUT036
      TM=T-1                                                            PRMUT037
      L=PM-TM+1                                                         PRMUT038
      DO 10 I=L,PM                                                      PRMUT039
   10 K=K*I                                                             PRMUT040
      DO 11 I=1,TM                                                      PRMUT041
   11 K=K/I                                                             PRMUT042
C                                                                       PRMUT043
C***********************************************************************PRMUT044
C                                                                      *PRMUT045
C     FILL OUT THE ARRAY PERM.                                         *PRMUT046
C                                                                      *PRMUT047
C***********************************************************************PRMUT048
C                                                                       PRMUT049
      I=1                                                               PRMUT050
      DO 15 LOOP=1,T
   15 PERM(1,LOOP)=LOOP
C
   20 I=I+1
      IF(I.EQ.K+1) GO TO 100
      DO 50 LOOP=1,T
   50 PERM(I,LOOP)=PERM(I-1,LOOP)
      IF(PERM(I,T).EQ.P) GO TO 55
      PERM(I,T)=PERM(I,T)+1
      GO TO 20
C
   55 DO 60 J=2,T
      L=T-J+2
      IF(PERM(I,L).LT.P-J+2) GO TO 70
   60 CONTINUE
      GO TO 100
C
   70 PERM(I,L)=PERM(I,L)+1
      LA=L+1
      DO 75 LOOP=LA,P
   75 PERM(I,LOOP)=PERM(I,LOOP-1)+1
      GO TO 20
C
  100 CONTINUE
      RETURN
      END
      FUNCTION GAUSS(SD,AM)
C
C COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A GIVEN
C MEAN (AM) AND STANDARD DEVIATION (SD)
C
C METHOD USED IS BOX-MULLER
C REF: NEWMAN AND ODELL, THE GENERATION OF RANDOM VARIATES.
C
      REAL*8 GAUSS,SD,AM
      LOGICAL FIRST
      COMMON /IWX/ IX
      DATA FIRST/.TRUE./
C
      IF( .NOT.FIRST ) GO TO 100
C
      CALL RANDU(IX,IY,U)
      IX=IY
      CALL RANDU(IX,IY,V)
      IX=IY
C
      S=SQRT( -2.*ALOG(U) )
      P=6.283185*V
      RN1=S*COS(P)
      RN2=S*SIN(P)
C
      GAUSS = RN1*SD + AM
      FIRST=.FALSE.
      RETURN
C
  100 GAUSS = RN2*SD + AM
      FIRST=.TRUE.
      RETURN
C
      END
      SUBROUTINE LINEQ1(A,B,X,ND,N,NR,S,C,R,E)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(ND,ND),B(ND,NR),X(ND,NR),C(N,N),R(N),E(N)
      INTEGER S
      ABS(D)=DABS(D)
      AMAX1(F,G)=DMAX1(F,G)
C
C     .............................................................
C
C
C        SUBROUTINE LINEQ1
C
C        DECKNAME - LINEQ1
C
C        PURPOSE
C           SOLVES THE REAL MATRIX EQUATION AX=B WITH NR RIGHT-HAND
C           SIDES.
C
C        USAGE
C           CALL LINEQ1(A,B,X,ND,N,NR,S)
C
C        DESCRIPTION OF PARAMETERS
C              A       - (N X N) REAL COEFFICIENT MATRIX.
C              B       - (N X NR) REAL RIGHT-HAND SIDE ARRAY.
C              X       - (N X NR) REAL ARRAY FOR RETURN OF SOLUTION
C                        VECTORS.
C              ND      - THE NUMBER OF ROWS FOR THE ARRAYS A, B, AND X
C                        IN THE DIMENSION STATEMENT IN THE USER PROGRAM.
C              N       - THE NUMBER OF EQUATIONS TO BE SOLVED.
C              NR      - THE NUMBER OF RIGHT-HAND SIDES TO BE SOLVED.
C              S       - INTEGER VARIABLE RETURNED NON-ZERO ONLY IF
C                        MATRIX A IS SINGULAR TO MACHINE ACCURACY.
   90 CONTINUE
C
C        REMARKS
C           ARRAYS A AND B ARE NOT DESTROYED.
C
C           EXTRA PARAMETERS IN THE ACTUAL FORMAL PARAMETER LIST
C           ARE USED IN CONJUNCTION WITH THE SUBROUTINE DYNAMIC.
C
C        METHOD
C           THE MATRIX A IS FACTORED INTO LOWER AND UPPER TRIANGULAR
C           MATRICES L AND U AND THEN THE EQUATIONS LZ=B AND UX=Z ARE
C           SOLVED IN TURN.  DOUBLE PRECISION ACCUMULATION OF INNER
C           PRODUCTS AND ITERATIVE REFINEMENT ARE USED SO SOLUTIONS
C           ARE VERY ACCURATE WHENEVER S IS RETURNED EQUAL TO ZERO.
C
C           RALSTON AND WILF   MATHEMATICAL METHODS FOR DIGITAL
C           COMPUTERS   VOLUME 2   WILEY 1967.
C
```

```
C           WRITTEN BY DAVID S. DODSON,  06/01/70
C                      COMPUTER SCIENCES DEPARTMENT
C                      PURDUE UNIVERSITY
C
C     ..............................................................
C
C
C     FORM EQUILIBRATION FACTORS IN VECTOR E.
C
      DO 12 I=1,N
      P=0.0
      DO 10 J=1,N
      C(I,J)=A(I,J)
   10 P=AMAX1(ABS(C(I,J)),P)
      IF(P.EQ.0.0)GO TO 73
   12 E(I)=1.0/P
C
C     FACTOR COEFFICIENT MATRIX WITH PARTIAL PIVOTING.
C
      M=1
   14 MM=M-1
      P=0.0
      DO 22 I=M,N
      C(I,M)=ARITH1(C(I,M),MM,C(I,1),N,C(1,M),1)
      Q=E(I)*ABS(C(I,M))
      IF(P.GE.Q)GO TO 22
      P=Q
      K=I
   22 CONTINUE
      IF (N+P.EQ.N) GO TO 73
      IF(M.EQ.K)GO TO 30
      DO 28 J=1,N
      P=C(M,J)
      C(M,J)=C(K,J)
   28 C(K,J)=P
      E(K)=E(M)
   30 E(M)=K
      IF(M.EQ.N)GO TO 37
      MP=M+1
      DO 34 J=MP;N
   34 C(M,J)=ARITH1(C(M,J),MM,C(M,1),N,C(1,J),1)/C(M,M)
      M=MP
      GO TO 14
C
C     BACK SUBSTITUTE RIGHT-HAND SIDES WITH ITERATIVE REFINEMENT.
C
   37 IF(NR.LE.0)GO TO 71
      DO 68 M=1,NR
      P=0.0
      Q=0.0
      DO 43 I=1,N
      R(I)=B(I,M)
   43 X(I,M)=0.0
   44 DO 48 I=1,N
      K=E(I)
      T=R(K)
      R(K)=R(I)
   48 R(I)=ARITH1(T,I-1,C(I,1),N,R(1),1)/C(I,I)
      I=N
      GO TO 52
   51 R(I)=ARITH1(R(I),N-I,C(I,IP),N,R(IP),1)
   52 IP=I
      I=I-1
      IF(I.GT.0)GO TO 51
      T=P
      P=0.0
      DO 59 I=1,N
      P=AMAX1(ABS(R(I)),P)
   59 X(I,M)=X(I,M)+R(I)
C
C     TEST FOR CONVERGENCE OF ITERATIVE REFINEMENT.
C
      IF(P.EQ.0.0)GO TO 68
      IF(Q.EQ.0.0)Q=P
      IF ( SNGL(Q+P).EQ. SNGL(Q)) GO TO 68
      IF(T.NE.0.0.AND.P+P.GT.T)GO TO 67
      DO 65 I=1,N
   65 R(I)=ARITH1(B(I,M),N,A(I,1),ND,X(1,M),1)
      GO TO 44
   67 IF ( SNGL(Q+P/N) .NE. SNGL(Q) ) GO TO 73
   68 CONTINUE
C
C     SET NON-SINGULAR/SINGULAR FLAG AND RETURN.
C
   71 S=0
      RETURN
C
   73 S=1
      WRITE(6,700) Q,P,N
  700 FORMAT('-MATRIX SINGULAR IN LINEQ1 : Q, P, N :',5X,
     1 2E20.8,I5)
      RETUPN
C
      END
      FUNCTION ARITH1(C,N,A,KA,B,KB)                                    ARITH001
C                                                                       ARITH002
C CALLED FROM LINEQ1                                                    ARITH003
C                                                                       ARITH004
      IMPLICIT REAL*8 (A-H,O-Z)                                         ARITH005
      DIMENSION A(KA,N),B(KB,N)                                         ARITH006
      T=C                                                               ARITH007
      IF(N.EQ.0) GO TO 5                                                ARITH008
      DO 4 I=1,N                                                        ARITH009
    4 T=T-A(1,I)*B(1,I)                                                 ARITH010
    5 ARITH1=T                                                          ARITH011
      RETURN                                                            ARITH012
      END                                                               ARITH013
      SUBROUTINE VSORTM (A,LA)
C.VSORTM........S.....................................................
C.VSORTA
C
C    FUNCTION        VSORTM - SORT ARRAYS BY ABSOLUTE VALUE
C                    VSORTA - SORT ARRAYS BY ALGEBRAIC VALUE
C    USAGE                  - CALL VSORTM (A,LA)
C                           - CALL VSORTA (A,LA)
C    PARAMETERS     A       - ON INPUT, CONTAINS THE ARRAY TO BE SORTED
C                             ON OUTPUT, A CONTAINS THE SORTED ARRAY
C                   LA      - INPUT VAPIABLE CONTAINING THE NUMBER OF
C                               ELEMENTS IN THE ARRAY TO BE SORTED
C    PRECISION              - SINGLE
C    AUTHOR/IMPLEMENTER     - N.E. BOSTEN
C    LANGUAGE               - FOPTRAN
C.....................................................................
C    LATEST REVISION        - DECEMBER 7,1970
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION            A(1),IU(21),IL(21)
C
                                 FIND ABSOLUTE VALUES OF ARRAY A
        DO 5 I=1,LA
        IF (A(I) .LT. 0.0) A(I)=-A(I)
    5   CONTINUE
C
      ENTRY VSORTA (A,LA)
C
      M=1
      I=1
      J=LA
      R=.375
   10 IF (I .EQ. J) GC TO 55
   15 IF (R .GT. .5898437) GO TO 20
      R=R+3.90625E-2
      GO TO 25
   20 R=R-.21875
   25 K=I
C
C                                SELECT A CENTRAL ELEMENT OF THE
      IJ=I+(J-I)*R               ARRAY AND SAVE IT IN LOCATION T
      T=A(IJ)
C
C                                IF FIRST ELEMENT OF ARRAY IS GREATER
      IF (A(I) .LE. T) GC TO 30  THAN T, INTERCHANGE WITH T
      A(IJ)=A(I)
      A(I)=T
      T=A(IJ)
   30 L=J
C
                                 IF LAST ELEMENT OF ARRAY IS LESS THAN
```

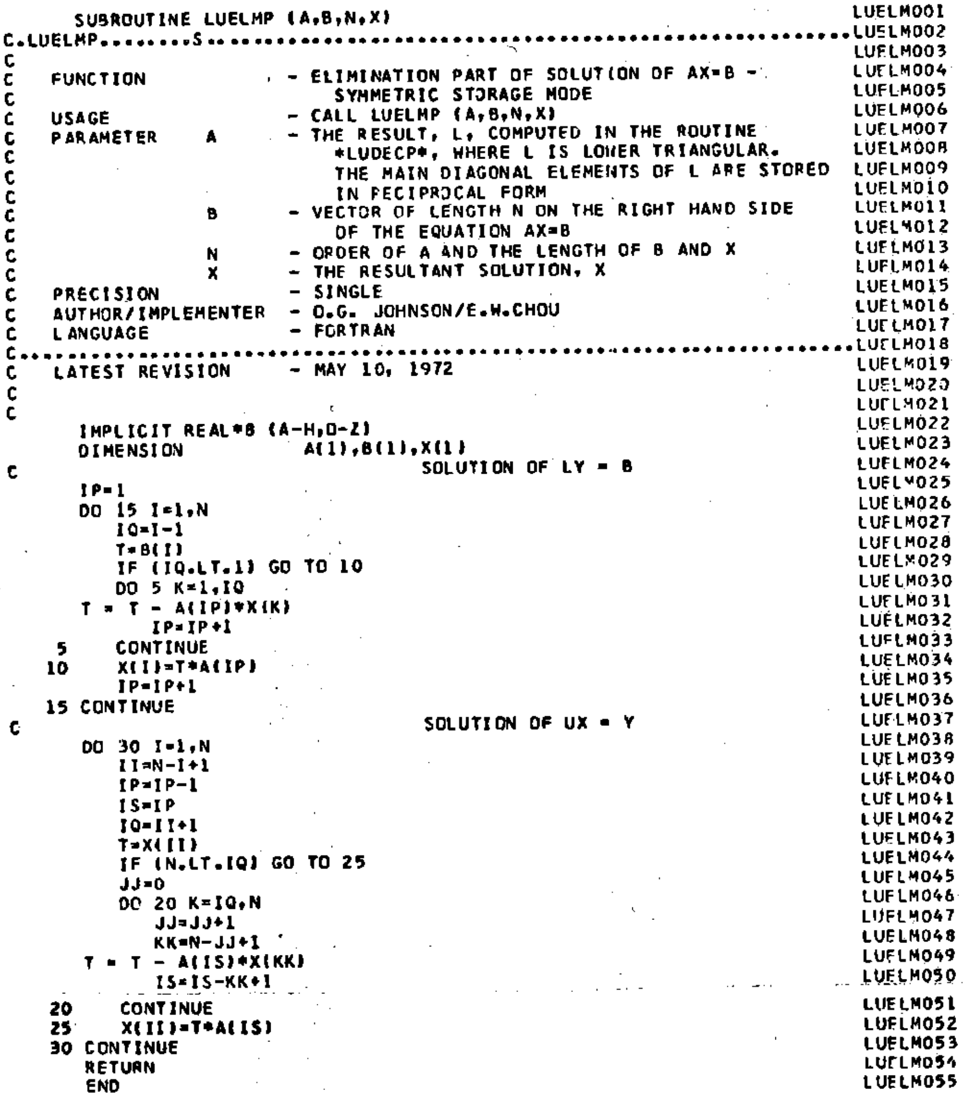
```
C                                                 T, INTERCHANGE WITH T
      IF (A(J) .GE. T) GO TO 40
      A(IJ)=A(J)
      A(J)=T
      T=A(IJ)
C
C                                                 IF FIRST ELEMENT OF ARRAY IS GREATER
                                                  THAN T, INTERCHANGE WITH T
      IF (A(I) .LE. T) GO TO 40
      A(IJ)=A(I)
      A(I)=T
      T-A(IJ)
      GO TO 40
   35 TT=A(L)
      A(L)=A(K)
      A(K)=TT
C
C                                                 FIND AN ELEMENT IN THE SECOND HALF OF
                                                  THE ARRAY WHICH IS SMALLER THAN T
   40 L=L-1
      IF (A(L) .GT. T) GO TO 40
C
C                                                 FIND AN ELEMENT IN THE FIRST HALF OF
                                                  THE ARRAY WHICH IS GREATER THAN T
   45 K=K+1
      IF (A(K) .LT. T) GO TO 45
C                                                 INTERCHANGE THESE ELEMENTS
      IF (K .LE. L) GO TO 35
C
C                                                 SAVE UPPER AND LOWER SUBSCRIPTS OF
                                                  THE ARRAY YET TO BE SORTED
      IF (L-I .LE. J-K) GO TO 50
      IL(M)=I
      IU(M)=L
      I=K
      M=M+1
      GO TO 60
   50 IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 60
C
C                                                 BEGIN AGAIN ON ANOTHER PORTION OF
                                                  THE UNSORTED ARRAY
   55 M=M-1
      IF (M .EQ. 0) RETURN
      I=IL(M)
      J=IU(M)
   60 IF (J-I .GE. 1) GO TO 25
      IF (I .EQ. 1) GO TO 10
      I=I-1
   65 I=I+1
      IF (I .EQ. J) GO TO 55
      T=A(I+1)
      IF (A(I) .LE. T) GO TO 65
      K=I
   70 A(K+1)=A(K)
      K=K-1
      IF (T .LT. A(K)) GO TO 70
      A(K+1)=T
      GO TO 65
      END
      SUBROUTINE LUDECP (A,UL,N,D1,D2,IER)
C.LUDECP........S.........................................................
C       .................................................................
C   FUNCTION            - CHOLESKY DECOMPOSITION OF A MATRIX -
C                           SYMMETRIC STORAGE MODE
C   USAGE               - CALL LUDECP (A,UL,N,D1,D2,IER)
C   PARAMETERS   A      - A IS N X N POSITIVE DEFINITE MATRIX AND IS
C                           IN SYMMETRIC STOPAGE MODE.
C                UL     - THE RESULT L OF THIS ROUTINE IS STORED IN THE
C                           MATRIX UL. UL MAY OCCUPY THE SAME STORAGE AS
C                           A. THE MAIN DIAGONAL VALUES OF UL AFF STOPFD
C                           IN RECIPROCAL FORM.
C                N      - ORDER OF A
C                D1     - A NUMBEP COMPUTED IN THE SUBROUTINE SUCH THAT
C                           THE DETERMINANT OF A = D1*2.**D2
C                D2     - SFE D1
C                IER    - ERROR PARAMETER
C                           TERMINAL ERROR = 128 + N
C                             N = 1 INDICATES THAT "HE MATRIX IS
C                                     SINGULAR
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - UERTST
C   AUTHOR/IMPLEMENTEP  - O.G. JOHNSON/E.W.CHOU
C   LANGUAGE            - FORTRAN
C.........................................................................
C   LATEST REVISION     - MAY 10, 1972.......................................
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION          A(1),UL(1)
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.0D0,1.D0,4.D0,16.D0
     1,.0625D0/
      ABS(D)=DABS(D)
      D1=ONE
      D2=ZERO
      IP=0
      IER=0
      DO 40  I=1,N
         IQ = IP+1
         IR = 0
         DO 35  J=1,I
            X=A(IP+1)
            IF (IP.LT.IQ) GO TO 10
            DO 5  K=IQ,IP
               IR = IR+1
     X = X - UL(K)*UL(IR)
    5       CONTINUE
   10       IR = IR+1
            IP = IP+1
            IF (I.NE.J) GC TO 30
            D1 = D1*X
            IF (X.NE.0.0D0) GO TO 15
            D2=ZERO
            GO TO 45
  15        IF (ABS(D1).LT.ONE) GO TO 20
            D1 = D1 * SIXTH
            D2 = D2 + FOUR
            GO TO 15
  20        IF (ABS(D1) .GE. SIXTH) G) TO 25
            D1 = D1 * SIXTN
            D2 = D2 - FOUR
            GO TO 20
  25        IF (X.LT.0.0D0) GO TO 45
            UL(IP) = 1.D0/DSQRT(X)
            GO TO 35
  30        UL(IP) = X * UL(IR)
  35     CONTINUE
  40 CONTINUE
     GO TO 9005
  45 IER = 129
9000 CONTINUE
     CALL UERTST(IER,6HLUDECP)
9005 RETUPN
     END
```



```
      SUBROUTINE UERTST(IER,NAME)                                       UERTS001
C.UERTST.................................................................UERTS002
C                                                                       UERTS003
C   FUNCTION            - ERROR MESSAGE GENERATION                      UERTS004
C   USAGE               - CALL UERTST(IER,NAME)                         UERTS005
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE              UERTS006
C                           TYPE= 128 IMPLIES TERMINAL ERROR            UERTS007
C                                  64 IMPLIES WARNING WITH FIX          UERTS008
C                                  32 IMPLIES WARNING                   UERTS009
C                           N   = ERROR CODE RELEVANT TO CALLING ROUTINEUERTS010
C                NAME   - THE LITERAL STRING IDENTIFYING THE NAME OF    UERTS011
C                           THE CALLING ROUTINE(FROM 4 TO 6 ALPHANUMERICUERTS012
C                           CHARACTERS)                                 UERTS013
C   AUTHOR/IMPLEMENTER  - PEDER SVENDSEN                                UERTS014
C   LANGUAGE            - FORTRAN                                       UERTS015
C.......................................................................UERTS016
C   LATEST REVISION     - JANUARY 19, 1971                              UERTS017
C                                                                       UERTS018
C                                                                       UERTS019
      REAL*8 ITYP,NAME                                                  UERTS020
      DIMENSION          ITYP(2,4),IBIT(4)                              UERTS021
      INTEGER            WARN,WARF,TERM,PRINTR                          UERTS022
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)   UERTS023
      DATA  ITYP  /8HWARNING ,8H        ,                               UERTS024
     *             8HWARNING(,8HWITH FIX,                               UERTS025
     *             8HTERMINAL,8H        ,                               UERTS026
     *             8HNON-DEFI,8HNED     /,                              UERTS027
     *       IBIT      / 32,64,128,0/                                   UERTS028
      DATA     PRINTR    / 6/                                           UERTS029
      IER2=IER                                                          UERTS030
      IF (IER2 .GE. WARN) GO TO 5                                       UERTS031
C                                   NON-DEFINED                         UERTS032
      IER1=4                                                            UERTS033
      GO TO 20                                                          UERTS034
    5 IF (IER2 .LT. TERM) GO TO 10                                      UERTS035
C                                   TERMINAL                            UERTS036
      IER1=3                                                            UERTS037
      GO TO 20                                                          UERTS038
   10 IF (IER2 .LT. WARF) GO TO 15                                      UERTS039
C                                   WARNING(WITH FIX)                   UERTS040
      IER1=2                                                            UERTS041
      GO TO 20                                                          UERTS042
C                                   WARNING                             UERTS043
   15 IER1=1                                                            UERTS044
C                                   EXTRACT *N*                         UERTS045
   20 IER2=IER2-IBIT(IER1)                                              UERTS046
C                                   PRINT ERROR MESSAGE                 UERTS047
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,2),NAME,IER2                  UERTS048
   25 FORMAT(26H *** I M S L(UERTST) ***    ,2A8,4X,A6,4X,I2)           UERTS049
      RETURN                                                            UERTS050




      END                                                               UERTS051
```



```
      SUBROUTINE LEQT1P (A,M,N,IB,B,IDGT,WKAREA,IER)                    LEQT1001
      IMPLICIT REAL*8 (A-H,O-Z)                                         LEQT1002
C.LEQT1P.........S.....................................................LEQT1003
C   FUNCTION            - LINEAR EQUATION SOLUTION - SYMMETRIC STORAGE  LEQT1004
C                           MODE - SPACE ECONOMIZER SOLUTION            LEQT1005
C   USAGE               - CALL LEQT1P (A,M,N,IB,B,IDGT,WKAREA,IER)      LEQT1006
C   PARAMETERS   A      - THE COEFFICIENT MATRIX OF THE EQUATION        LEQT1007
C                           AX = B, WHERE A IS ASSUMED TO BE POSITIVE   LEQT1008
C                           DEFINITE OF SIZE N X N AND IS IN SYMMETRIC  LEQT1009
C                           STORAGE MODE. A IS REPLACED BY THE N X N    LEQT1010
C   LATEST REVISION     - MAY 10, 1972                                  LEQT1011
C                                                                       LEQT1012
C                                                                       LEQT1013
10101 CONTINUE                                                          LEQT1014
C                           LU DECOMPOSITION MATRIX                     LEQT1015
C                M      - NUMBER OF COLUMNS IN MATRIX B                 LEQT1016
C                N      - ORDER OF A AND NUMBER OF ROWS IN B            LEQT1017
C                IB     - ROW DIMENSION OF B IN THE CALLING PROGRAM     LEQT1018
C                B      - MATRIX OF THE RIGHT HAND SIDE OF THE EQUATION LEQT1019
C                           AX=B, WHERE B IS NXM. THE NXM SOLUTION X    LEQT1020
C                           OVERWRITES B.                               LEQT1021
C                IDGT   - THE ELEMENTS OF A ARE ASSUMED TO BE CORRECT   LEQT1022
C                           TO IDGT DECIMAL DIGITS(CURRENTLY NOT USED)  LEQT1023
C                WKAREA - WORK AREA OF DIMENSION GREATER THAN OR        LEQT1024
C                           EQUAL TO N                                  LEQT1025
C                IER    - ERROR PARAMETER                               LEQT1026
C                           TERMINAL ERROR = 128 + N                    LEQT1027
C                             N = 1  INDICATES THAT LEQT1P FAILED TO    LEQT1028
C                                      FIND A SOLUTION. A IS SINGULAR   LEQT1029
C   PRECISION           - SINGLE                                        LEQT1030
C   REQD. IMSL ROUTINES - LUDECP,LUELMP,UERTST                          LEQT1031
C   AUTHOR/IMPLEMENTER  - C.G. JOHNSON/E.W.CHOU                         LEQT1032
C   LANGUAGE            - FORTRAN                                       LEQT1033
C......................................................................LEQT1034
      DIMENSION          A(1),B(IB,1),WKAREA(1)                         LEQT1035
C                                  INITIALIZE IER                       LEQT1036
      IER=0                                                             LEQT1037
C                                  DECOMPOSE A                          LEQT1038
      CALL LUDECP (A,A,N,D1,D2,IER)                                     LEQT1039
      IF (IER.NE.0) GO TO 15                                            LEQT1040
C                                  PERFORM ELIMINATION                  LEQT1041
      DO 10  I=1,M                                                      LEQT1042
         CALL LUELMP (A,B(1,I),N,WKAREA(1))                             LEQT1043
         DO 5  II=1,N                                                   LEQT1044
            B(II,I)=WKAREA(II)                                          LEQT1045
    5    CONTINUE                                                       LEQT1046
   10 CONTINUE                                                          LEQT1047
      GO TO 9005                                                        LEQT1048
   15 IER=129                                                           LEQT1049
 9000 CONTINUE                                                          LEQT1050
      CALL UERTST(IER,6HLEQT1P)                                         LEQT1051
 9005 RETURN                                                            LEQT1052
      END                                                               LEQT1053
      SUBROUTINE LINV1P (A,N,AINV,IDGT,WKAREA,IER)
C.LINV1P........S.........................................................
C
C   FUNCTION            - INVERSION OF MATRIX - SYMMETRIC STORAGE MODE -
C                            SPACE ECONOMIZER SOLUTION
C   USAGE               - CALL LINV1P (A,N,AINV,IDGT,WKAREA,IER)
C   PARAMETERS   A      - A GIVEN POSITIVE DEFFINITE MATRIX OF SIZE
C                            N X N. A IS DESTROYED, REPLACED BY THE LU
C                            DECOMPOSITION MATRIX.
C                N      - ORDER OF A
C                AINV   - THE INVERSE OF A IS STORED IN THE N X N MATRIX
C                            AINV (IN SYMMETRIC STORAGE MODE)
C                IDGT   - THE ELEMENTS OF A ARE ASSUMED TO BE CORRECT
C                            TO IDGT DECIMAL PLACES.(CURRENTLY NOT USED)
C                WKAREA - WORK AREA OF DIMENSION GREATER THAN OR EQUAL
C                            TO 2*N
C                IER    - ERRCR PARAMETER
C                            TERMINAL ERROR = 128 + N
C                               N=1,  INDICATES THAT LINV1P FAILED TO
C                                        FIND THE INVERSE
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - LEQT1P,LUELMP,UERTST,LUDECP
C   AUTHOR/IMPLEMENTER  - D.G. JOHNSON/E.W.CHOU
C   LANGUAGE            - FORTRAN
C.........................................................................
C   LATEST REVISION     - MAY 10, 1972
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION          A(1),AINV(1),WKAREA(1)
      DATA               ZERO,ONE/0.0D0,1.0D0/
      IER=0
      K = N
      DO 25  I=1,N
         DO 5  J=1,N
            L=K+J
            WKAREA(L)=ZERO
            IF (J.EQ.I) WKAREA(L)=ONE
    5    CONTINUE
         IF (I.NE.1) GO TO 10
C                                  DECOMPOSE A
         CALL LEQT1P (A,1,N,N,WKAREA(K+1),IDGT,WKAREA,IER)
         IF (IER.EQ.0) GO TO 15
         GO TO 30
C                                  COMPUTE THE INVERSE AND MOVE INTO
C                                  ARRAY AINV
   10    CALL LUELMP (A,WKAREA(K+1),N,WKAREA(K+1))
   15    K1 = I*(I-1)/2 +1
         DO 20  J=I,N
            K1 = K1+J-1
            K2 = K+J
            AINV(K1)=WKAREA(K2)
   20    CONTINUE
   25 CONTINUE
      GO TO 9005
   30 IER=129
 9000 CONTINUE
      CALL UERTST(IER,6HLINV1P)
 9005 RETURN
      END
```



```
      SUBROUTINE LUELMP (A,B,N,X)
C.LUELMP........S........................................................
C
C     FUNCTION            - ELIMINATION PART OF SOLUTION OF AX=B -
C                              SYMMETRIC STORAGE MODE
C     USAGE               - CALL LUELMP (A,B,N,X)
C     PARAMETER    A      - THE RESULT, L, COMPUTED IN THE ROUTINE
C                              *LUDECP*, WHERE L IS LOWER TRIANGULAR.
C                              THE MAIN DIAGONAL ELEMENTS OF L ARE STORED
C                              IN RECIPROCAL FORM
C                  B      - VECTOR OF LENGTH N ON THE RIGHT HAND SIDE
C                              OF THE EQUATION AX=B
C                  N      - ORDER OF A AND THE LENGTH OF B AND X
C                  X      - THE RESULTANT SOLUTION, X
C     PRECISION           - SINGLE
C     AUTHOR/IMPLEMENTER  - O.G. JOHNSON/E.W.CHOU
C     LANGUAGE            - FORTRAN
C........................................................................
C     LATEST REVISION     - MAY 10, 1972
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION          A(1),B(1),X(1)
C                                       SOLUTION OF LY = B
      IP=1
      DO 15 I=1,N
         IQ=I-1
         T=B(I)
         IF (IQ.LT.1) GO TO 10
         DO 5 K=1,IQ
      T = T - A(IP)*X(K)
            IP=IP+1
    5    CONTINUE
   10    X(I)=T*A(IP)
         IP=IP+1
   15 CONTINUE
C                                       SOLUTION OF UX = Y
      DO 30 I=1,N
         II=N-I+1
         IP=IP-1
         IS=IP
         IQ=II+1
         T=X(II)
         IF (N.LT.IQ) GO TO 25
         JJ=0
         DO 20 K=IQ,N
            JJ=JJ+1
            KK=N-JJ+1
      T = T - A(IS)*X(KK)
            IS=IS-KK+1
   20    CONTINUE
   25    X(II)=T*A(IS)
   30 CONTINUE
      RETURN
      END
```