

**IBM APPLICATION PROGRAM
IBM 1130 SCIENTIFIC SUBROUTINE PACKAGE
SYSTEM MANUAL**

This manual provides flowchart information on the logic used in each subroutine of the Scientific Subroutine Package. All subroutine descriptions and listings are contained in the Programmer's Manual.

(1130-CM-02X)

CONTENTS

Introduction.	1
Guide to Flowcharts	3

INTRODUCTION

Each subroutine in the Scientific Subroutine Package is documented by means of a flowchart. In the guide to the flowcharts each flowchart is listed under a general heading: statistics, special matrix operations, matrices, or other mathematics; and under a subheading relating to its basic function.

SYMBOLS USED

Figure 1 illustrates the various blocks used in the flowcharts and their particular meaning. Lines connecting these blocks are made up of asterisks. Arrows showing the direction of flow are represented by a "V" for downward, a left parenthesis "(" for right to left, and a right parenthesis ")" for left to right. Special symbols are used for mathematical operators within the flowchart boxes. These are "*", "&", and "-" for multiplication, addition, and subtraction respectively.

BLOCK NUMBERING

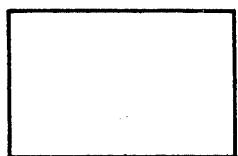
Numbers appearing inside the connector circles refer to the chart block location on the page; these are numbered vertically zero to nine and horizontally A to C. Numbers shown outside the blocks correspond to FORTRAN statement numbers in the programs.

EQUATIONS

Many of the flowcharts contain references to equation numbers. These equations can be found in the mathematical description for the particular subroutines in the Programmer's Manual.



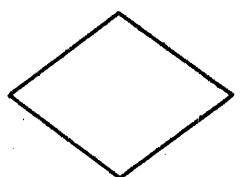
Enter or exit block



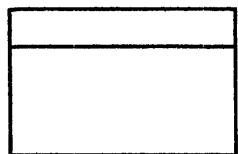
Processor block



Modification block



Decision block



Call to subroutine block



Connector to block within page



Connector to block on another page

Figure 1. Flowchart blocks

GUIDE TO FLOWCHARTS

FLOWCHARTS

Statistics

Data Screening

TALLY — totals, means, standard deviations, minimums, and maximums.	8
BOUND — selection of observations within bounds	9
SUBST — subset selection from observation matrix	10
ABSNT — detection of missing data.	11
TAB1 — tabulation of data (1 variable)	12
TAB2 — tabulation of data (2 variables)	13
SUBMX — build subset matrix	15

Elementary Statistics

MOMEN — first four moments.	16
TTSTT — tests on population means	17

Correlation

CORRE — means, standard deviations, and correlations	18
---	----

Multiple Linear Regression

ORDER — rearrangement of intercorrelations.	19
MULTR — multiple regression and correlation	20

Polynomial Regression

GDATA — data generation.	21
----------------------------------	----

Canonical Correlation

CANOR — canonical correlation	22
NROOT — Eigenvalues and Eigenvectors of a special, nonsymmetric matrix	23

Analysis of Variance

AVDAT — data storage allocation	24
AVCAL — Σ and Δ operation	25
MEANQ — mean square operation	26

Discriminant Analysis

DMATX — means and dispersion matrix	27
DISCR — discriminant functions	28

Factor Analysis

TRACE — cumulative percentage of Eigenvalues	29
LOAD — factor loading	30
VARMX — varimax rotation	31

Time Series

AUTO — autocovariances	32
CROSS — crosscovariances	33
SMO — application of filter coefficients (weights)	34
EXSMO — triple exponential smoothing	35

Nonparametric Statistics

CHISQ — χ^2 test for a contingency table	36
UTEST — Mann-Whitney U test	37
TWOAV — Friedman two-way analysis of variance	38
QTEST — Cochran Q test	39
SRANK — Spearman rank correlation	40
KRANK — Kendall rank correlation	41
WTEST — Kendall coefficient of concordance	42
RANK — rank observations	43
TIE — calculation of ties in ranked observations	44

Random Number Generators

RANDU — uniform random numbers	45
GAUSS — normal random numbers	46

Special Matrix Operations

MINV — matrix inversion.	47
EIGEN — Eigenvalues and Eigenvectors of a real, symmetric matrix	48
SIMQ — solution of simultaneous linear, algebraic equations	49

Matrices

GMADD — add two general matrices.	50
GMSUB — subtract two general matrices.	51
GMprd — product of two general matrices.	52
GMTRA — transpose a general matrix	53
GTPRD — transpose product of two general matrices	54
MADD — add two matrices	55
MSUB — subtract two matrices	56
MPRD — matrix product (row into column)	57
MTRA — transpose a matrix	58
TPRD — transpose product	59
MATA — transpose product of matrix by itself	60
SADD — add scalar to matrix	61
SSUB — subtract scalar from a matrix.	62
SMPY — matrix multiplied by a scalar	63
SDIV — matrix divided by scalar	64
RADD — add row of one matrix to row of another matrix.	65
CADD — add column of one matrix to column of another matrix	66
SRMA — scalar multiply row and add to another row.	67
SCMA — scalar multiply column and add to another column.	68
RINT — interchange two rows.	69
CINT — interchange two columns	70
RSUM — sum the rows of a matrix.	71
CSUM — sum the columns of a matrix	72
RTAB — tabulate the rows of a matrix	73
CTAB — tabulate the columns of a matrix	74
RSRT — sort matrix rows	75
CSRT — sort matrix columns	76
RCUT — partition row-wise.	77
CCUT — partition column-wise	78
RTIE — adjoin two matrices row-wise	79
CTIE — adjoin two matrices column-wise	80
MCPY — matrix copy	81
XCPY — copy submatrix from given matrix.	82

RCPY — copy row of matrix into vector	83
CCPY — copy column of matrix into vector	84
DCPY — copy diagonal of matrix into vector.	85
SCLA — matrix clear and add scalar	86
DCLA — replace diagonal with scalar.	87
MSTR — storage conversion.	88
MFUN — matrix transformation by a function	89
RECP — reciprocal function for MFUN	90
LOC — location in compressed-stored matrix	91
ARRAY — vector storage-double dimensioned storage conversion	92

Other Mathematics

Integration

QUADR — integral of tabulated function by quadrature	93
SMPSN — integral of given function by Simpson's rule	94
RK1 — integral of first-order differential equation by Runge-Kutta method	95
RK2 — tabulated integral of first-order differential equation by Runge-Kutta method	96
RK3 — tabulated integral of a system of six first-order differential equations by Runge-Kutta method.	97

Fourier Analysis

FORIF — Fourier analysis of a given function	98
FORIT — Fourier analysis of a tabulated function	99

Special Operations and Mathematical Functions

GAMMA — gamma function	100
LEGEN — Legendre polynomial	101
BESJ — J Bessel function	102
BESY — Y Bessel function.	103
BESI — I Bessel function	104
BESK — K Bessel function.	105
CELI1 — elliptic integral of the first kind.	106
CELI2 — elliptic integral of the second kind.	107
EXPI — exponential integral.	108

SICI — sine and cosine integral.	109
CS — Fresnel integrals	110

Roots of Nonlinear Equations

RTWIT — refine estimate of root by Wegstein's iteration . . .	111
RTMIT — determine root within a range by Mueller's iteration	112
RTNIT — refine estimate of root by Newton's iteration . . .	113

Roots of Polynomial

POLRT — real and complex roots of polynomial	114
--	-----

Polynomial Operations

PADD — add two polynomials	115
PADDM — multiply polynomial by constant and add to another polynomial	116
PCLA — replace one polynomial by another	117
PSUB — subtract one polynomial from another	118
PMPY — multiply two polynomials	119
PDIV — divide one polynomial by another	120
PQSD — quadratic synthetic division of a polynomial	121
PVAL — value of a polynomial	122
PVSUB — substitute variable of polynomial by another polynomial	123
PCLD — complete linear division.	124
PILD — evaluate polynomial and its first derivative	125
PDER — derivative of a polynomial	126
PINT — integral of a polynomial	127
PGCD — greatest common divisor of two polynomials. . . .	128
PNORM — normalize coefficient vector of polynomial. . . .	129

SUBROUTINE TALLY

SUBROUTINE BOUND

SUBROUTINE SUBST

SUBROUTINE ABSNT

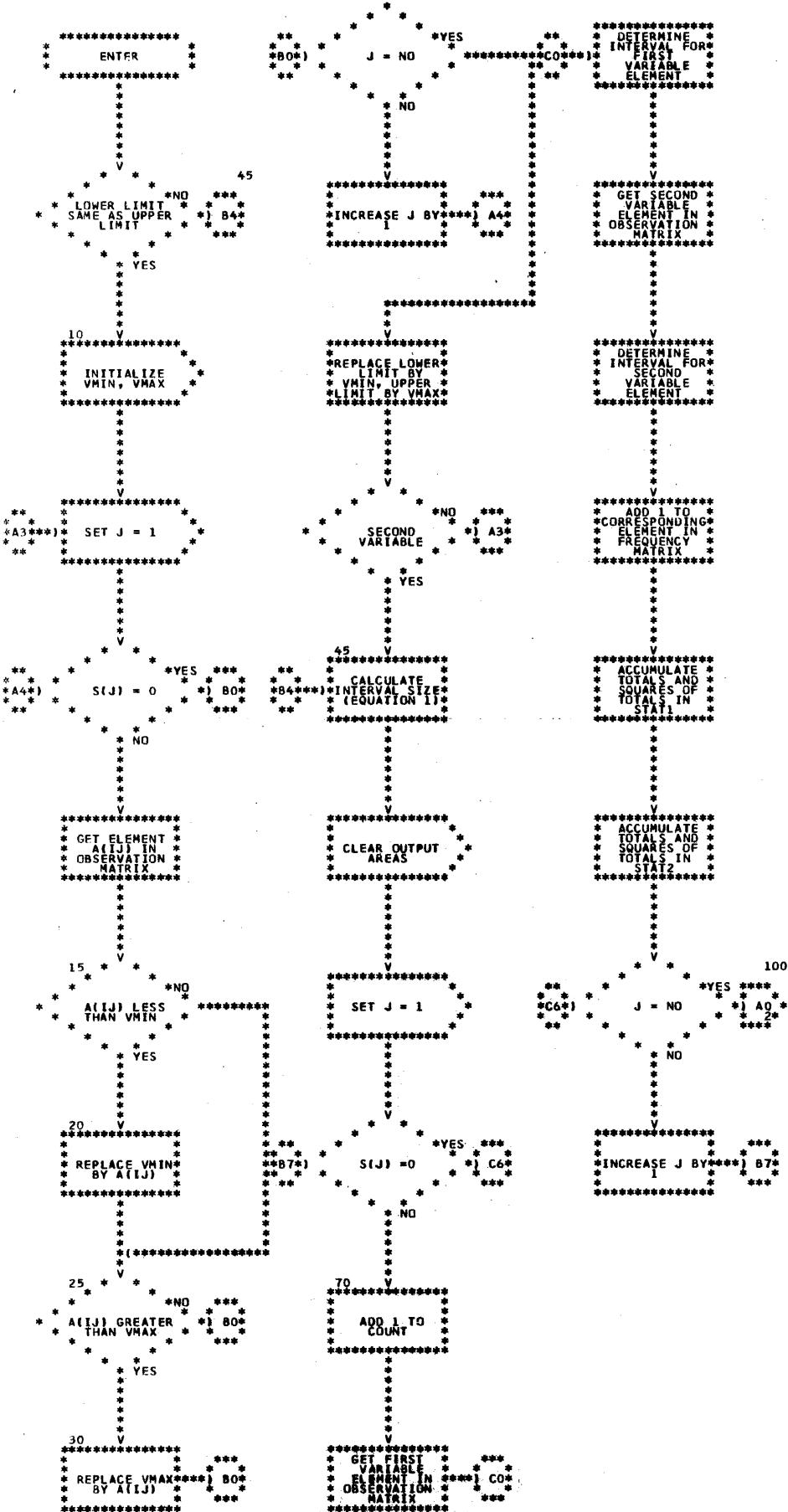
SUBROUTINE TABLE

```

    * **** ENTER ****
    * **** 60* * YES
    * **** J = NO ****
    * **** CO* * ACCUMULATE TOTALS AND SQUARES OF TOTALS (EQUATION 3)
    * **** V
    * **** INITIALIZE VMIN AND VMAX *
    * **** V
    * **** SET J = 1 *
    * **** V
    * **** YES *** 80* * J = NO
    * **** NO
    * **** V
    * **** INCREASE J BY*** A3* * DETERMINE INTERVAL FOR A(i,j)
    * **** I
    * **** V
    * **** LOWER LIMIT SAME AS UPPER LIMIT * NO
    * **** YES
    * **** V
    * **** 35 V
    * **** REPLACE LOWER LIMIT BY VMIN, UPPER LIMIT BY VMAX
    * **** V
    * **** GET ELEMENT A(i,j) IN OBSERVATION MATRIX
    * **** V
    * **** YES *** 80* * J = NO
    * **** NO
    * **** V
    * **** 10 V
    * **** A(i,j) LESS THAN VMIN * NO
    * **** YES
    * **** V
    * **** 15 V
    * **** REPLACE VMIN BY A(i,j)
    * **** V
    * **** 20 V
    * **** A(i,j) GREATER THAN VMAX * NO
    * **** YES
    * **** V
    * **** 25 V
    * **** REPLACE VMAX * 80* * BY A(i,j)
    * **** V
    * **** 55 V
    * **** ADD 1 TO COUNT * CO*
    * **** V
    * **** 70 V
    * **** ADD 1 TO INTERVAL IN FREQ VECTOR *
    * **** V
    * **** 80 V
    * **** CALCULATE RELATIVE FREQUENCIES * C5*** * (EQUATION 2)
    * **** V
    * **** 80 V
    * **** CALCULATE MEANS * (EQUATION 4)
    * **** V
    * **** 80 V
    * **** CALCULATE STANDARD DEVIATIONS * (EQUATION 5)
    * **** V
    * **** EXIT
    * **** V

```

SUBROUTINE TAB2



SUBROUTINE TAB2

```
100 ****
*** *   CALCULATE *
* AC **) * RELATIVE *
*** *   FREQUENCIES *
*   * (EQUATION 2)*
***** V ****
***** *
*   CALCULATE *
*   MEANS
* (EQUATION 3)*
***** V ****
***** *
*   CALCULATE *
*   STANDARD
*   DEVIATIONS
* (EQUATION 4)*
***** V ****
***** *
*   EXIT
*****
```

SUBROUTINE SUBMX

SUBROUTINE WOMEN

```

***** ENTER ****
***** B0* ***** NO
***** V ***** YES
***** 300 ***** V
***** CALC FOURTH
***** MOMENT EQ.(6) ****
***** V
***** EXIT ****
***** V
***** CALC NUMBER
***** OF FREQUENCY
***** INTERVALS
***** EQ.(1) ****
***** V
***** CALC SUM OF
***** ALL
***** FREQUENCIES
***** EQ.(2) ****
***** V
***** CALCULATE
***** FIRST MOMENT
***** EQ.(13) ****
***** V
***** IF OPTION =
***** 5, CALC
***** SECOND,
***** THIRD AND
***** FOURTH MOMENT
***** V
***** NO
***** OPTION = 2 ****
***** YES
***** V
***** 200 ***** V
***** CALC SECOND
***** MOMENT EQ.(4) ****
***** B2* ****
***** V
***** NO
***** OPTION = 3 ****
***** B0* ****
***** YES
***** V
***** 250 ***** V
***** CALC THIRD
***** MOMENT EQ.(5) ****
***** B2* ****

```

SUBROUTINE TTSTTT

```

160 *****
*      ** * CALC T-VALUE*
*      ** * AND DEGREES *****
*      ** * OF FREEDOM *
*      ** * EQ. (5)-(7) *
***** V 170 *****
*      ** * CALC T-VALUE*
*      ** * AND DEGREES *****
*      ** * OF FREEDOM *
*      ** * EQ. (6)-(9) *
***** V 180 *****
*      ** * CALC T-VALUE*
*      ** * AND DEGREES *****
*      ** * OF FREEDOM *
*      ** * EQ. (10)-(13) *
***** V 200 *****
*      * * YES
*      * * OPTION > 4 *****
*      * * B3* * EXIT
*      * * NO
***** V 180 *****
*      * * YES
*      * * OPTION = 4 *****
*      * * B2* *
*      * * NO
***** V 200 *****
*      * * YES
*      * * OPTION < 1 *****
*      * * B3* *
*      * * NO
***** V 125 *****
*      * * CALCULATE
*      * * VARIANCES EQ.
*      * * (2)
***** V 150 *****
*      * * NO
*      * * OPTION = 1 *****
*      * * YES
***** V 200 *****
*      * * CALC T-VALUE*
*      * * AND DEGREES *****
*      * * B3* *
*      * * OF FREEDOM *
*      * * EQ. (3)-(4) *
***** V 170 *****
*      * * NO
*      * * OPTION = 2 *****
*      * * B1* *
*      * * YES
*      * * B0* *
***** V 160 *****

```

SUBROUTINE CORRE

SUBROUTINE ORDER

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      COPY CORR  *  
* COEFF IN VS   *  
* YI IN RY(J)   *  
* FOR RET     *  
* VARIABLES    *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      COPY      *  
* INTER-CORR   *  
* COEFF OF     *  
*RETAINED IND.*  
*VARIABLES    *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      STORE      *  
* DEPENDENT    *  
* VARIABLE     *  
* NUMBER IN    *  
* ISAVE(K+1)    *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE MULTR

SUBROUTINE GDATA

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      GENERATE X *  
*      VARIABLES   *  
*      EQUATION (1)*  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*      CALCULATE   *  
*      MEANS       *  
*      EQUATION (2)*  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*      CALC Sums OF*  
*      CROSS-PRODUCTS*  
*      OF DEVIATIONS*  
*      EQ. (3)        *  
*****  
*  
*  
*  
*  
* V  
*****  
*      CALCULATE   *  
*      CORRELATION  *  
*      COEFFICIENTS.*  
*****  
*  
*  
*  
*  
* V  
*****  
*      CALCULATE   *  
*      STANDARD    *  
*      DEVIATIONS.  *  
*      EQ. (5)        *  
*****  
*  
*  
*  
*  
* V  
*****  
*      EXIT          *  
*****
```

SUBROUTINE CANON

```

***** ENTER ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** PARTITION ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CORRELATION ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (1) ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EIGENVALUES ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EIGENVECTORS ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** NRDOT ****
***** SET I = 1 ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CANONICAL ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CORRELATION ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (5) ****
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CHI-SQUARE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (6) ****
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CAE DEGREES ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** OF FREEDOM ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** FOR ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** CHI-SQUARE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (7) ****
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** LEFT HAND ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** COEFFICIENTS ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (8) ****
***** CALCULATE ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** LEFT HAND *** 60 ***
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** COEFFICIENTS *** 91 ***
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** EQUATION (9) ***
***** EXIT ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** ADD 1 TO I **** A5 ***
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** NO ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** YES ****
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** I=0 (I IS NUMBER OF CRITERIA) ****

```

SUBROUTINE NROOT

```

*****
*          ENTER
*****
*
*
*
*
*
*
*
V
*****
*FIGEN      *
***** ****
* EIGENVALUES *
* EIGENVECTOR *
*****
*
*
*
*
*
*
V
*****
* FORM RECIP. *
* RIF SQUAREROOT *
* EIGENVALUES *
* EQ. (1) *
*****
*
*
*
*
*
V
*****
* FORM      *
* B**(-1/2) *
*****
*
*
*
*
*
V
*****
* FORM      *
* SYMMETRIC *
* MATRIX S. EQ. *
* (21-3) *
*****
*
*
*
*
*
V
*****
*FIGEN      *
***** ****
* EIGENVALUES *
* EIGENVECTOR *
*****
*
*
*
*
*
*
V
*****
* NORMALIZE *
* EIGENVECTORS *
* EQ. (4-(1)-(5)) *
*****
*
*
*
*
*
V
*****
*          EXIT
*****

```

SUBROUTINE AVDAT

```

***** ENTER ****
***** EXIT ****
***** CALC SIZE OF DATA ARRAY X ****
***** SEE EQUATION (1) ****
***** MOVE DATA TO UPPER PART OF ARRAY X ****
***** CALC STEPS USED TO STORE DATA SEE EQUATION (2-3) ****
***** INITIALIZE KOUNT(J)=1 ****
***** SUBSCRIPTS FOR J=1,K ****
***** SET I=1 ****
***** CALC LOCATION ALL TO STORE A DATA SEE EQUATION (4) ****
***** REDISTRIBUTE I-TH DATA POINT TO X(L) ****
***** SUBTRACT 1 FROM N ****
***** N=0 NO B1 ****
***** YES ***** B0 ****
***** SET J=1 ****
***** KOUNT(J) = LEVEL(J) ****
***** ADD 1 TO KOUNT(J) ****
***** J=K YES ****
***** NO ****
***** ADD 1 TO J B2 ****
***** SET KOUNT(J)=1 B4 ****
***** ADD 1 TO I A6 ****

```

SUBROUTINE AVCAL

SUBROUTINE MEANQ

SUBROUTINE DMATX

```

***** * ****
*   ENTER   *   ** * CALC POOLED *
*          *   *BO***) * DISPERSION *
*          *   * MATRKA *
*          *   ** * EQUATION (3) *
***** * ****
*
*
*
*
*
V
*****
* INITIALIZE *   ****
* WORKING   *   * EXIT   *
* STORAGE   *   ****
*****
*
*
*
*
*
V
*****
* SET K = 1   *
*****
*
*
*
*
*
V
*****
**   ****
* SUM K-TH   *
* GROUP DATA *
*A3***) AND CALCULATE*
* MEANS
**   * EQUATION (1)*
*****
*
*
*
*
*
V
*****
* SUBTRACT *
* MEANS FROM *
* DATA
*****
*
*
*
*
*
V
*****
* CALC SUM OF *
*CROSS PRODUCT*
*OF DEVIATIONS*
* EQUATION (2)*
*****
*
*
*
*
*
V
*****
* ACCUMULATE *
* SUM OF CROSS*
* PRODUCTION OF *
* DEVIATIONS *
*****
*
*
*
*
*
V
*****
* * * YES *** *
* * * K = LAST GROUP * * * BO* *
* * * NO    * * *
*****
*
*
*
*
*
V
*****
* * * ADD 1 TO K *** A3*
* * * *** *
*****

```

SUBROUTINE DISCR

```

***** * FOR EACH CASE *
* IN EACH GROUP *
* DO THE *
* FOLLOWING *
* STEPS *
*****
*
*
*
*
*
V
*****
* CALCULATE *
* OTHER MEANS *
*FOR VARIABLES*
* EQUATION {1}*
*****
*
*
*
*
*
V
*****
* CALCULATE *
* GENERAL *
*MAHALANOBIS D*
*FOR VARIABLES*
* EQUATION {2}*
*****
*
*
*
*
*
V
*****
* CALCULATE *
* DEGREES OF *
* FREEDOM *
*****
*
*
*
*
*
V
*****
* SET K = 1 *
*****
*
*
*
*
*
V
*****
** * CALC *
* COEFFICIENTS *
* FOR DISCRIMINANT *
* FUNCTION *
* EQ.(3) *
*****
*
*
*
*
*
V
*****
* CALCULATE *
* CONSTANT *
* EQUATION (4) *
*****
*
*
*
*
*
V
*****
* YES ***
* K = LAST * *) BO*
* FUNCTION * ***
*
*
*
*
NO
*
*
*
*
*
V
*****
* ADD 1 TO K ****) A5*
* *
* *
* *
* *
V

```

SUBROUTINE TRACE

```
*****  
*      ENTER      *  
*****  
  
*  
*  
*  
*  
*  
* V  
*****  
* SET I = 0 AND *  
* K = 0           *  
*****  
  
*  
*  
*  
*  
* V  
** * * * * NO  
*A2*) * * * * I-TH * *****  
* EIGENVALUE *  
* LESS THAN   *  
* CONSTANT    *  
* * * * * YES  
* * * * *  
* V  
*****  
* CALCULATE *  
* DIVIDE I-TH *  
* PERCENTAGE OF *  
* K EIGENVALUES *  
*****  
  
*  
*  
*  
*  
* V  
*****  
* EXIT          *  
*****  
  
*****  
*  
* 105  V  
*****  
*  
* ADD 1 TO K *  
*  
*  
* V  
*****  
* DIVIDE I-TH *  
* EIGENVALUE BY *  
* N.           *  
*  
*  
* V  
*****  
*  
*  
*  
* ADD 1 TO I ****) A2*  
*  
* ***  
*****
```

SUBROUTINE LOAD

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* SET I=1 *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
** CALCULATE **  
*SQUARE ROOT OF *  
*A2*** * TH *  
* * EIGENVALUE *  
* * CALL IT SQ *  
*****  
*  
*  
*  
*  
* V  
*****  
*MULTIPLY I-TH*  
* EIGENVECTOR *  
* BY SQ *  
*****  
*  
*  
*  
* V  
*****  
* * * * * NO *  
* * * * * I=K *  
* * * * * * * *  
* * * * * YES *  
* * * * *  
* * * V  
*****  
* EXIT *  
*****  
*  
*  
* V  
*****  
* * * * * ***  
* ADD 1 TO I *** A2*  
* * * * * ***  
*****
```

SUBROUTINE VARMX

SUBROUTINE AUTO

SUBROUTINE CROSS

SUBROUTINE SMO

SUBROUTINE EXSMO

SUBROUTINE CHISO

SUBROUTINE UTEST

```

***** * TIE * ****
* * * COMPUTE SUM *
* * * OF T. (EQ-3) *
* * *
* * V *
***** * NO * ****
* * * SUM OF T = 0 *
* * *
* * YES *
* * *
* * V *
***** * 60 * ****
* * *
* * COMPUTE *
* * STANDARD *
* * DEVIATION *
* * (EQUATION 4) *
***** * 50 * ****
* * *
* * COMPUTE *
* * STANDARD *
* * DEVIATION *
* * (EQUATION 5) *
***** * 70 * ****
* * *
* * CALCULATE Z *
* * (EQUATION 6) *
***** * V *
***** * EXIT * ****
* * *
* * U PRIME LESS * NO *
* * THAN U * *
* * * YES *
* * *
* * V *
***** * 20 * ****
* * *
* * SET U = U *
* * PRIME *
***** * V *
***** * 30 * ****
* * *
* * N2 LESS THAN * NO * ***
* * 20 * * } BO *
* * * YES *
* * *
* * V *
***** * V *
* * *
* * SET Z = 0 *
* * *
***** * 80 * ****
* * *
* * V *
***** * EXIT * ****

```

SUBROUTINE TWOAV

```

***** * ****
* * ENTER * *
***** * ****
***** * ****
* * DATA RANKED * *
***** * ****
* * NO * *
***** * ****
* * YES * *
***** * ****
* * 10 * *
***** * ****
* * INITIALIZE * *
* * ROW INDEX * *
***** * ****
***** * ****
* * A3***) GET ROW IN * *
* * MATRIX A * *
***** * ****
***** * ****
* * MOVE ROW TO * *
* * WORK VECTOR * *
***** * ****
***** * ****
* * V * *
***** * ****
* * RANK * *
***** * ****
* * RANK DATA IN * *
* * ROW * *
***** * ****
***** * ****
* * V * *
***** * ****
* * MOVE RANKED * *
* * ROW TO MATRIX * *
* * A * *
***** * ****
***** * ****
* * V * *
***** * ****
* * LAST ROW * *
***** * ****
* * NO * *
***** * ****
* * A3* * *
***** * ****
* * YES * *
***** * ****
***** * ****
* * INITIALIZE * *
* * COLUMN INDEX * *
***** * ****
***** * ****
* * A9***) GET COLUMN IN * *
* * MATRIX R * *
***** * ****
***** * ****

```

SUBROUTINE QTEST

SUBROUTINE SRANK

SUBROUTINE KRANK

SUBROUTINE WTEST

SUBROUTINE RANK

```

***** ENTER ****
*
*
*
*
*
*
*
V
*****
* INITIALIZE *
* CONSTANTS AND *
* INDICATOR *
*****
*
*
*
*
*
*
V
*****
** * FIND EACH *
<A2***> * SMALLEST *
** * ELEMENT *
*****
*
*
*
*
*
V
*****
* COUNT NUMBER *
* OF SMALLER *
* DATA POINTS *
*****
*
*
*
*
*
V
*****
* COUNT NUMBER *
* OF EQUAL DATA *
* POINTS *
*****
*
*
*
*
*
V
*****
* IS THERE A *
* TIE *
* YES *
*
*
*
*
*
NO
*
*
*
*
60 V
*****
* STORE RANK *
*****
*
*
*
*
V
*****
70 V
*****
* CALCULATE *
* AVERAGE RANKS *
* AND STORE *
*****
*
*
*
*
V
*****
*
*
*
*
*
V
*****
*
*
*
*
*
V
*****
* LAST ELEMENT * ) A2*
*
*
*
*
NO
*
*
YES
*
*
V
*****
*
*
*
*
V
*****
*
*
*
*
V
*****
* EXIT *
*
*

```

SUBROUTINE TIE

SUBROUTINE RANDU

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* COMPUTE IY *  
* USING INPUT *  
* IX *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* CONVERT *  
* RESULT TO *  
* FLOATING PT *  
* IN RANGE 0.1*  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE GAUSS

```

***** * ***** * ***** * *****
*          ENTER          *
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          V          *
***** * ***** * ***** * *****
*          COMPUTE 12          *
*          *          *
*          UNIFORM          *
*          *          *
*          RANDOM          *
*          *          *
*          NUMBERS          *
***** * ***** * ***** * *****
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          V          *
***** * ***** * ***** * *****
*          ADD 12 RANDOM          *
*          *          *
*          NUMBERS          *
*          *          *
*          TOGETHER          *
***** * ***** * ***** * *****
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          V          *
***** * ***** * ***** * *****
*          COMPUTE Y          *
*          *          *
*          (EQUATION 2)          *
***** * ***** * ***** * *****
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          *          *
*          V          *
***** * ***** * ***** * *****
*          EXIT          *
***** * ***** * ***** * *****

```

SUBROUTINE MINV

SUBROUTINE EIGEN

SUBROUTINE SIMQ

SUBROUTINE GMADD

```
* *****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
* CALCULATE *  
* NUMBER OF *  
* ELEMENTS *  
*****  
*  
*  
*  
*  
*  
* 10 V  
*****  
*  
* ADD MATRICES*  
* R=A+B *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE GMSUB

```
*****  
*      ENTER      *  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
*      V  
*****  
*      CALCULATE   *  
*      NUMBER OF    *  
*      ELEMENTS    *  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
10   V  
*****  
*      SUBTRACT    *  
*      MATRICES    *  
*      R=A-B        *  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
*      V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE GMRD

SUBROUTINE GMTRA

```
*****  
* FNTFR *  
*****  
  
V  
*****  
* INITIALIZE *  
* POINTER AT *  
* FIRST ELEMENT *  
* OF OUTPUT *  
* MATRIX *  
*****  
  
*  
*  
*  
*  
*  
* V  
*****  
** *  
*A2***) GET ELEMENT *  
* FROM A *  
** *  
*****  
  
*  
*  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
* MOVE ELEMENT *  
* TO R *  
*  
*****  
  
*  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* * * * NO * * *  
* * LAST ELEMENT * * * A2 *  
* * * * YES * * *  
*  
*  
*  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE GTPRD

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* INITIALIZE *  
* DATA TYPE *  
* FIRST ELEMENT *  
* OF OUTPUT *  
* MATRIX *  
*****  
*  
*  
*  
*  
* V  
*****  
** *  
* A2***) * GET ELEMENTS*  
* * * FROM A AND B*  
** *  
*****  
*  
*  
*  
*  
*  
* 10 V  
*****  
* MULTIPLY *  
* TRANSPOSE OF*  
* A BY B *  
*****  
*  
*  
*  
*  
* V  
*****  
* * * * * NO  
* * LAST ELEMENT * * *  
* * * * * YES  
* *  
* V  
*****  
* EXIT *  
*****  
*  
*  
* V  
*****  
* STEP POINTER* ***  
* TO NEXT * *** A2*  
* ELEMENT OF *  
*OUTPUT MATRIX* ***
```

SUBROUTINE MADD

SUBROUTINE MSUB

SUBROUTINE MPRD

SUBROUTINE MTRA

```

***** *****
* FENTER *
*****
*          *
*          *
*          *
*          *
*          V
*          *
*          * YES
*          * STORAGE MODE *****
*          * GENERAL   *
*          *          *
*          * NO
*          *
*          *
*          *
*          *
*          10 V
***** *****
* MCPY   *
***** *****
* COPY MATRIX *
*          *
*****
*          *
*          *
*          *
*          *
*          V
***** *****
* EXIT   *
*****
***** *****
*          *
*          *
*          *
*          *
*          *
*          V
*****
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          20 V
***** *****
* INITIALIZE *
* POINTER AT   *
* FIRST ELEMENT *
* ON OUTPUT    *
* MATRIX      *
*****
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          V
***** *****
** * * * * GET ELEMENT *
* * * * * FROM A *
* * * * *
*****
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          30 V
***** *****
* MOVE ELEMENT*
* TO R        *
*          *
*****
*          *
*          *
*          *
*          *
*          V
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          *
*          * NO
*          * * * *
*          * LAST ELEMENT * *) A5*
*          * * * *
*          * YES
*          *
*          *
*          *
*          *
*          V
***** *****
* EXIT   *
*****

```

SUBROUTINE TPRD

SUBROUTINE MATA

SUBROUTINE SADD

```
*****  
*      ENTER      *  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      V  
*****  
*      *  
*      COMPUTE    *  
*      VECTOR LENGTH*  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      1      V  
*****  
*      *  
*      ADD SCALAR TO*  
*      EACH ELEMENT*  
*****  
*      *  
*      *  
*      *  
*      *  
*      *  
*      V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE SSUB

```
* *****  
* : ENTER : *  
* *****  
* :  
* :  
* :  
* :  
* :  
* :  
* : V  
* *****  
* :  
* : COMPUTE  
* : VECTOR LENGTH  
* :  
* *****  
* :  
* :  
* :  
* :  
* :  
* : L  
* : V  
* *****  
* : SUBTRACT : *  
* : SCALAR FROM : *  
* : EACH ELEMENT : *  
* *****  
* :  
* :  
* :  
* :  
* :  
* :  
* : V  
* *****  
* : EXIT : *  
* *****
```

SUBROUTINE SMPY

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
* COMPUTE      *  
* VECTOR LENGTH *  
*****  
*  
*  
*  
*  
*  
*  
*  
* 1  V  
*****  
*  
* MULTIPLY EACH *  
* ELEMENT BY    *  
* SCALAR        *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE SDIV

```
*****  
* FENTER *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
* COMPUTE  
* VECTOR LENGTH  
*  
*****  
*  
*  
*  
* V  
* * * * * NO  
* * SCALAR IS * *  
* * ZERO * *  
* *  
* * * * * YES  
* *  
* *  
* *  
* *  
* *  
* 1  
* V  
*****  
*  
* SET VECTOR *  
* LENGTH TO 1 *  
*  
*****  
*  
*  
*  
*  
* (*****  
*  
*  
* 3  
* V  
*****  
*  
* DIVIDE *  
* ELEMENTS BY *  
* SCALAR *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE RAOD

```
*****  
* ENTER *  
*****  
  
*****  
* INITIALIZE *  
* POINTER AT *  
* FIRST ELEMENT *  
* OF *  
*****  
  
V  
*****  
** * GET ELEMENT *  
* A2 **) * FROM INPUT *  
** * MATRIX *  
** *  
*****  
  
1 V  
*****  
* ADD ELEMENT *  
* TO OUTPUT *  
* MATRIX *  
*****  
  
V  
*****  
* * * YES  
* * LAST ELEMENT *  
* * OF ROW *  
* * *  
* * NO  
* *  
* *  
* *  
* V  
*****  
* STEP POINTER * ***  
* TO EXIT * A2 *  
* ELEMENT OF * ***  
* ROW * ***  
*****  
  
V  
*****  
* EXIT *  
*****
```

SUBROUTINE CADD

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* INITIALIZE *  
* POINTER AT *  
* FIRST ELEMENT *  
* OF COLUMN *  
*****  
*  
*  
*  
*  
* V  
*****  
** * GET ELEMENT *  
* A2 ***) * FROM INPUT *  
* * MATRIX *  
** *  
*****  
*  
*  
*  
*  
*  
* 1 V  
*****  
* ADD ELEMENT *  
* TO OUTPUT *  
* MATRIX *  
*****  
*  
*  
*  
*  
* V  
* * * * YES  
* * LAST ELEMENT * ****  
* * OF COLUMN * *  
* * * NO  
* *  
* *  
* *  
* V  
*****  
* STEP POINTER * ***  
* TO NEXT ****) A2*  
* ELEMENT OF * * A2*  
* COLUMN * ***  
*****  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE SRMA

SUBROUTINE SCHA

```

***** *****
*      ENTER      *
***** *****
*
*
*
*
*
*
V
*****
*      SET I = 1      *
*****
*
*
*
*
*
*
V
*****
**      * GET ELEMENTS *
* A2**) * IN BOTH *
*      * COLUMNS *
**      * *****
*
*
*
*
*
V
*****
*      *      *      * NO
*      LB = 0      * *****
*
*
*
*
*
YES
*
*
*
*
?
V
*****
*      *      *      * NO
*      LB = 0      * *****
*
*
*
*
*
YES
*
*
*
*
*
V
*****
*      *      *      * NO
*      I = N      * *****
*
*
*
*
*
*
V
*****
*      *      *      * NO
*      EXIT      * *****
*
*
*
*
V
*****
*      *      *      * *** *
*      INCREASE I BY    * A2*
*      I      *      * *** *
*****

```

SUBROUTINE RINT

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* INITIALIZE *  
* PTRNTR OF *  
* FIRST ELEMENT *  
* OF ROW *  
*****  
*  
*  
*  
*  
* V  
*****  
** * * * GET ELEMENT *  
* A2***) * IN BOTH ROWS*  
** * * *  
*****  
*  
*  
*  
*  
* V  
*****  
* INTERCHANGE *  
* ELEMENTS *  
*****  
*  
*  
*  
*  
* V  
*****  
* * * * YES  
* LAST ELEMENT *  
* * * *  
* * * * NO  
* * * *  
* *  
* V  
*****  
* STEP PTRNTR *  
* TO NEXT ***} A2*  
* ELEMENT OF *  
* ROW *  
*****  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE CINT

SUBROUTINE RSUM

```

*****
* ENTR *
*****
*
*
*
*
*
*
*
V
*****
* SET J=1
*****
*
*
*
*
*
*
V
*****
* SPT J=1
*****
*
*
*
*
*
*
V
*****
* GET ELEMENT *
* IJ IN INPUT *
* MATRIX *
*****
*
*
*
*
*
*
V
*****
* ACCUMULATE *
* SUM IN OUTPUT *
* VFCCTR *
*****
*
*
*
*
*
*
V
*****
* YES
* J = M
* NO
*****
*
*
*
*
*
V
*****
* INCREASE **** A3 *
* J BY 1
*****
*
*
*
*
*
V
*****
* YES
* I = N
* NO
*****
*
*
*
*
*
V
*****
* INCREASE **** A3 *
* I BY 1
*****
*
*
*
*
*
V
*****
* EXIT
*****

```

SUBROUTINE CSUM

```

***** ENTER ****
*
*
*
*
*
*
*
*
V
*****
* SET J=1 *
*
*
*
*
*
*
V
*****
* SET I=1 *
*
*
*
*
*
V
*****
* GET ELEMENT *
* IJ IN INPUT *
* MATRIX *
*****
*
*
*
*
*
*
V
*****
* ACCUMULATE *
* SUM IN OUTPUT *
* VECTOR *
*****
*
*
*
*
*
V
*****
* YES *
* I = N *
*
*
*
*
*
*
NO
*
*
*
*
*
V
*****
* *** A3 ***
* INCREASE ****) A3 *
* I BY I *
*****
*
*
*
*
V
*****
* YES *
* J = M *
*
*
*
*
*
NO
*
*
*
*
*
V
*****
* *** A3 ***
* INCREASE ****) A3 *
* J BY I *
*****
*
*
*
*
V
*****
* EXIT *
*****

```

SUBROUTINE RTAB

SUBROUTINE CTAB

```

***** ENTER ****
*
*
*
*
*
*
10 V
*****
* * CLEAR_OUTPUT *
* AREAS R AND S *
*****
*
*
*
*
*
*
V
*****
* * SET I = 1 *
*****
*
*
*
*
*
*
V
*****
* * * * * NO
* * * * * OUTSIDE *
* * * * * RANGE *
* * * * * YES
* * * * *
*
50 V
*****
* * ADD 1 TO S(L+1)
* * * * *
*****
*
*
*
*
*
*
V
*****
* * ADD 1 TO S *
* * VECTOR *
*****
*
*
*
*
*
*
V
*****
* * * * * NO
* * * * * I = M *
* * * * * YES
* * * * *
*
V
*****
* * EXIT *
*****

```

SUBROUTINE RSRT

SUBROUTINE CSRT

SUBROUTINE RCUT

SUBROUTINE CCUT

SUBROUTINE RTIE

SUBROUTINE CTIE

```

***** ENTER ****
*
*
*
*
*
*
*
V
*****
* SFT. STORAGE *
* MODE AND *
* NUMBER OF *
* COLUMNS FOR A *
*****
*
*
*
*
*
V
*****
* INITIALIZE *
* ARRAY POINTER *
* AT FIRST *
ELEMENT
*****
*
*
*
*
*
V
*****
* A2**) ) GET ELEMENT *
*****
*
*
*
*
*
V
*****
?
V
*****
* MOVE ELEMENT *
* TO MATRIX R *
*****
*
*
*
*
V
*****
*
*
*
*
*
V
*****
* YES
* LAST ELEMENT *
*
*
*
*
*
NO
*
*
*
*
*
V
*****
* STEP POINTER * *** A3*
* TO NEXT *** ) A3*
* ELEMENT * ***
*****
*
*
*
*
V
*****
*
*
*
*
V
*****
* NO
* BOTH INPUT *
* MATRICES *
* PROCESSED *
*
*
*
YES
*
*
*
*
V
*****
*
*
*
*
V
*****
* EXIT *
*****
*
*
V
*****
* RESET STORAGE * ***
* MODE AND * ) A2*
* NUMBER OF *
* COLUMNS FOR B * ***
*****

```

SUBROUTINE MCPY

```
*****  
* ENTFR *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
*  
* COMPUTE  
* VECTOR LENGTH  
*  
*****  
*  
*  
*  
*  
*  
*  
*  
*  
* 1 V  
*****  
* COPY MATRIX *  
* ELEMENT BY *  
* ELEMENT *  
*  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE XCPY

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
* COMPUTE  
* LIMITS L2,K2*  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*  
* INITIALIZE  
* ARRAY POINTER  
* AT L2,K1  
* MATRIX A  
*****  
*  
*  
*  
*  
* V  
*****  
*  
* A3*)*) GET ELEMENT *  
*  
*  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*  
* MOVE ELEMENTS*  
* TO MATRIX R *  
*  
*****  
*  
*  
*  
*  
* V  
*****  
* * * * * NO  
* * LAST ELEMENT *  
* * (L2,K2) * *  
* * * * YES  
* *  
* *  
* *  
* V  
*****  
*  
* EXIT  
*  
*****  
*  
*  
* V  
*****  
*  
* STEP POINTERS ***  
* TO NEXT ****] A3*  
* ELEMENT * * ***  
*****
```

SUBROUTINE RCPY

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* INITIALIZE *  
* POINTER AT *  
* FIRST ELEMENT *  
* OF ROW *  
*****  
*  
*  
*  
*  
* V  
*****  
** * GET ELEMENT *  
* A2***) FROM INPUT *  
* MATRIX *  
** *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* MOVE ELEMENT *  
* TO OUTPUT *  
* VECTOR *  
*  
*****  
*  
*  
*  
*  
* V  
*****  
* * * * YES  
* * LAST ELEMENT *  
* * OF ROW *  
* * * *  
* * * NO  
* *  
* *  
* *  
* * V  
*****  
* STEP POINTER* *** *  
* TO NEXT ***) A2* *  
* ELEMENT OF * * A2* *  
* ROW * *** *  
*****  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE CCPY

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      INITIALIZE   *  
*      POINTER AT    *  
* FIRST ELEMENT     *  
* OF COLUMN         *  
*****  
*  
*  
*  
*  
* V  
*****  
**      GET ELEMENT  *  
* A2**) FROM INPUT  *  
* MATRIX             *  
**  
*****  
*  
*  
*  
*  
*  
* V  
*****  
1  
*  
*  
*  
*  
*  
* V  
*****  
*      MOVE ELEMENT  *  
* TO OUTPUT          *  
* VECTOR              *  
*  
*****  
*  
*  
*  
*  
* V  
*****  
*      LAST ELEMENT  * YES  
* OF COLUMN          *  
*  
*      NO  
*  
*  
*  
*  
* V  
*****  
* STEP POINTER        ***  
* TO NEXT  ****) A2*  
* ELEMENT OF          *  
* COLUMN              *  
*****  
*  
*  
*****  
*  
*  
* V  
*****  
*      EXIT           *  
*****
```

SUBROUTINE DCPY

```
* *****  
*      ENTER      *  
* *****  
*  
*  
*  
*  
*  
*  
* V  
* *****  
*      INITIALIZE   *  
*      POINTER AT    *  
*      FIRST ELEMENT  *  
* *****  
*  
*  
*  
*  
*  
* V  
* *****  
** * * GET DIAGONAL*  
* A2*) ELEMENT FROM*  
* INPUT MATRIX*  
* *****  
*  
*  
*  
*  
*  
* V  
* *****  
* MOVE ELEMENT*  
* TO OUTPUT*  
* VECTOR*  
* *****  
*  
*  
*  
*  
* V  
* *****  
* * * * YES  
* * LAST ELEMENT * *****  
* * OF DIAGONAL * *  
* * * * NO  
* *  
* *  
* *  
* *  
* V  
* *****  
* * STEP POINTER* *** *  
* * TO NEXT ****) A2* *  
* * ELEMENT OF * * * *  
* * DIAGONAL * * * *  
* *****  
*  
*  
* V  
* *****  
* EXIT      *  
* *****
```

SUBROUTINE SCLA

```

***** ENTR ****
*
*
*
*
*
*
*
*
V
*****
* COMPUTE *
* VECTOR LENGTH *
*****
I
*****
REPLACE BY *
RECALLER
ELEMENT BY *
ELEMENT
*****
V
*****
EXIT
*****

```

SUBROUTINE DCLA

```

***** *
*      ENTER      *
*      ****
*      *
*      *
*      *
*      *
*      *
*      V
***** *
*      INITIALIZE   *
*      POINTER AT    *
* FIRST ELEMENT   *
*      ****
*      *
*      *
*      *
*      *
*      *
*      V
***** *
*      ****
*      * A2 ***) * GET DIAGONAL *
*      *      ELEMENT *
*      *      *
*      ****
*      *
*      *
*      *
*      *
*      *
*      *
*      V
***** *
*      REPLACE      *
*      ORIGINAL     *
* ELEMENT WITH   *
* SCALAR        *
*      ****
*      *
*      *
*      *
*      *
*      V
***** *
*      *      * YES
*      * LAST ELEMENT *****
*      * IF DIAGONAL   *
*      *      *
*      *      *
*      * NO
*      *      *
*      *      *
*      *      *
*      V
***** *
*      *      *** *
* STEP POINTER   * * * *
*      * NEXT   * * * * A2 * *
* ELEMENT OF   * * * *   *
*      * DIAGONAL * * * *   *
***** *
*      *
*      *
*      V
***** *
*      EXIT      *
*      ****

```

SUBROUTINE MSTR

```

***** ENTER ****
*
*
*
*
*
*
*
V
*****
* INITIALIZE *
* ARRAY PATTERN *
* AT FIRST *
* ELEMENT *
*****
*
*
*
*
V
*****
* A2***) * GET ELEMENT *
* *
* *
* *
* *
* V
*****
* YES
* OUTPUT MATRIX *
* GENERAL *
* *
* *
* NO
* *
* *
* V
*****
* YES
* ELEMENT IN *
* LOWER *
* TRIANGLE *
* *
* *
* NO
* *
* *
* V
*****
* YES
* OFF-DIAG *
* ELEMENT OF *
* DIAGONAL *
* OUTPUT MATRIX *
* *
* NO
* *
* *
* V
*****
* FORM ELEMENT *
* OF OUTPUT *
* MATRIX *
*****
*
*
*
V
*****
* NO
* LAST ELEMENT *
* *
* *
* YES
* *
* *
* V
*****
* EXIT
* *
*****
* STEP POINTER *
* TO NEW *
* ELEMENT *
* A2*
* *
*****

```

SUBROUTINE MFUN

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      COMPUTE      *  
* VECTOR LENGTH *  
*****  
*  
*  
*  
*  
*  
*  
*  
* 5   V  
*****  
*      APPLY      *  
* FUNCTION TO *  
* MATRIX      *  
* ELEMENT BY *  
* ELEMENT      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

FUNCTION RECP

SUBROUTINE LOC

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
* V  
* * * * * NO  
* * MS LESS THAN *  
* * 1             *  
* * * * * YES  
* *  
* *  
* *  
* * 10            V  
*****  
* COMPUTE      *  
* SUBSCRIPT FOR *  
* GENERAL      *  
* MATRIX       *  
*****  
*  
*  
*  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* * * * * NO  
* * MS EQUAL TO 1 *  
* * * * * YES  
* *  
* *  
* *  
* *  
* *  
* * V  
*****  
* COMPUTE      *  
* SUBSCRIPT FOR * A6 *  
* SYMMETRIC    *  
* MATRIX       *  
*****  
*  
*  
*  
*  
*  
*  
*  
*  
* 30            V  
*****  
* COMPUTE      *  
* SUBSCRIPT FOR *  
* GENERAL      *  
* MATRIX       *  
*****  
*  
*  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
* * * * * EXIT  
* A6 * *  
* * * * *
```

SUBROUTINE ARRAY

SUBROUTINE QUADR

```

***** * ****
*   ENTER    *   ** * ****
*   ***** *   *B0***) ADD RESULT TO *
*   ***** *   * SUM
*   ***** *
*   V
***** * ****
*   * ****
*   SET SUM EQUAL *
*   TO 0
*   ***** *
***** *
*   V
***** *
*   * ****
*   ARE THERE 2 NO
*   LESS THAN TWO FUNCTIONAL * ****
*   VALUES *   *B2*) ARE THERE 4 NO
*   * ****
*   YES
*   V
32 ***** *
*   * ****
*   SET ERROR CODE TO 1
*   ***** *
***** *
*   V
***** *
*   EXIT    *   * ****
*   ***** *
***** *
*   V
***** *
*   * ****
*   DOES THE NO
*   SPACING IF FUNCTIONAL * ****
*   VALUES EQUAL *   * ZERO *
*   * ****
*   YES
*   V
35 ***** *
*   * ****
*   SET ERROR CODE TO 2
*   ***** *
***** *
*   V
***** *
*   EXIT    *   * ****
*   ***** *
***** *
*   V
***** *
*   * ****
*   ARE THERE 3 NO
*   POINTS LEFT * ****
*   * ****
*   YES
*   V
48 ***** *
*   * ****
*   USE EQ.(2) *
*   AND ADD TO *
*   SUM
*   ***** *
***** *
*   V
***** *
*   * ****
*   ARE THERE 2 NO
*   POINTS LEFT * ****
*   * ****
*   YES
*   V
***** *
*   * ****
*   ARE THERE 1 NO
*   POINTS LEFT * ****
*   * ****
*   YES
*   V
***** *
*   * ****
*   ARE THERE 5 YES ***
*   LESS THAN 5 ) B2*
*   FIVE POINTS *   * REMAINING *
*   * ****
*   NO
*   V
39 ***** *
*   * ****
*   USE 5 POINT *** * ****
*   FORMULA *** ) B0*
*   EQ.(1) *** * ****
*   * ****
***** *

```

SUBROUTINE SMPSN

SUBROUTINE RK1

SUBROUTINE RK2

SUBROUTINE RK3

SUBROUTINE FORIP

SUBROUTINE FOR IT

SUBROUTINE GAMMA

SUBROUTINE LEGEND

SUBROUTINE BESJ

SUBROUTINE BESY

SUBROUTINE BESI

SUBROUTINE BESK

SUBROUTINE CBLII

```

***** *****
*      *
*      ENTER      *
***** *****
*      *
*      *
*      *
*      *
*      V
***** *****
*      *
*      IER=0      *
*      *
***** *****
*      *
*      *
*      *
*      *
*      V
*      *      *
*      *      YES      5
*      IS CK LESS      *RESULT EQUALS*
*      THAN OR EQUAL    *****) * INFINITY. *
*      TO 1.0           *      IER=1      *
*      *
*      *
*      NO      *
*      *
*      *
*      *
*      *
*      V
*      *      *
*      INITIALIZE      *      *****
*      ARITHMETIC      *      EXIT      *
*      AND GEOMETRIC   *      *****
*      MEANS ARI,GEO   *
***** *****
*      *
*      *
*      *
*      *
*      *
*      30      V
***** *****
*      *      SET AARI=ARI      *
*      *      COMPUTE      *
*      *      ARI=AARI+GEO      *
***** *****
*      *
*      *
*      *
*      *
*      V
*      *      *
*      *      YES      50
*      IS ARI-GEO      *RESULT EQUALS*
*      NEGIGIBLE      *      PI/ARI      *
***** *****
*      *
*      *
*      NO      *
*      *
*      *
*      *
*      *
*      V
*      *      *
*      COMPUTE NEW      *      *****
*      GEOMETRIC      *      EXIT      *
*      MEAN GEO      *      *****
***** *****
*      *
*      *
*      *
*      *
*      *
*      V      30
***** *****
*      *      ***
*      *      REPLACE ARI      *      A4*
*      BY ARI/2      *      ***
***** *****

```

SUBROUTINE CEL12

SUBROUTINE EXP1

SUBROUTINE SICI

SUBROUTINE CS

```

***** ENTER ****
*
*
*
*
*
*
*
V
*****
* SET Z EQUAL *
* TO ABS(X) *
*****
*
*
*
*
*
*
V
*****
IS Z GREATER
THAN 4
*****
YES
NO
*****
3 V
*****
* CALCULATE *
* C=SQR(T(Z)) *
*****
V
*****
* SFT_S EQUAL *
* TUE_SQ *
* REPLACE Z BY*
* Z=S *
*****
V
*****
* EVALUATE *
* APPROXIMATION*
*FOR C AND FOR*
* S*
*****
V
*****
* EXIT *
*****
4
*****
* SET D EQUAL *
* TO COS(Z) AND*
* S EQUAL TO *
* SIN(Z) *
*****
V
*****
* EVALUATE *
* POLYNOMIAL *
* EXPRESSIONS *
* FOR A AND B *
*****
V
*****
* REPLACE Z BY*
* SQRT(Z) *
*****
V
*****
* CALCULATE *
* C=.5+
* Z*(D*A+S*B) *
*****
V
*****
* CALCULATE *
* Z*.5+
* Z*(S*A-D*B) *
*****
V
*****
* EXIT *
*****

```

SUBROUTINE RTWIT

SUBROUTINE RTHIT

SUBROUTINE RTNIT

SUBROUTINE POLRT

SUBROUTINE PADD

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
*  DEFINE      *  
*  DIMENSION OF *  
*  RESULTING    *  
*  POLYNOMIAL   *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
*  ADD          *  
*  CORRESPONDING*  
*  COEFFICIENTS*  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE PADM

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      DEFINE      *  
*      DIMENSION OF *  
*      RESULTING    *  
*      POLYNOMIAL   *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      COMBINE      *  
*      CORRESPONDING*  
*      COEFFICIENTS*  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE PCLA

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*  
*      SET      *  
* IDIMY-IDIMX *  
*****  
*  
*  
*  
*  
*  
*  
*  
* 20 V  
*****  
*  
*      MOVE      *  
* POLYNOMIAL X*  
* INTO      *  
* POLYNOMIAL Y*  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE PSUB

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      DEFINE      *  
*      DIMENSION      OF      *  
*      RESULTING      *  
*      POLYNOMIAL      *  
*****  
*  
*  
*  
*  
*  
*  
* V  
*****  
*      SUBTRACT      *  
*      CORRESPONDING      *  
*      COEFFICIENTS      *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*      EXIT      *  
*****
```

SUBROUTINE PMPY

SUBROUTINE PDIV

SUBROUTINE PQSD

```

***** ENTER ****
*
*
*
*
*
*
V
*****
* INITIALIZE *
* A=0 *
* J=IDIMX *
*****
1   * * * * NO
** * * IS J GREATER * *****) * 3   **** EXIT ****
** * * THAN 0 * * * *
** * * * YES
*
*
*
*
*
2   V
*****
* SET Z=P*A+B *
* AND *
* B=Q*A+X(J) *
*****
*
*
*
*
V
*****
* REPLACE A BY*
* Z *
*****
1
*****)
* SUBTRACT 1 ****) A2*
* FROM J *
*****

```

SUBROUTINE PVAL

```
*****  
* ENTER *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
* SET RESULT *  
* EQUAL TO 0 *  
*****  
*  
*  
*  
*  
* V  
* * * * * YES  
* * IS X ZERO *  
* POLYNOMIAL *  
* * * * *  
* * NO *  
* *  
* *  
* *  
* *  
* V  
*****  
* PERFORM *  
* NESTED *  
* MULTIPLE-  
* CALLON *  
*****  
*  
*  
*  
* { *****  
*  
*  
* V  
*****  
* EXIT *  
*****
```

SUBROUTINE PVSUB

```

      *****
      *    ENTER          *    ** B0**)SET DIMENSION   *
      *    *****          *    ** OF Z TO ZERO   *
      *****
      *
      *
      *
      *
      V
      1      2
      *    *    *    *    YES   ***
      *    IS X A ZERO    *    *) B0*   *B1*)*    EXIT   *
      *    POLYNOMIAL    *    ***   ***   *
      *
      *    *    *    *    NO
      *
      *
      *
      V
      2
      *    *    *    *    YES   ***
      *    IS Y A ZERO    *    *) B1*   *    *
      *    POLYNOMIAL    *    ***   ***   *
      *
      *    *    *    *    NO
      *
      *
      V
      *****
      *    INITIALIZE    *
      *    I=2             *
      *****
      *
      *
      *
      *
      V.
      *****
      ** *PMPY            *
      *    *    *    *    *    *
      *    *A4***)MULTIPLY Y BY*
      *    *    *    1       *
      ** *    *    *    *
      *****
      *
      *
      *
      *
      V
      *****
      *PCLA             *
      *    *    *    *    *
      *    MOVE WORK     *
      *    AREAS         *
      *    *    *    *
      *****
      *
      *
      *
      *
      V
      *****
      *    SET FACTOR =   *
      *    X(IDIMX-I)   *
      *****
      *
      *
      *
      *
      V
      *****
      *PADDM            *
      *    *    *    *    *
      *    ADD Z TO     *
      *    FACTOR*WORK  *
      *    *    *    *
      *****
      *
      *
      *
      *
      V
      2
      *    *    *    *    YES   ***
      *    I=IDIMX        *    *) B1*   *    *
      *
      *    *    *    *    NO
      *
      *
      V
      *****
      *    *    *    *    *
      *    ADD I TO I    *    A4*
      *    *    *    *    *
      *****

```

SUBROUTINE PCLD

```

***** ENTR ****
*
*
*
*
*
V
*****
*
*
SET K=1
*
*****
*
*
L V
*****
*
*
*
*
V
2 * * * * NO
** * * IS J GREATER * *****) ADD 1 TO K *
** * * THAN K * *****) A2*
** * * * * YES
*
*
3 V
*****
*
REPLACE
*(J-1) BY
*X(J-1)+U*(J)*
*****
*
V
*****
* * * * * YES *** 1
* * * * IS IDTMX * * * * *
* * * * GREATER THAN * * * * A2*
* * * * * * * * * * * *
* * * * * NO
*
*
5 V
*****
*
* * * * * SUBTRACT 1 *****) A3*
* * * * * FROM J * * * * EXIT
* * * * * * * * * * * *
*****
```

SUBROUTINE PILO

```
*****  
*      ENTER      *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*      INITIALIZE   *  
*      QUADRATIC    *  
*      SYNTHETIC     *  
*      DIVISION      *  
*****  
*  
*  
*  
*  
*  
* V  
*****  
*      COMPUTE VALUE *  
*      OF DERIVATIVE *  
*      BY SUBROUTINE *  
*      PQSD          *  
*****  
*  
*  
*  
*  
* V  
*****  
*      COMPUTE VALUE *  
*      OF POLYNOMIAL *  
*****  
*  
*  
*  
*  
* V  
*****  
*      EXIT          *  
*****
```

SUBROUTINE POER

```

* **** * **** * **** *
*      ENTER      *
* **** * **** * **** *
*
*
*
*
*
V
*
*   * IS IDIMX *NO 3
*   * GREATER THAN ***** RESULT IS *
*   *          1    * ZERO *
*   *          *    * POLYNOMIAL *
*   *          *    ****
*
*   * YES
*
*
*
*
1
V
**** * **** * **** *
*   * SET
*   * IDIMY=IDIMX-1*
* **** * **** * **** *
*
*
*
*
*
V
**** * **** * **** *
*   * SET I=0
*   * ****
*   * ****
*
*
*
*
*
V
**** * **** * **** *
** * ** ADD 1 TO I **
*   * ****
*   * ****
*   * ****
*   * ****
*   * ****
*   * ****
2
V
**** * **** * **** *
*   * SET
*   * Y(I)=I*X(I+1)*
* **** * **** * **** *
*
*
*
*
*
V
*
*   * * * * NO  *** *
*   * IS I EQUAL TO *1 A4* ***
*   * * * * *
*   * YES
*
*
*
*
V
**** * **** * **** *
*   * EXIT
* **** * **** * **** *

```

SUBROUTINE PINT

```
*****  
* ENTER *  
*****  
  
*****  
* SET *  
* IDIMY=IDIMX+1 *  
* AND Y(1)=0 *  
*****  
  
*****  
* * * IS IDIMX * * NO *  
* * * GREATER THAN * * *  
* * * 0 * * *  
* * * YES * * *  
* * *  
*****  
* SET I=0 *  
*****  
  
*****  
* * * * A4 * * ADD 1 TO I *  
* * * *  
*****  
  
*****  
* * * SET *  
* * * Y(I+1)=X(I)/I *  
* * *  
*****  
  
*****  
* * * IS I * * NO * * *  
* * * EQUAL TO * * * I A4 * * *  
* * * IDIMX * * *  
* * * YES * * *  
* * *  
* * *  
*****  
1  
*****  
* EXIT *  
*****
```

SUBROUTINE PGCD

SUBROUTINE PNORM

```
*****  
*      ENTER      *  
*****  
  
*      V  
*  
*      1  *  *  
*      *  * NO  
*      IS IDIMX *  
*      * A1* GREATER THAN 0 *  
*      *  
*      * YES  
*      *  
*      V  
*      2  *  *  
*      *  * NO  
*      IS X(IDIMX)  
*      * NEGIGIBLE *  
*      *  
*      * YES  
*      *  
*      V  
*      3  *  *  
*      ***  
*      *REPLACE IDIMX*) A1*  
*      BY IDIMX-1 *  
*      ***  
*****  
*      EXIT      *  
*****  
*      V  
*      4  *  *  
*****
```