IBM Application Program

Y20-0093-1

1130 Statistical System (1130-CA-06X)

System Manual

This manual provides detailed information on the logic used in each program of the 1130 Statistical System.

RESTRICTED DISTRIBUTION

CONTENTS

Second Edition

Y20-0093-1 is a minor revision incorporating Technical Newsletter Y20-0144 and does not obsolete Y20-0093-0 with Y20-0144.

Significant changes or additions to the specifications contained in this publication will be reported in subsequent revisions or Technical Newsletters.

Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

1.0 INTRODUCTION

Most subroutines in the 1130 Statistical System are documented by means of a flowchart. The exceptions are those very short routines, FORTRAN-coded, which can be easily understood from the listings, and the Assembly Language subroutines as noted in the index on the next page. The comments and flowcharts associated with the Assembly Language subroutines will be supplied upon request.

Figure 1 illustrates the various blocks used in the flowcharts and their particular meaning. Lines connecting these blocks are made up of periods. Arrows showing the direction of flow are represented by an X.

Connector symbols use the following conventions: four-digit symbols refer to chart symbol (two digits) and block (two digits). For example, ABH1 refers to block H1 on Chart AB. Two-digit symbols refer to a block on the chart where the reference appears. For example, H1 appearing on Chart AB refers to block H1 on the chart.

Index of Subroutines

* Item is not included; listing is considered to be of sufficient aid.

** Item is not included; also, listings are not commented.

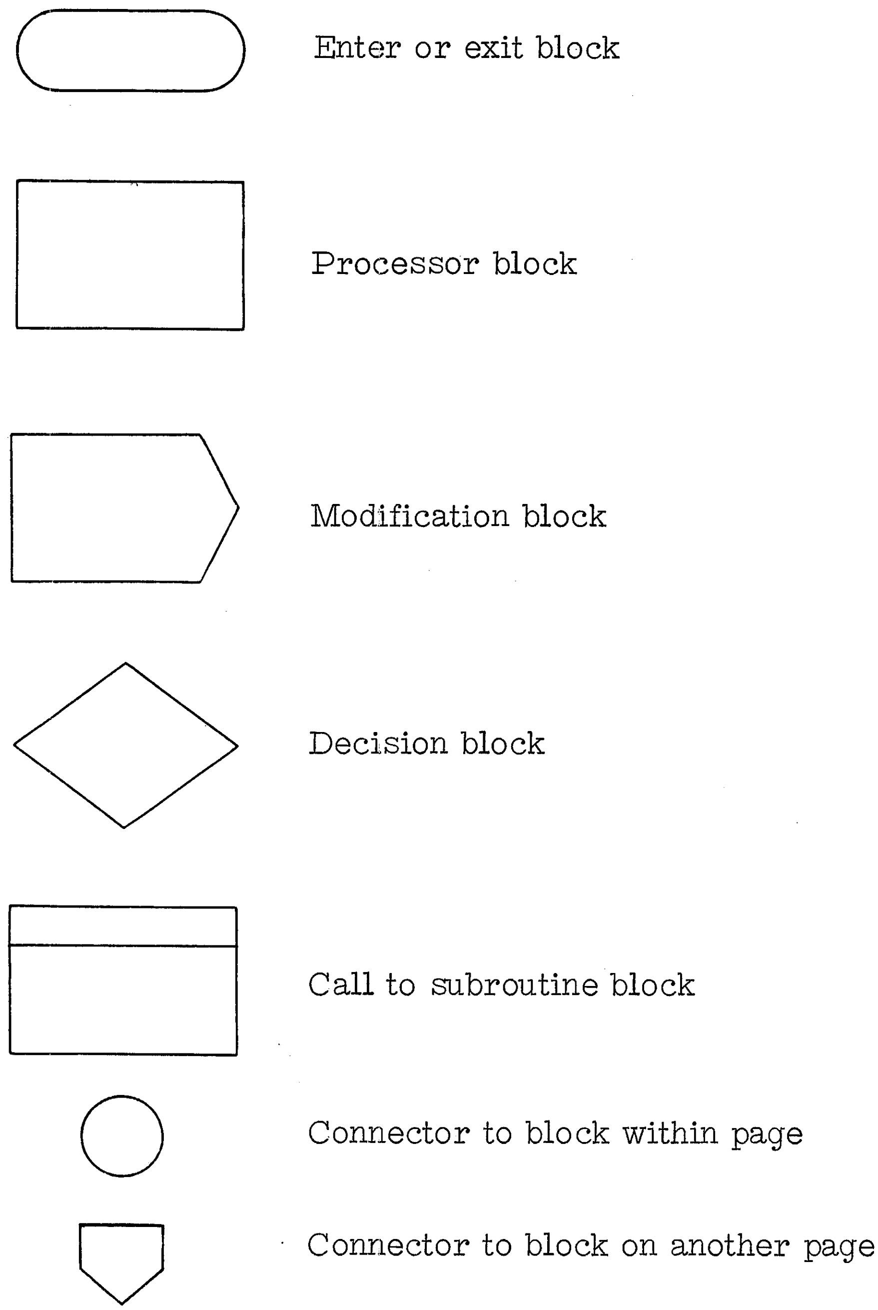


Figure 1. Flowchart blocks

2.0 GENERAL SYSTEM FLOWCHART NARRATIVES

A. Regression and Factor Analysis

(1) Regression

This program uses three main linkage routines, REGR, COREL, and REGR2. These routines perform functions as follows.

(a) REGR: This routine reads standard program control cards, and either calls the matrix read subroutine, MXRAD, or reads the data format cards, with FMTRD, and the data cards, with DATRD. In either case, initialization is performed, and if matrix input is called for, and the matrix is a correlation matrix, the link REGR2 is called. If the matrix is the raw sum of squares matrix, the link COREL is called. If card input is the type called for, the raw sum of squares matrix is computed, sequence checks are made on option, the transformation routine TRAN is called on option, sums of observations on variables are computed and the observations as transformed are placed on the disk.

If disk input is desired, the data format cards are not read, sequence check is ignored, but TRAN is called on option. Thus observations previously transformed will be again transformed if the option is taken.

Finally, for card or disk input, the link COREL is called.

(b) COREL: This routine computes the residual cross products matrix, mean, standard deviations, and correlation matrices. Each of the matrices is printed and/or punched on option by calling the subroutine PRNT. If the raw sums of squares matrix or correlation matrix has been punched, COREL then punches raw sums and sums of squares, or means and standard deviations.

Finally, depending on a switch set in REGR or FCTR, COREL calls the link REGR2 or FCTR1.

(c) REGR2: REGR2 exits to the monitor if no regression is called for. Otherwise the subroutine REGRE is called, after which an exit is made.

(2) Factor Analysis:

This program uses five main linkage routines, FCTR, COREL, FCTR1, FCTR2, and FCTR3, as follows.

(a) FCTR: Input logic in FCTR is identical with that described under REGR above. On exit, FCTR calls the links COREL or FCTR1, depending on the need for a correlation matrix.

(b) COREL: Described above under Regression.

(c) FCTR1: This routine chooses and calculates the communalities, if necessary. Then the subroutines TRIDI and QR are called for eigenvalue computation, after which the link FCTR2 is called.

(d) FCTR2: After determining the number of factors to be rotated, FCTR2 calls VECTR to compute the eigenvectors. It is at this point that, for the minimization of the number of links required, certain arrays are given a maximum length of ten. The number of eigenvectors computed and the number of factors rotated could exceed ten but for this limitation, which could be eliminated. Eigenvectors are then printed on option, then standardized so that the unrotated factor loadings can be printed, and then communalities are computed and printed.

Finally, if a rotation is called for, link FCTR3 is called; if not, FCTR2 exits to the monitor.

(e) FCTR3: FCTR3 either calls VARMX for a varimax rotation or exits to the monitor. If VARMX is called, then an oblique rotation is performed by calling PROMX or an exit is performed. VARMX and PROMX use RFOUT for output. Factor scores and regression coefficients are computed on option by calling SCORE. Finally, FCTR3 exits to the monitor.

B. Analysis of Variance:

This program uses two main linkage routines, ANOVA and ANOV2, which function as follows.

(1) ANOVA: Standard program control cards are read, followed by the option card. After initialization is performed, card or disk input is chosen. If the data is on cards, FMTRD is used to specify data format, and DATRD to read according to that format. After each card, the data is written on the disk. After transforming (TRAN) on option, the program uses STORE if disk storage is required for the design being analyzed. Finally, link ANOV2 is called.

If the data is to be read from disk, format read is ignored, and the program reads from disk, and calls TRAN and STORE if necessary, before calling link ANOV2.

(2) ANOV2: This routine calls SDOP, which generates sums and deviates for each factor, MNSQ, which computes component and interaction sums of squares, and REPRT, which arranges the analysis of variance table according to the user specified table generation cards. Then the program exits to the monitor.

C. Orthogonal Polynomials: The main linkages for this program are POLY and POL2.

(1) POLY: After reading all program control cards, POLY chooses disk, cards, or solution vector input. In the case of cards or solution vectors, format read (FMTRD) is called to set up data card format. If input is from cards, data is read by DATRD, and written on the disk. Disk input or card input then is transformed on option, initialization is performed, and if scaling is to be performed, the scaling equation is calculated. Then link POL2 is called.

If solution vectors are to be read, scaling constants and solution vectors are accepted, secondary input (points for polynomial evaluation) is read with DATRD, and link APOL2 is called.

(2) POL2: This link calls POLSQ unless solution vectors were the input data. POLSQ calculates the orthogonal polynomials and prints them, as well as the solution vectors. If solution vector output is called for, those and the scaling constants are punched by POLSQ.

If the polynomial coefficients are requested, PCOEF is called. If derivatives are required, PDER is called. Finally, PFIT is called if predicted values are desired, after which APOL2 exits to the monitor.

3.0 DETAILED FLOWCHART NARRATIVES S

A. Regression Analysis: This program contains three links.

LINK	SUBROUTINES	USE
REGR	Main Program	Inputs parameter cards and source data
COREL	Main Program	Computes correlation matrix
REGR2	REGRE	Computes regression equations

The links communicate with their successors by storing results in common storage.

COMMON DATA STORAGE MAP - Regression Analysis

Name	Common Dimension*	Type	Meaning
ICR	1	I	Card read symbolic unit
ICP	1	I	Card punch symbolic unit
IPR	1	I	PRINT-TYPE Switch
ITW	1	I	Output unit numbers
IT1	1	I	Not used
IT2	1	I	Not used
IPROB	1	I	Problem number
N	1	I	Number of variables
NF	1	I	Not used
CASES	1	F	Sum of weights
NPAGE	1	I	Page number
INMD	1	I	Input mode switch
IPRED	1	I	Predicted score switch
ISTEP	1	I	Print steps switch
ICNST	1	I	Pooling switch
IREAR	1	I	Dependent variable
KX	1	I	Not used
MX	20	I	Matrix output options
NCD	3	I	Number of variables on Cards 1, 2, and 3
ISEQ	1	I	Sequence check switch
NCASE	1	I	Number of data cases

NX	10	I	Not used
EFOUT	1	F	Criterion for removing variables in REGRE
EFIN	1	F	Criterion for entering variables in REGRE
TOL	1	F	Tolerance for inverse
FLVB	2	F	Not used
KNN	1	I	REGR or FCTR switch
TITLE	18	F	Page title
VNAME	30	F	Variable names
SUMY	30	F	Summary vector - (Means)
SD	30	F	Summary vector - (standard deviations)
X	30	F	Temporary data vector storage
R	(30, 30)	F	Storage matrix (Correlation)
HIGH	30	F	High value of each variable
HLOW	30	F	Low value of each variable
MF	(50, 3)	I	Variable format storage

* The actual number of storage locations occupied by the common variables depends on the variable type. An I, or integer variable, occupies 1 location for each dimension, whereas an F, or Floating Point variable, occupies 2 storage locations.

LINK NAME: REGR
CALLED BY: // XEQ

This link is used to set common storage with all necessary parameters and data for a multiple regression (REGR). The program begins by reading an input/output units designation card from the card reader. This will store the symbolic units ICR, ICP, IPR, ITW, IT1, IT2. The job-title card, regression card and variable names cards are then read from the symbolic unit ICR and job-title and option cards are printed with verbal designation of their meaning on symbolic unit ITW. If INMD = 1 a variable format card will be read and printed. If NCD2 $\neq$ 0 a second variable format card will be read and printed and if NCD3 $\neq$ 0 a third format card will be read and printed. Storage and accumulation arrays are initialized and a branch is taken to the appropriate input section determined by the parameter INMD.

If INMD = 1 a data vector containing case identification, card number, weight, field and data elements X (I), I = 1, N where N is the number of variables set by the user, will be read from the card reader and stored on the disk. If the parameter ISEQ $\neq$ 0 and the NCD(I) are set to the proper values the input cards will sequence check within case before the elements X (I) from the card are stored. If INMD=2 the data vector will be transferred from the disk to the core vector X When INMD=3 the source data is a matrix and will be read from the card reader, by the subroutine MXRAD.

Once the data vector has been transferred from the input device to the core vector X a test is made to see if the case identification field ID1 is negative or zero. If it is non-positive, the next link (REGR2 or COREL) is read into core storage and executed.

If INMD $\neq$ 3, the program accumulates the sums vector and sums of squares and cross products matrix from the data vector X. In addition, the high and low value of each element in X is also determined. When this information is completed the program branches back to read another data vector.

On exit, all options, heading information, and I/O unit designations are stored in common, along with the summary statistics and cross-product matrix of the input matrix (if the input matrix was a data matrix) or the input matrix itself (if it was a correlation or cross-product matrix). The common variable NCASE indicates which type of input was accepted.

LINK NAME: COREL

CALLED BY: REGR or FCTR

For a description of COREL, see Section 3C.

LINK NAME: REGR2

CALLED BY: COREL or REGR

The first thing that the link REGR2 does is test the parameter IREAR, which normally contains the column number of the dependent variable, to determine whether a regression analysis should take place. If IREAR = 0, subroutine REGRE will not be called and the program will finish with a call exit statement. If IREAR is greater than zero the subroutine REGRE will be called and the regression equations computed from the correlation matrix, means and standard deviations located in common storage.

SUBROUTINE REGRE

CALLED BY: REGR2

REGRE performs the following functions:

1. The dependent variable is placed in the last row and column of the correlation matrix R. That is, r_{ij} is moved to the last row and column of R.

Other pertinent vectors are similarly changed.

2. $r_{i,y}^2/r_{i,i}$ is checked to determine entry variables. If none is entered, REGRE returns to REGR2. Otherwise, requested output is prepared and printed.

3. Entry and exit significance levels are checked, variables for entry or exit are chosen, and output is presented until either degrees of freedom are zero, no more variables are to be entered or removed, or the residual mean square is negative.

B. Factor Analysis: This program will perform a complete factor analysis from either the raw data or a pre-computed correlation matrix.

The factor analysis program contains five links:

LINK	SUBROUTINE	USE
FCTR	Main program	Inputs parameter cards and source data
COREL	Main program	Computes correlation matrix
FCTR1	TRIDI	Tridiagonalizes matrix
	QR	Computes eigenvalues
FCTR2	VECTR	Computes eigenvectors
	COVEC	Solves tridiagonal equations
FCTR3	VARMX	Orthogonal factor rotation
	PROMX	Oblique factor rotation
	SCORE	Computes and outputs factor scores

Each of these links communicates with its successor by storing its results in common storage.

COMMON DATA STORAGE MAP - Factor Analysis

Name	Common Dimension*	Type	Meaning
ICR	1	I	Card reader symbolic unit
ICP	1	I	Card punch symbolic unit
IPR	1	I	Print-type switch
ITW	1	I	Printer-typewriter unit
IT1	1	I	Not used
IT2	1	I	Not used
IPROB	1	I	Problem number
N	1	I	Number of variables
NF	1	I	Number of factors
CASES	1	F	Sum of weights
NPAGE	1	I	Page number
INMD	1	I	Input mode switch
IPRED	1	I	Factor score switch

* The actual number of storage locations occupied by the common variables depends on the variable type. An I, or integer variable, occupies 1 location for each dimension, whereas an F, or floating point variable, occupies 2 storage locations.

ICOM	1	I	Communality option
IROT	1	I	Rotation switch
NFRT	1	I	Number of factors to rotate
KX	1	I	VARMX/PROMX switch
MX	20	I	Matrix output options
NCD	3	I	Number of variables on Cards 1, 2, and 3
ISEQ	1	I	Sequence check switch
NCASE	1	I	Number of data cases
KCNT	1	I	Parameter for factor count
KNN	1	I	REGR or FCTR Switch
NX	9	I	NX(1) is a pooling switch
TRC	1	F	Trace
FLVB	4	F	Not used
TITLE	18	F	Page title
VNAME	30	F	Variable names
SUMY	30	F	Summary vector (Means)
SD	30	F	Summary vector (Standard deviations)
X	30	F	Temporary data vector storage
R	(30,30)	F	Storage matrix(Correlation)
HIGH	30	F	High value of each variable
HLOW	30	F	Low value of each variable
MF	(50, 3)	I	Format storage
ALPHA	30	F	Contain elements
BETA	30	F	of tridiagonal matrix

LINK NAME : FCTR

CALLED BY: //XEQ

The first link loaded is FCTR which reads all parameter cards and stores the analysis options and parameters in common storage. Then either a pre-computed matrix is read or a raw cross product matrix is formed from the raw data matrix. The common variable NCASE is set to either a negative or positive value depending on whether a correlation matrix or input data was read. Link FCTR1 is then loaded into core if NCASE is less than zero; otherwise link COREL is loaded.

LINK NAME: FCTR1

CALLED BY: LINK FCTR

This link is used as a factor analysis setup program. From the parameter ICOM, which has been determined by the user, the diagonal elements of the correlation matrix are replaced by estimates of the communalities. There are three possible values of ICOM and these correspond to the three primary branches in the program.

If ICOM = 0, the diagonal elements of the correlation matrix are unchanged. In effect, this corresponds to a principal components analysis where the communality estimate is equal to 1.

If ICOM = 1, each element on the diagonal will be replaced by the absolute value of the largest off-diagonal element in a row.

If ICOM = 2, each diagonal element will be replaced by the square of the multiple correlation coefficient (i.e., if i represents the ith diagonal element, then R_{ii} will be the multiple correlation between the ith variable and all other variables in the matrix).

After the communality estimates have been determined, the program computes the trace of the matrix by summing the diagonal elements and storing the result at the symbolic location TRC. The subroutines TRIDI and QR are then called to compute the eigenvalues of the new matrix.

Upon entry to the program, the correlation matrix, or matrix to be factored, is assumed to be located in the matrix R. The parameters N and ICOM have been read into common storage by link FCTR.

When link FCTR2 is called the trace of the correlation matrix is in location TRC, the diagonal elements of the tridiagonalized correlation matrix are in array ALPHA, the off-diagonal elements are in array BETA and the eigenvalues are in array X.

SUBROUTINE NAME: INVRS

CALLED BY: FCTR1

Description: INVRS inverts a symmetric matrix with unit diagonal elements. On entry, the matrix is in array R. The upper triangular part of the matrix is replaced by the elements of the inverse. The part below the diagonal is not modified.

SUBROUTINE: TRIDI

CALLED BY: FCTR1

This subroutine converts a symmetric matrix to tridiagonal form.

The method employed is Householder's method. In this method, N-2 elementary orthogonal transformations are chosen in such a way that the transformation will leave only the first subdiagonal element in the rth column nonzero. The final matrix

$$A' = P_{n-2}P_{n-3}\cdots P_2 P_1 A P_1 P_2 \cdots P_{n-3} P_{n-2}$$

can be stored in two arrays, the first (ALPHA) containing the elements of the main diagonal and the second (BETA) containing the first subdiagonal element in each column (except the last).

Along with the transformed matrix a transformation matrix is formed and stored over the original matrix. This matrix is computed as

$$T = P_1 P_2 P_3 \cdots P_{n-2}$$

and has the property that an eigenvector of the tridiagonal matrix, when premultiplied by T, becomes an eigenvector of the original input matrix.

On entry, the correlation matrix is in array R.

On exit, the transformed matrix is in common arrays ALPHA and BETA. The transformation matrix is in R. The infinity norm of the transformed matrix is in the common cell ANORM.

SUBROUTINE QR

CALLED BY: FCTR1

QR finds up to thirty eigenvalues of the tridiagonal matrix previously prepared by the routine TRIDI. The QR method is used.

On exit, the eigenvalues are in descending order in the vector X.

LINK NAME: FCTR2

CALLED BY: LINK FCTR1

This subroutine is used to compute and output the factor matrix. First it determines the number of factors to compute from the parameters NF and KCNT. If NF = 0 then the factors computed will be those which have characteristic values greater than or equal to 1. If NF = 2 then the number of factors computed will be the number which is in KCNT. If NF = 3 the number of factors computed will be only those factors which account for KCNT percentage of the trace.

As each characteristic value is examined a cumulative sum is computed and the cumulative percentage of trace is computed.

The routine VECTR is called to compute the required number of eigenvectors, and the factor matrix is computed by the expressions:

R(I, J) = R(I, J) * SQRT(X[J])

where R(I, J) on the right side of the equal sign contains the characteristic vector and X(J) is the Jth characteristic value.

Once the factor computation has been completed the characteristic values and cumulative percent of trace are printed along with the title, page number, trace, sum of the characteristic values, and the difference between the trace and sum. Subroutine PRNT is then called to output the factor matrix. Communalities are calculated from the sums of squares of the row elements of the factor matrix, and printed.

When the link is called the elements of the tridiagonal matrix are in arrays ALPHA and BETA, and the characteristic values are in array X. The parameters NF and KCNT are at values assigned by the user.

Upon exit from the link R contains the principal axis factor matrix. The parameters NF and KCNT have been changed to the following values:

Entry	Exit	
NF	NF	KCNT
0	0	0
1	0	0
2	KCNT	0
3	0	KCNT

SUBROUTINE NAME: VECTR

CALLED BY: FCTR2

VECTR is a subroutine that computes NF eigenvalues of an N x N matrix by computing the eigenvalues of the tridiagonalized matrix obtained by subroutine TRIDI and transforming them to the characteristic vectors of the original matrix via a transformation matrix. (See TRIDI narrative.)

The method by which the K^{th} eigenvector is found is as follows:

The eigenvector array (V) is initialized to ones as a first approximation. Subroutine COVEC is called to compute

$$Q = (A - X_k I)^{-1} V$$

where A is the tridiagonalized matrix and X_k is the k^{th} eigenvalue. If V and Q, when normalized, do not agree (element for element) within .05, V is set equal to Q and the routine reiterates. If V and Q agree, the vector

$$R = V - (A - X_k I) Q$$

is formed and COVEC is called to compute

$$Y = (A - X_k I)^{-1} R.$$

Z = V + Y is then the eigenvector of the tridiagonal matrix. Z is then normalized and premultiplied by the transformation matrix and written onto the disk.

When all NF eigenvectors have been found and written on the disk, they are read back in over the transformation matrix.

The tridiagonalized matrix must be in arrays ALPHA and BETA, its eigenvalues must be in array X, and the transformation matrix must be in the Matrix R, on entry.

On exit, the eigenvectors are in R and the transformation matrix is destroyed. If a rotation is required, FCTR3 is called.

SUBROUTINE NAME: COVEC

CALLED BY: VECTR

COVEC solves the system of tridiagonal equations

$$(A - XI) \cdot V = C$$

for V, where A is a tridiagonal matrix of the form

$$\begin{bmatrix} a_1 & b_2 & 0 & 0 & 0 & . & . & 0 \\ b_2 & a_2 & b_3 & 0 & 0 & . & . & 0 \\ 0 & b_3 & a_3 & b_4 & & & & \\ 0 & 0 & b_4 & a_4 & & & & \\ 0 & 0 & 0 & & . & & & \\ . & . & . & & & & . & 0 \\ . & . & . & & & & . & b_n \\ 0 & 0 & . & . & & & 0 & b_n \; a_n \end{bmatrix}$$

which was stored in the arrays ALPHA and BETA. X is an approximate eigenvalue of A, and C is a column vector. An eliminative scheme is used which uses the largest element in each column as its pivot element.

The arguments used when calling this routine have the following meaning:

CONS	= Vector or right side of equation
VECT	= Vector to be solved for

In addition, the program requires the arrays ALPHA and BETA.

The solution vector is in the argument VECT, on exit.

SUBROUTINE NAME: PRNT

CALLED BY: FCTR2

This subroutine is described in Section 3C.

LINK NAME: FCTR3

CALLED BY: FCTR2

This link calls VARMX if an orthogonal rotation is required, PROMX if an oblique rotation is required, and SCORE for factor score calculations. RFOUT is used for output, and MATIN for matrix inversion. FCTR3 then exits to the monitor.

SUBROUTINE NAME: VARMX

CALLING PROGRAM : FCTR3

After initializing NFRT, the number of factors to be rotated, and setting the tolerance EPS, the program sets the B matrix to an identity matrix. The A matrix, which contains the factors to be rotated, is then row normalized by dividing each row element by the communality for that row.

The main iteration cycle is then initiated by computing a convergence criterion and comparing it to the criterion on the previous cycle. If it is approximately zero, control will be returned to calling program. If greater than zero it initiates a new cycle. A cycle consists of a pairwise rotation of the factor matrix. The program determines the sine and cosine of the angle of rotation and proceeds to apply this angle to the matrix. However, if this angle is less than 1 minute of 1 degree (EPS) then a rotation will not be effected.

After the factor matrix has been rotated by the sine and cosine of the rotating angle, the B matrix,which initially contains the identity matrix, is also rotated by this angle. The program then begins another iteration cycle. At the beginning of each iteration the cycle count and convergence criterion are printed. A test is made to determine if more than 50 cycles have taken place. If so the program will terminate.

Upon entry to the program the factor matrix is located at A, the number of factors to be rotated is contained in NFRT. If this field is zero, the program will set it equal to the number of factors as determined by the program FCTR2.

Upon exit from the program the array A contains the orthogonally rotated matrix and B contains the transformation matrix.

SUBROUTINE NAME: PROMX

CALLING PROGRAM: FCTR3

This subroutine, in conjunction with RFOUT, is used to perform an oblique rotation of a factor matrix.

After setting IAL to four,the program computes the inverse of $A^T \cdot A = B$ where A is the orthogonal factor matrix and A^T its transpose.

Row- and column-normalizing vectors H and G are then computed for use in the computation of the E matrix. The expression for this is:

$$E = A^T * P$$

where P = row normalized A matrix with each element raised to the IALth power. The sign of each element remains the same as the unpowered element.

Following this the transformation matrix to the oblique reference vector structure matrix is computed by:

$$B = B * E$$

where B on the right contains $A^T \cdot A$.

The transformation matrix is then formed by normalizing the columns of B.

Once this transformation matrix is complete, it is applied to the A matrix to form the reference vector structure matrix. Also, multiplying it by its transpose produces the correlations among reference vectors.

Upon entry to the program, array A contains the orthogonally rotated factor matrix from VARMX.

Upon exit, A contains the oblique reference vector structure matrix, B contains the transformation matrix and E contains the correlations among reference vectors. The common variable KX(1) has been set equal to 1 for program RFOUT.

SUBROUTINE NAME: RFOUT

CALLING PROGRAM: FCTR3

This subroutine is used to output the results of an orthogonal rotation and/ or compute and output the remainder of the matrices associated with an oblique rotation. The program determines whether the program preceding it was VARMX or PROMX by the common variable KX(I). If KX(I) = 0 then VARMX preceded and RPRNT is called to output the transformation matrix and the orthogonal factor matrix. Before returning to the calling program, B is set to an identity matrix for possible factor score computation.

If KX(I) = 1 then the preceding program was PROMX and different output and computational functions are performed by RFOUT. RPRNT is called to output the correlations among oblique reference vectors, (E), oblique reference vector structure matrix, (A), and the correlations among oblique reference vectors, (B). Matrix E is then inverted by MATIN and the reference vector pattern matrix computed and RPRNT called to output this matrix. The correlations among reference vectors and primary factors are then computed and printed by RPRNT. Using this result the correlations among primary factors are computed and presented. Finally, the primary factor structure matrix and primary factor pattern matrix are computed and presented.

Upon entry KX(I) = 0 if entry was from VARMX and A contains the orthogonal factor matrix, B the transformation matrix.

When KX(I) = 1, A contains the oblique reference vector pattern matrix, B contains the transformation matrix and E contains the correlations among reference vectors.

Upon exit from the program A will contain the primary factor pattern matrix and B, if from VARMX, will contain an identity matrix, or, if from PROMX, will contain the correlations among primary factors.

SUBROUTINE NAME: RPRNT

CALLING PROGRAM: RFOUT, SCORE

This subroutine is used to print the following matrices:

1. Orthogonal transformation matrix
2. Orthogonal factor matrix
3. Transformation to oblique reference vector structure
4. Oblique reference vector structure
5. Correlations among oblique reference vectors
6. Oblique references vector patterns
7. Correlations between reference vectors and primary factors
8. Oblique primary factor structure
9. Correlations among oblique primary factors
10. Oblique primary factor loadings
11. Factor score regression coefficients

This program has the same logic and structure as subroutine PRNT except for two minor differences. Column headings on printout are numerical sequence values and are not taken from the array VNAME. The second major difference is the meaning of the argument KODE. RPRNT will output either the common array A or B; if KODE = 0, then A will be printed with row headings taken from VNAME and columns in generated numerical sequence. If KODE = 1 then the array B will be printed with generated numerical sequence for column and row headings.

Matrix B or E contains the output matrix if KODE = 1 or the A array contains the output matrix if KODE = 0. MID contains the matrix identification number, KODE = 1 or 0. NR is the number of columns of the output matrix. These are the exit conditions.

SUBROUTINE NAME: MATIN

CALLING PROGRAM: RFOUT, SCORE

MATIN inverts a symmetric matrix.

SUBROUTINE NAME: SCORE

CALLING PROGRAM: FCTR3

SCORE is used to compute the factor score regression coefficients and factor scores from either an oblique or orthogonal factor matrix. The program divides each element in the factor matrix A by 1-X(I) where X(I) are the communalities for each row. The resultant matrix is stored in the last ten columns of the array A.

The transpose of the original matrix multiplied by this matrix is then added to the B matrix. The B matrix contains either an identity matrix if the factors are orthogonal or the inverse of the intercorrelations of the primary factors and oblique factors. The resultant matrix B is then inverted by subroutine MATIN and the inverse multiplied by the modified A matrix to form the factor score regression coefficients. Subroutine RPRNT is then called to output this matrix.

Factor scores are then computed from the regression coefficients by reading a data factor from the disk. If the problem number ID is positive, each variable X(I) in the data vector is standardized by :

Z (I) = (X[I] - SUMY[I])/SD(I)

where SUMY(I) contains the mean of the ith variable and SD(I) contains the standard deviation. The N elements of the standardized data vector Z are then multiplied by the N elements in each of the NFRT regression coefficients in A to form the factor scores for this data vector. The vector is then printed out with a sequence number and case identification ID. The title, job number and column headings are also printed on each page.

If ID is negative the program will terminate and return to the main calling program.

Upon entry to the subroutine matrix A contains the factor matrix. The raw data, followed by an artificial data vector with a negative identification must be located on the disk. Matrix B will contain an identity matrix if the factors are orthogonal or the primary factor intercorrelations if the factors are oblique. The arrays SUMY and SD contain the means and standard deviations respectively.

Upon exit the array A contains the factor score regression coefficients and the factor scores have been printed and/or punched.

SUBROUTINE NAME: XMAX

XMAX computes the maximum of two arguments.

C. Routines Used by Regression and Factor Analysis

LINK NAME: COREL

CALLING PROGRAM: REGR and FCTR

After initialization of switches and moving the sums of squares from the diagonal elements of the cross product matrix R to the vector SD for possible punchout, subroutine PRNT is called to examine MX(1) for either printing and/or punching the raw cross products matrix. From the raw cross products matrix, the residual cross products matrix is then computed by:

R(I, J) = R(I, J) - SUMY(I) * SUMY(J)/CASES

where: R(I, J) on the right hand side of the equal sign contains I, Jth raw cross products

SUMY(I) contains the raw sum

CASES contains the number of observations

After the computation is completed, subroutine PRNT is then called for printing and/or punching.

From the residual cross product matrix, the variance-covariance matrix is computed by

R(I, J) = R(I, J) /(CASES - 1)

where: R(I, J) on the right contains the residual cross products and and CASES contains the number of observations.

After the computation is completed, subroutine PRNT is again called for possible output.

Once the variance-covariance matrix is computed the means and standard deviations are computed by:

SUMY(I) = SUMY(I)/CASES

SD(I) = SQRT(R[I, I])

A summary statistics table is then printed which contains the number of cases, variable names, high and low value of each variable, and means and standard deviations.

Once this printout is completed, the correlation matrix is computed by:

R(I, J) = R(I, J) / (SD[I]*SD[J])

In the computation a test is made to determine if either SD(I) or SD(J), the standard deviation of the Ith and Jth variable respectively, is zero. If either one is zero, the correlation coefficient R(I, J) is set to zero and a message indicating which variable has the zero variance is printed.

After the computation is completed, subroutine PRNT is again called to output the matrix.

Upon entry to the program, CASE (the number of observations), SUMY (the cumulative raw sums of each variable), and R(I, J) (the cumulative raw cross product matrix) have been either read in as matrices or accumulated previously. The high and low values of each variable are also present in the vectors HIGH and HLOW.

Upon exit from the program the means, standard deviations, correlation matrix and sum of weights are in common storage at locations represented by SUMY, SD, R, and CASES, respectively.

SUBROUTINE NAME: PRNT

CALLING PROGRAM: COREL, FCTR2

Subroutine PRNT is used to print and/or punch the following six matrices:

1. Matrix of raw cross products
2. Matrix of residual cross products
3. Variance-covariance matrix
4. Matrix of correlation coefficients
5. Matrix of characteristic vectors
6. Principal axis factor matrix

The program examines the output option array MX subscripted by the argument MID. If MX(MID) = 0, control will be returned to the calling program and no output will occur. If MX(MID) = 2 or 3, control will be transferred to the punch routine and the matrix will be punched, 5 elements to a card with identification indicating the problem number (IPROB), the matrix identification number (MID), the row of the matrix in which these 5 elements are located (I), and the column of the first element on the card (K).

After the matrix has been punched MX(MID) is again examined to determine if it contains a value of 1 or 2. If it does not, the program will return to the calling program. However, if it does contain 1 or 2, the program will branch to the print routine. The print routine will print the title, page number, and job number followed by the name of the matrix as identified by MID. The matrix is printed, 8 elements to a line, with each column and row identified by a variable name as contained in VNAME. After the entire matrix is printed, control is returned to the calling program.

There are four arguments used in the calling sequence to the subroutine. These have the following meaning:

MID	Matrix identification number
KODE	KODE is unused, but could be used for a switch allowing different formats.
NR	contains the number of rows in the matrix
NC	contains the number of columns in the matrix

The matrix to be printed or punched is located in the common array R.

On exit, the common variable NPAGE has been incremented by the number of pages required to print the entire matrix. There are no other changes to any other common locations.

SUBROUTINE NAME: MXRAD

CALLING PROGRAM: FCTR, REGR

This subroutine is used by link FCTR and link REGR to read and/or add matrices. The program starts by reading a card containing the problem number(IP), matrix identification number (MID), the column number of the first data element on the card (IC), the row number (IR), and 5 data elements X(I), I = 1, 5. If IP is negative, the program branches to a termination routine which will set the common variable

NCASE to either a positive or negative value depending on whether the correlation matrix was processed. Control is then transferred to link FCTR or link REGR.

If IP is positive and the row card is from a matrix other that 21 or 22, the 5 elements X(I) are added to contents of the core matrix R, subscripted by IC and IR. If MID = 21, there is only one legitimate element on the row card, and this, added to the common variable CASES, is the number of observations. If MID = 22, there are only two legitimate elements on the row card, and these are added to the common vectors SUM and X.

After the row card has been added to a matrix or vector (R, CASES, SUM or X) another card is read and the same process is initiated. Cards will be read until a negative problem number card comes up and the process is terminated, unless ICNST is non zero. If this is the case, a second matrix is accepted, and subtracted from those previously read. In this case, matrices should be raw cross products matrices.

Upon entry to the program, the common variables R, NCASE, CASES, SUM and X have been set to zero. The variable ICNST is set to allow pooling.

Upon exit, R, CASES, SUM, X have been set with input from the card reader. The variable NCASE has been made either positive or negative to determine logic flow in the main calling program. NCASE positive implies a raw cross products data set has been read and control will be passed to the correlation matrix generation program COREL. If NCASE is negative or zero, control will bypass COREL as a correlation matrix data set has been read.

D. Analysis of Variance

LINK	SUBROUTINES	USE
ANOVA	Main Program	Inputs parameter cards Inputs source data
ANOV2	SDOP	Forms sums and deviates
	MNSQ	Forms sum of squares
	REPRT	Forms mean square and output table.

COMMON DATA STORAGE MAP - Analysis of Variance

Name	Common Dimension*	Type	Meaning
ICR	1	I	Card reader symbolic unit
ICP	1	I	Card punch symbolic unit
IPR	1	I	Print-Type switch
IT1	1	I	Not used
IT2	1	I	Not used
IPROB	1	I	Problem number
NPAGE	1	I	Page number
INMD	1	I	Input mode
NF	1	I	Number of factors
ITRN	1	I	Transformation switch
NA	1	I	Number of levels +1, Factor 1
NB	1	I	Number of levels +1, Factor 2
NC	1	I	Number of levels +1, Factor 3
ND	1	I	Number of levels +1, Factor 4
TITLE	18	F	Page title
NX	5	I	Number of levels for each factor
LS	5	I	Temporary constants
IN	4	I	Data input array
NDIV	20	I	Divisors for sum of squares
SMQR	20	I	Summary vector for sum of squares
XDEV	20	I	Storage for deviates
X	1500	F	Data storage array
ITW	1	I	Output unit numbers

* The actual number of storage locations occupied by the common variables depends on the variable type. An I, or integer variable, occupies 1 location for each dimension, whereas an F, or floating point variable,occupies 2 storage locations.

LINK NAME: ANOVA

LOADED BY: // XEQ

This link is used to read parameter cards and source data for analysis of variance. The program first reads an input-output units designation card from the card reader. It then reads a title card and the analysis of variance parameter card. If the parameter INMD = 1 a variable format card is read and printed.

After initializing the storage parameters for the number of factor levels the program reads a data record from either the card reader if INMD = 1, or from the disk if INMD = 2. The data record contains an index array IN and a data item. If the first index array item IN(1) is positive, the program will compute the storage location IS for this data item from the index array IN and the storage parameter LS. If the transformation switch is on, the transformation program will be called. Following this, the STORE program will be called to either store the data item DATA in storage or on disk. Following the return from program STORE, the program will branch back to read another data record.

Upon exit from the program all the parameters are in common and all the required data has been stored either on the disk or in the array X. The condition for storing the data in X is determined by the storage parameter IS. If IS is greater than 1500 the data will be stored on the disk; otherwise, in the array X.

SUBROUTINE NAME: STORE, GET

CALLING PROGRAM: ANOVA, SDOP, MNSQ

These subroutines are used to store or get data either from the array X or disk. The programs test the argument IS; if IS is less than 1500 the data will be stored or retrieved from the core array X. If IS is greater than 1500 the item will be stored on the disk at storage location IS-1500. After the item has been either stored or retrieved, control is returned to the calling program.

On entry, DATA contains the item.
IS contains the location parameter.

On exit,

STORE - DATA has been stored in X or on the disk.
GET - DATA has been retrieved from X or from the disk.

LINK NAME: ANOV2

CALLED BY: ANOVA

This program calls the remaining analysis of variance programs, SDOP, MNSQ, and REPRT, and exits to the monitor.

SUBROUTINE NAME: SDOP

CALLING PROGRAM: ANOV2

This subroutine is used to generate the analysis of variance sums and deviates for each factor. The program computes appropriate storage locations for the data and calls subroutine GET to obtain the data item for the Kth factor from either the array X or from the disk. Each data item is then summed over all levels for this factor and the sum located at SUMX is stored back in the array X at the appropriate location IS.

After the sum is computed for the Kth factor the data array is again used to compute the analysis of variance deviate. Each element used to form SUMX is replaced by

DATA = FN*DATA - SUMX

where FN is the number of levels in the factor.

After this computation, the storage pointers IS and ISPM are incremented and a test is made to determine the appropriate level. The factor count K is then incremented and computations are performed on the transformed data elements. After passing through the data the program returns to the main calling program

Upon entry to the program all data items have been stored in either the array X or on the disk. The number of levels for each factor is located in the array NX and the number of factors is located in NF.

Upon exit from the program the data array (either X or disk) contains the sums and analysis of variance deviates.

SUBROUTINE NAME: MNSQ

CALLING PROGRAM: ANOV2

This subroutine is used to compute the component and interaction sums of squares for the final analysis of variance table. After initialization of the cumulation arrays, the program determines which component in the analysis of variance table is to be incremented for the current data item. The analysis of variance table SMQR can contain at most 15 values. These are related to the component and interaction sum of squares as follows:

Index	Component
1	A
2	B
3	C
4	D
5	AB
6	AC
7	AD
8	BC
9	BD
10	CD
11	ABC
12	ABD
13	ACD
14	BCD
15	ABCD

where A, B, C, D are names of the factors. It should be noted that even if a particular job does not involve four factors, the subscript for the particular component is still the same.

By passing through the data array (core and/or disk) in a sequential manner, the program is able to determine which index value is required for SMQR by testing the individual factor level counts IA, IB, IC, ID and comparing these to the number of levels in each factor, NA, NB, NC, ND. When the proper subscript is determined,K is set equal to this value and the program adds the square of the deviate to the appropriate cell in SMQR. When all deviates have been processed the program returns to the main calling program.

On entry, the anaylsis of variance deviates are located either in the X array and/or on the disk. The number of levels in each factor are located in NA, NB, NC and ND.

On exit, the component and interaction sums of squares multiplied by the component or interaction are located in SMQR. The deviates of interest are in XDEV and the divisor necessary to obtain the component sum of squares is located in NDIV.

SUBROUTINE NAME: REPRT

CALLING PROGRAM: ANOV2

This program is used to output the analysis of variance table. The program begins by setting up a general array for the degrees of freedom. Next, the appropriate divisor and accumulation arrays are initialized, and the total sum of squares is computed. A card, containing a 24-character row heading (HEAD), a control indicator (INDI), and a component summary index array (INX), is then read from the card reader. The index array, INX, is then used to subscript the SMQR array, which contains the component sums of squares. To add the appropriate elements to form the component to SMSQ and degrees of freedom NDF1 after all emements of INX are chosen, the mean square SMSQM is computed by dividing the sums of squares by the degrees of freedom. Once this computation is completed, a line is printed containing the sums of squares, degrees of freedom and mean square. If INDI is greater than zero, a page will be skipped and a title line with column headings will be printed before the component line. If INDI is negative the program will terminate by printing a residual line if necessary and/or total line. The residual is the difference between the total sum of squares computed in the beginning of the program and the sum of squares accumulated after each line has been printed.

On entry, except for the proper divisor, the component sums of squares are located in SMQR, and NDIV contains the divisor to compute the sums of squares.

On exit, SMQR contains the component sums of squares and the requested component lines have been printed.

E. Orthogonal Polynomials

The program contains the two links:

LINK	SUBROUTINES	USE
POLY	Main Program	Inputs parameter cards and source data
POL2	POLSQ	Determines degrees and computes orthogonal polynomials
	PCOEF	Computes coefficients of fitted polynomial
	PDER	Computes derivatives at a point
	PFIT	Computes predicted Y for a given X.

COMMON DATA STORAGE MAP - Orthogonal Polynomials

Name	Common Dimensions*	Type	Meaning
ICR	1	I	Card reader symbolic unit
ICP	1	I	Card punch symbolic unit
IPR	1	I	Print-Type switch
ITW	1	I	Output unit numbers
IT1	1	I	Not used
IT2	1	I	Not used
IPROB	1	I	Problem number
N	1	I	Maximum degree of polynomial
NF	1	I	Actual degree of polynomial
CASES	1	F	Not used
NPAGE	1	I	Page number
INMD	1	I	Primary input mode
ISCR	1	I	Predicted values switch
NCASE	1	I	Number of data cases
ICOF	1	I	Coefficient computation switch
IDER	1	I	Derivative computation switch
NDER	1	I	Order of derivatives
IALP	1	I	Polynomial solution vector output switch
INMD2	1	I	Secondary input mode
KX	3	I	Not used
EPS	1	F	Tolerance criterion

FLVB	4	F	Not used
XB	1	F	Scaling Constant
X14	1	F	Scaling Constant
TITLE	18	F	Page title
ID	150	F	Identification codes
X	150	F	Values of X
Y	150	F	Values of Y
C	51	F	Polynomial solution vector
ALPHA	51	F	Polynomial solution vector
BETA	51	F	Polynomial solution vector
MF1	50	I	Format for input data

* The actual number of storage locations occupied by the common variables depends on the variable type. An I, or integer variable, occupies 1 location for each dimension, whereas an F, or floating point variable,occupies 2 storage locations.

LINK NAME: POLY

LOADED BY: // XEQ

This program is used to read parameter cards and source data for orthogonal polynomials. The program first reads an input-output units designation card from the card reader, followed by a title card and the orthogonal polynomials parameter card. If the parameter INMD = 1 or 3, a variable format card is read and printed. The program then branches to a special input section for each value (1, 2, or 3) of the parameter INMD.

If INMD = 1 the program will read the source data from the card reader. Each input record contains an identification field (ID [I]), a derivative computation indicator (IDR), an X value X(I), and a Y value Y(I). If ID(I) is positive , the program will test IDR for non-zero. If zero, the program will read another card record; if non-zero, the identification for this record ID(I), will be made negative for examination by the program PDER. If ID(I) is negative, link POL2 is loaded and executed.

If INMD = 2, the input data will be read from the disk instead of the card reader. It was placed there by the previous use of INMD = 1.

If INMD = 3, the polynomial solution vectors will be read into arrays

ALPHA, BETA, and C respectively, along with any necessary scaling constants. A branch is then made to the section corresponding to INMD = 1 in order to read the data points.

Upon exit, the analysis parameters and data points are in common. In addition, if the parameter INMD2 $\neq 0$ the polynomial solution vectors are in COMMON. If scaling was requested, scaling constants are also in COMMON.

LINK: POL2

CALLED BY: POLY

POL2 calls the remaining programs in this analysis type if they are required, i.e., POLSQ for polynomials, PCOEF for coefficients, PDEF for derivatives, and PFIT for evaluation and prediction.

SUBROUTINE NAME: POLSQ

CALLING PROGRAM: POL2

After initializing the computational parameters and accumulation arrays the program begins the main iteration loop by computing the first polynomial solution vector C by:

$$C(II) = S/RO$$

where II is the current degree of the computed orthogonal polynomial,
S is the inner product of Y and IIth degree orthogonal polynomial,
RO is the inner product of the polynomial with itself.

Once S is computed the cumulative predicted values for 1, 2..IIth degree polynomials are computed and stored in array YA. The variance criterion for the cycle is then computed and compared to its value on the previous cycle. If the difference is approximately equal (within the tolerance EPS) it will transfer to the output routine and return to the main calling program.

If the variance criteria are not equal the next order polynomial will be computed utilizing the next order solution vectors ALPHA and BETA. After each order polynomial has been computed it is stored in the array POL. A test is made to determine if four polynomials have been stored. If so, the array POL is printed along with the input values contained in ID, X and Y. Also printed for each X(I), Y(I) are the cumulative predicted values from YA(I) and their difference.

The title information, job number, page number, and column headings are printed at the top of each output page. The current solution vectors are also printed at the bottom of each page.

At the conclusion of the output stage, the current variance criterion is stored in the previous criterion location and the polynomial order II is incremented. The program then branches back to initiate another cycle.

After either the variance criterion has been satisfied or the maximum degree of the polynomial (as determined by the user) has been reached, the program tests the input parameter IALP to determine if the final solution vectors are to be punched. If punching has been requested, the vectors are punched, with a matrix identification number, row and column number in the standard matrix punch output format. Also, necessary scaling constants are punched.

Upon entry to the program the data is stored in array X and Y. The number of data cases are in location NCASE and all necessary common parameters are located in COMMON storage.

On exit from the program, the solution vectors ALPHA, BETA and C are located in COMMON storage and the degree is that of the resultant polynomial which either satisfied the variance criterion or is the input parameter **N** which is located at NF. Arrays X, Y, ID and location NCASE have not changed.

SUBROUTINE NAME: PCOEF

CALLING PROGRAM: POL2

This subroutine is used to compute the coefficients of the fitted polynomial from the polynomial solution vectors.

After initialization, the program computes orthogonal polynomials using the solution vectors ALPHA and BETA. From the orthogonal polynomial the coefficients of the fitted polynomial are computed by multiplying the solution vector C by this polynomial. This process continues until the degree NF+1 is reached.

After the computation is completed, the coefficients are printed with title and heading information.

Upon entry to the program the solution vectors are contained in arrays ALPHA, BETA and C. The degree of the polynomial is located at NF.

Upon exit from the program, the polynomial solution vectors ALPHA, BETA and C are in common storage and the degree of the polynomial is at location NF.

SUBROUTINE NAME: PDER

CALLING PROGRAM: POL2

This subroutine is used to compute the derivative of the fitted polynomial at a given point. The program begins by examining the identification vector ID for a negative value. If ID(I) is positive, I is incremented and another identification value is examined. This will continue until I is equal to NCASE in which case control will be returned to the main calling program.

If, for a given I, ID(I) is negative, the value of X for this I will be stored in XB and other derivative computations for this point started by initializing the computational arrays and parameters. The parameter NN represents the order of the derivative to be computed and is initially set equal to 1.

The program then computes the NNth order derivative by utilizing the polynomial solution vectors ALPHA, BETA and C to compute the orthogonal polynomial DOPOL.

As each order polynomial is computed a recurrence solution is utilized to build up the value of the derivative and the next order. When NN is equal to NDD1, the order of the requested derivative, the program will print line or lines containing the identification ID(I), the value of

X(I), the value of Y(I), the order of the derivative and its value. Each page will also contain title and column headings.

After the derivative for a point has been printed the program will transfer back to examine another ID(I) for a negative value.

Upon entry to the program, the array ID contains identification values, X, Y contain data and ALPHA, BETA and C contain the polynomial solution vectors. NF contains the degree of the polynomial, NCASE the number of data points and NDER the order of the derivatives to be computed.

Upon exit from the program the derivatives for all points indicated in the ID vector have been printed, and the polynomial solution vectors ALPHA, BETA and C are in their respective arrays.

SUBROUTINE NAME: PFIT

CALLING PROGRAM: POL2

This subroutine is used primarily to compute predicted values from a set of data values X(I) that are different than those used to compute the initial polynomial. After initialization the program uses the solution vectors ALPHA, BETA and C to compute orthogonal polynomials.

As each order orthogonal polynomial is generated the cumulative predicted value is computed from X(I) and stored in the array YA. After NCASE values of YA are computed the program will print the predicted values, with identification, the actual value of Y, and the difference. Title and column headings will also appear on each page.

Upon entry to the program the solution vectors are in ALPHA, BETA and C. The data points are located in X and Y and the degree of polynomial is in NF. The number of data points is located at NCASE.

Upon exit from the program the predicted values for all data points have been printed.

F. Routines Used by All System Programs

The following five routines in assembly language allow user-specified format statements at object program time. Of the routines called by these, CARDZ, PRNTZ, NORM, IFIX, TYPE Z, and FSTOX are utility routines available to the assembler.

SUBROUTINE FMTRD

FMTRD reads one card containing a format and stores it in a form suitable for the subroutine DATRD.

Calling sequence:

CALL FMTRD (FORMT, ERROR)

FORMT must be an integer vector fifty (50) words long. ERROR is an integer word.

Upon return, FORMT contains the translated format and ERROR will be zero. If the translation was completed, ERROR will be the next column to be processed if an error was detected. When an error is detected no attempt is made to complete the translation and format may have to be changed.

Format codes: The following specifications are acceptable:

wX nIw nFw.d nEw.d

n may be omitted if it is one. One level of parentheses is allowed for group repetition. In addition, parentheses are required around the entire format. Every specification, including wX and parenthesized groups, must be followed by either a comma or a right parenthesis. Multiple record formats (/), scaling (P) and alphabetic conversion (A, H and I) are not available. In addition, the format must be completed on one card.

Length: 225

Subroutines required: CARD Z

SUBROUTINE PRNTB

PRNTB prints the I/O buffer after a previous read statement.

Calling sequence:

CALL PRNTB

Function and use: When called, PRNTB prints the first eighty positions of the I/O buffer on the printer with a double space. It may be used after a call to FMTRD or DATRD, whether or not an error occured, to print the card just processed. It may also be called after a normal card read statement. No I/O statements may intervene between a call to PRNTB and the associated read statement.

Length: 16

Subroutines used: PRNT Z, TYPE Z

SUBROUTINE DATRD

DATRD reads one card of data according to a format previously stored by FMTRD.

Calling sequence: CALL DATRD (FORMT, ERROR, VAR1, N1, VARZ, NZ, ..., 0, 0)

FORMT is an integer vector fifty words long previously named in a call to FMTRD. ERROR is an integer word. VAR1, VAR2, etc. are integer or real variables or vectors. N1, N2, etc. are integer variables or constants. Each is positive if the corresponding variable is integer, negative if real.

Upon return, the first N_i locations of each VAR_i are replaced by data. Automatic type conversion from I specification to real and from E or F specification to integers is performed. If no error is detected, ERROR is set to zero; otherwise it is set to the next column to be processed. The error is either in the specified column or in the preceding field. None of the N_i may be zero. Two zeros end the list of variables.

Data: Only one data card can be read by this routine. An attempt to read beyond the end of the format is treated as an error. Numbers may have any number of leading or trailing blanks. Signs may have leading and trailing blanks. If the sign is omitted, it is assumed to be positive. For F and E conversions, a decimal point is allowed; if omitted it is implied by the format. E type numbers may have an exponent part which must start with an E, a blank or a sign. Blanks may not precede the E. If the exponent minus the number of decimals (explicit or implicit) is not in the range $\pm$ 63, an error is indicated. If the absolute value of the number ignoring the decimal point and exponent is greater than $2^{31}-1$, the result will be incorrect with no error indication given. An overflow or underflow condition is possible and is ignored.

Length: 350

Subroutines required: CARDZ, NORM, GMPYX, GDIVX, IFIX, FSTOX

SUBROUTINE NAME: GMPYX

GMPYX is equivalent to EMPYX, from the IBM 1130 FORTRAN Library.

SUBROUTINE NAME: GDIVX

GDIVX is equivalent to EDIVX, from the IBM 1130 FORTRAN Library. (GMPYX and GDIVX are required by DATRD in a form accessible to assembly language routines.)

The following routine is written in FORTRAN.

SUBROUTINE NAME: FMAT

FMAT is called to allow correct output from the typewriter or printer; when a format statement is handled by the typewriter, the carriage control character is printed unless FMAT is used.

The following routine is called on user option by each of the four system programs. It is included to aid the user in preparing a program for variable transformation. The User's Manual which is distributed with the 1130 Statistical System discusses such a program.

SUBROUTINE TRAN

TRAN is a user written subroutine which currently returns to the calling program.

4.0 PROGRAMMING NOTES

An experienced system user may desire to modify sections of the package. For example, larger arrays could be analyzed by modifying dimension statements, primarily those evident in COMMON. Such revision may be desirable in Regression and Factor Analysis, and may require that the number of main linkages be increased, to provide adequate storage facilities. In Orthogonal Polynomials, if scaling is used, and the user desires the original coefficients for his polynomial, those for X, rather than X', another link could be written to provide these. Considerable care should be taken concerning accuracy, so that the same problems do not arise that were bothersome in the unscaled situation.

In Factor Analysis, if the user retains the correlation matrix by saving it on the disk throughout the calculation, then factor scores could be calculated by the direct method, rather than the short method. The short method only calls for inversion of an m-by-m matrix, where m is the number of factors rotated. Modifications to allow direct estimation will require revision of links FCTR1, FCTR2, and FCTR3.

The following table gives core requirements for each program in the 1130 Statistical System using the 1130 Disk Monitor System, Mod. Level 2.

Program	Variables	Common	Program	Total
FMTRD, PRNTB DATRD, GMPY GDIV				578
TRAN	0	0	4	4
MXRAD	14	2142	234	2390
COREL	24	2262	656	2942
PRNT	8	2142	668	2818
PCOEF	14	1540	292	1846
POLY	32	1182	1024	2238
POL2	8	3232	62	3302
POLSQ	40	3232	1042	4314
PFIT	14	2032	318	2364
PDER	24	1438	432	1894
REGR	24	2412	1378	3814
REGR2	8	2262	48	2318
REGRE	112	2262	1902	4276
ANOVA	28	3166	724	3918
STORE	2	3166	44	3212
GET	2	3166	42	3210
ANOV2	14	4166	44	4224
SDOP	16	3166	206	3388

Program	Variables	Common	Program	Total
MNSQ	10	3166	348	3524
REPRT	34	3206	690	3930
FCTR	30	2412	1782	4224
FCTR1	20	2264	252	2536
INVRS	10	0	322	332
XMAX	2	0	28	30
TRIDI	156	2264	826	3246
QR	154	2264	638	3056
FCTR2	88	2264	480	2832
RFOUT	6	1342	596	1944
PROMX	12	1362	578	1952
VARMX	76	1142	1068	2286
VECTR	136	2264	530	2930
COVEC	196	2264	322	2782
FCTR3	14	1362	66	1442
RPRNT	8	942	808	1758
MATIN	72	0	482	554
SCORE	16	1162	822	2000
FMAT	0	0	34	34

5.0 LIST OF SWITCHES

One console entry switch is used by the 1130 Statistical System. If switch 15 is off (down), then each time a program punches cards, a message reminds the user to supply blank cards. This reminder can be suppressed by turning switch 15 on (up).

6.0 PROGRAM LISTINGS

```
// ASM   READ VARIABLE FORMAT
* READ AND DECODE FORMAT CARDS
*
      ENT      FMTRD
*
LPREN DC
      BSI      READ
      DC       .(
      MDX      *+1
      MDX      *+3
      BSI      READ
      DC       .(
      MDX      *+2
      MDX   L  LPREN,1
      BSC   I  LPREN
*
RPREN DC
      BSI      READ
      DC       .)
      MDX      *+1
      MDX      *+3
      BSI      READ
      DC       .)
      MDX      *+2
      MDX   L  RPREN,1
      BSC   I  RPREN
*
FMTRD DC
      STX   1  FMTEX-1
      STX   2  FMTEX-3
      STX   3  FMTEX-5
      LD       ZERO
      LIBF     CARDZ
      LDX   I1 FMTRD
      LD    1  0
      A        ONE
      STO      STORE-2
      LD    1  1
      STO      FMTEX-7
      MDX   1  2
      STX   1  FMTEX+1
      LDX   1  /3C
      LDX   2  -51
      LD       NUL
      STO   1  80
*
BEGIN BSI      LPREN
      MDX      FMTER
      LD       AD5
      BSI      STORE
ELEM  BSI      NUMBR
      NOP
      BSI      LPREN
      MDX      ELEN-2
      LD       NUM
      EOR      ONE
      STO      SW
      BSC   L  REP,+-
      EOR      AD2
      BSI      STORE
      STX   2  HOLD
REP   BSI      NUMBR
      NOP
      BSI      SPCIF
      MDX      FMTER
      BSI      READ
      DC       .,
      MDX      *+1
      MDX      REP
      BSI      RPREN
      MDX      FMTER
      LD       SW
      BSC   L  ELEN,+-
      LD       AD3
      A        HOLD
      BSI      STORE
      MDX      ELEN
      BSI      SPCIF
      MDX      FMTER
ELEN  BSI      READ
      DC       .,
      MDX      *+1
      MDX      ELEM
      BSI      RPREN
      MDX      FMTER
      LD       AD4
      BSI      STORE
      LDX   1  0
      STX   L1
      LDX   L3
      LDX   L2 0
      LDX   L1 0
FMTEX BSC   L  0
*
READ  DC
      BSI      GETCL
      LDX   I3 READ
      EOR   3
      BSC   L  *+2,Z
      MDX   1  1
      MDX   3  1
      BSC   L3 1
*
      STO   L2
      BSC   L
STORE EQU      *-1
      MDX   2  1
      MDX      STORE-3
*
FMTER MDX   1  1-/3C
```

```
      MDX     FMTEX-8                                              FMRD1100
*                                                                  FMRD1110
NUM   DC                                                           FMRD1120
AD1   DC      /0081                                                FMRD1130
AD2   DC      /0101                                                FMRD1140
AD3   DC      /0180+51                                             FMRD1150
AD4   DC      /0200                                                FMRD1160
OP0   DC      0                                                    FMRD1170
OP1   DC      /4000                                                FMRD1180
OP2   DC      /8000                                                FMRD1190
OP3   DC      /C000                                                FMRD1200
ONE   DC      1                                                    FMRD1210
WORK  DC                                                           FMRD1220
AD5   DC      /0281                                                FMRD1230
SW    DC                                                           FMRD1240
HOLD  DC                                                           FMRD1250
ZERO  EQU     OP0                                                  FMRD1260
NUL   EQU     ZERO                                                 FMRD1270
CHZER DC      .0                                                   FMRD1280
BLNK  DC                                                           FMRD1290
*                                                                  FMRD1300
NUMBR DC                                                           FMRD1310
      LD      ONE                                                  FMRD1320
      STO     NUM                                                  FMRD1330
      BSI     DIGIT                                                FMRD1340
      MDX     NUMEX                                                FMRD1350
      STO     NUM                                                  FMRD1360
      BSI     DIGIT                                                FMRD1370
      MDX     NUMEX-3                                              FMRD1380
      LD      NUM                                                  FMRD1390
      SLA     2                                                    FMRD1400
      A       NUM                                                  FMRD1410
      SLA     1                                                    FMRD1420
      A       DIG                                                  FMRD1430
      STO     NUM                                                  FMRD1440
      LD      NUM                                                  FMRD1450
      MDX  L  NUMBR,1                                              FMRD1460
NUMEX BSC  I  NUMBR                                                FMRD1470
*                                                                  FMRD1480
SPCIF DC                                                           FMRD1490
      BSI     READ                                                 FMRD1500
      DC      .X                                                   FMRD1510
      MDX     *+2                                                  FMRD1520
      LD      OP0                                                  FMRD1530
      MDX     SPCEX-4                                              FMRD1540
      LD      NUM                                                  FMRD1550
      EOR     ONE                                                  FMRD1560
      BSC  L  *+2,+-                                               FMRD1570
      EOR     AD1                                                  FMRD1580
      BSI     STORE                                                FMRD1590
      BSI     READ                                                 FMRD1600
      DC      .I                                                   FMRD1610
      MDX     *+5                                                  FMRD1620
      BSI     NUMBR                                                FMRD1630
      MDX     FMTER                                                FMRD1640
      SLA     7                                                    FMRD1650
      OR      OP1                                                  FMRD1660
      MDX     SPCEX-3                                              FMRD1670
      BSI     READ                                                 FMRD1680
      DC      .F                                                   FMRD1690
      MDX     *+2                                                  FMRD1700
      LD      OP2                                                  FMRD1710
      MDX     *+4                                                  FMRD1720
      BSI     READ                                                 FMRD1730
      DC      .E                                                   FMRD1740
      MDX     SPCEX                                                FMRD1750
      LD      OP3                                                  FMRD1760
      STO     WORK                                                 FMRD1770
      BSI     NUMBR                                                FMRD1780
      MDX     FMTER                                                FMRD1790
      SLA     7                                                    FMRD1800
      OR      WORK                                                 FMRD1810
      STO     WORK                                                 FMRD1820
      BSI     READ                                                 FMRD1830
      DC      ..                                                   FMRD1840
      MDX     FMTER                                                FMRD1850
      BSI     NUMBR                                                FMRD1860
      MDX     FMTER                                                FMRD1870
      LD      WORK                                                 FMRD1880
      OR      NUM                                                  FMRD1890
      BSI     STORE                                                FMRD1900
      MDX  L  SPCIF,1                                              FMRD1910
SPCEX BSC  I  SPCIF                                                FMRD1920
*                                                                  FMRD1930
GETCL DC                                                           FMRD1940
      LD    1 0                                                    FMRD1950
      EOR     BLNK                                                 FMRD1960
      BSC  L  *+2,Z                                                FMRD1970
      MDX   1 1                                                    FMRD1980
      MDX     GETCL+1                                              FMRD1990
      EOR     BLNK                                                 FMRD2000
      BSC  I  GETCL                                                FMRD2010
*                                                                  FMRD2020
DIG   DC                                                           FMRD2030
DIGIT BSS     1                                                    FMRD2040
      BSI     GETCL                                                FMRD2050
      S       CHZER                                                FMRD2060
      BSC  L  DIGEX,+Z                                             FMRD2070
      STO     DIG                                                  FMRD2080
      MDX   1 1                                                    FMRD2090
      MDX  L  DIGIT,1                                              FMRD2100
DIGEX BSC  I  DIGIT                                                FMRD2110
      END                                                          FMRD2120
// DUP                                                             FMRD2130
*STORE      WS  UA  FMTRD                                          FMRD2140
```

```
// ASM
* PRINT I-O BUFFER (80 CHARACTERS, DOUBLE SPACE)
* PRINT I-O BUFFER (80 CHARACTERS, DOUBLE SPACE)
*
       ENT        PRNTB
*
PRNTB  DC
       STX    2 SVE+1
       LDX    2 80
       LD     2 /3C-1
       STO    2 /3C
       MDX    2 -1
       MDX      *-4
       LD       CN1
       STO    2 /3C
       LDX    2 81
       LIBF     PRNTZ
SVE    LDX   L2
       BSC    I PRNTB
CN1    DC       .0
       END
// DUP
*STORE    03WS  UA  PRNTB
```

```
// ASM
* READ DATA ACCORDING TO FORMAT STATEMENT
* READ DATA ACCORDING TO FORMAT STATEMENT
*
       ENT        DATRD
DBL    A     I2 1
       STO    3 125
       A      2 0
       STO      *+7
       SLA      16
       S      3 125
       STO   L  1
       BSI      FMTEN
       MDX      DATER
       LIBF     FSTOX
       DC       0
       MDX    1 -2
       MDX      *-6
       MDX      LIST
*
DATER  MDX    2 2
       LD    I2 1
       BSC   L  DATER,Z
       LD       COLM+1
       S        CN1
*
       MDX    2 2
       STX    2 DATEX+1
       STO   L  0
       LDX   L2 0
       LDX   L1 0
DATEX  BSC   L  0
*
DATRD  DC
       STX    1 DATEX-1
       STX    2 DATEX-3
       SRA      16
       STO      SPEC+1
       LDX    1 /3C
       STX    1 COLM+1
       STO    1 80
       LIBF     CARDZ
       LDX   I2 DATRD
       LD     2 0
       STO      SPEC+3
       MDX   L  SPEC+3,-49
       LD     2 1
       STO      DATEX-5
*
LIST   MDX    2 2
       LD    I2 1
       BSC   L  DATEX-8,+-
       BSC   L  DBL,+
       STO   L  1
       LD     2 0
       S     L  1
```

```
        STO       *+4                                   DTRD 550
        BSI       FMTEN                                 DTRD 560
        MDX       DATER                                 DTRD 570
        LIBF      IFIX                                  DTRD 580
        STO   L1  0                                     DTRD 590
        MDX    1  -1                                    DTRD 600
        MDX       *-7                                   DTRD 610
        MDX       LIST                                  DTRD 620
*                                                       DTRD 630
CN1     DC        /3C-1                                 DTRD 640
XR1     DC                                              DTRD 650
XR2     DC                                              DTRD 660
*                                                       DTRD 670
MISC    LDX   L1  BRTB+4                                DTRD 680
        SLA       9                                     DTRD 690
        SLT       7                                     DTRD 700
        BSC   I1                                        DTRD 710
WIDTH   EQU       *-1                                   DTRD 720
XTYPE   A         COLM+1                                DTRD 730
        STO       COLM+1                                DTRD 740
        MDX       SPEC                                  DTRD 750
AXT1    STO       XR1                                   DTRD 760
        MDX       SPEC                                  DTRD 770
AXT2    STO       XR2                                   DTRD 780
        MDX       SPEC                                  DTRD 790
TIX2    MDX   L   XR2,-1                                DTRD 800
        STO       SPEC+1                                DTRD 810
        MDX       SPEC                                  DTRD 820
INIT    EQU       AXT1                                  DTRD 830
FMTEN   DC                                              DTRD 840
        STX    1  FMTEX+1                               DTRD 850
        STX    2  FMTEX+3                               DTRD 860
SPEC    LDX   L1                                        DTRD 870
        LD    L1  0                                     DTRD 880
        MDX   L   XR1,-1                                DTRD 890
        MDX       *+3                                   DTRD 900
        MDX   L   XR1,1                                 DTRD 910
        MDX    1  1                                     DTRD 920
        STX    1  SPEC+1                                DTRD 930
        SRT       14                                    DTRD 940
        STO   L   1                                     DTRD 950
        SLA       9                                     DTRD 960
        SLT       7                                     DTRD 970
        STO       WIDTH                                 DTRD 980
        BSC   I1  BRTB+2                                DTRD 990
*                                                       DTRD1000
GETCL   DC                                              DTRD1010
        LD        WIDTH                                 DTRD1020
        BSC   L   *+4,+                                 DTRD1030
COLM    LD    L                                         DTRD1040
        MDX   L   GETCL,1                               DTRD1050
        BSC   I   GETCL                                 DTRD1060
*                                                       DTRD1070
BLNKS   DC                                              DTRD1080
        BSI       GETCL                                 DTRD1090
        LDX       *+5                                   DTRD1100
        EOR       BLNK                                  DTRD1110
        BSC   I   BLNKS,Z                               DTRD1120
        BSI       STPCL                                 DTRD1130
        MDX       BLNKS+1                               DTRD1140
        MDX   L   BLNKS,1                               DTRD1150
        MDX       BLNKS+3                               DTRD1160
BLNK    DC                                              DTRD1170
*                                                       DTRD1180
STPCL   DC                                              DTRD1190
        MDX   L   COLM+1,1                              DTRD1200
        MDX   L   WIDTH,-1                              DTRD1210
        NOP                                             DTRD1220
        BSC   I   STPCL                                 DTRD1230
*                                                       DTRD1240
CHZER   DC        .0                                    DTRD1250
NUMEX   DC                                              DTRD1260
        BSI       GETCL                                 DTRD1270
        MDX       NUMXX                                 DTRD1280
        S         CHZER                                 DTRD1290
        BSC   L   NUMXX,+Z                              DTRD1300
        STO       DIG+1                                 DTRD1310
        BSI       STPCL                                 DTRD1320
CNTSW   MDX   L   COUNT,0                               DTRD1330
        LDD       NUM                                   DTRD1340
        SLT       2                                     DTRD1350
        AD        NUM                                   DTRD1360
        SLT       1                                     DTRD1370
SGN     AD        DIG                                   DTRD1380
        STD       NUM                                   DTRD1390
        MDX       NUMEX+1                               DTRD1400
NUMXX   BSC   I   NUMEX                                 DTRD1410
*                                                       DTRD1420
        STO       COUNT                                 DTRD1430
OP      DC                                              DTRD1440
        DC        TABLE+18                              DTRD1450
        LD        COUNT                                 DTRD1460
        MDX       SCL+2                                 DTRD1470
SCALE   LD        EDIVX                                 DTRD1480
        STO       OP                                    DTRD1490
        LD        COUNT                                 DTRD1500
        BSC   L   *+4,-                                 DTRD1510
        LD        EMPYX                                 DTRD1520
        STO       OP                                    DTRD1530
        LD        ZERO                                  DTRD1540
        S         COUNT                                 DTRD1550
        LDX    1  -18                                   DTRD1560
SCL     BSC   L   OP-1,E                                DTRD1570
        SRA       1                                     DTRD1580
        MDX    1  3                                     DTRD1590
        MDX       SCL                                   DTRD1600
        BSC       +                                     DTRD1610
CMMN    BSI       BLNKS                                 DTRD1620
        MDX       FMTEX                                 DTRD1630
        MDX   L   FMTEN,1                               DTRD1640
```

```
FMTEX LDX  L1 0                               DTRD1650
      LDX  L2 0                               DTRD1660
      BSC  I  FMTEN                           DTRD1670
HLT   EQU     FMTEX                           DTRD1680
*                                             DTRD1690
DIG   DEC     0                               DTRD1700
ZERO  DEC     0                               DTRD1710
*                                             DTRD1720
READ  DC                                      DTRD1730
      LDX  I2 READ                            DTRD1740
      BSI     GETCL                           DTRD1750
      MDX     *+5                             DTRD1760
      EOR   2 0                               DTRD1770
      BSC  L  *+2,Z                           DTRD1780
      BSI     STPCL                           DTRD1790
      MDX   2 1                               DTRD1800
      BSC  L2 1                               DTRD1810
*                                             DTRD1820
NUM   DEC     0                               DTRD1830
      MDX  L  NUMBR,1                         DTRD1S40
      BSC  L                                  DTRD1850
NUMBR EQU     *-1                             DTRD1860
      BSI     BLNKS                           DTRD1870
      NOP                                     DTRD1880
      LDD     ZERO                            DTRD1890
      STD     NUM                             DTRD1900
      LD      SWOFF                           DTRD1910
      STO     CNTSW                           DTRD1920
      BSI     NUMEX                           DTRD1930
      BSI     READ                            DTRD1940
      DC      ..                              DTRD1950
      MDX     NUMBR-3                         DTRD1960
      STO     COUNT                           DTRD1970
      LD      SWON                            DTRD1980
      STO     CNTSW                           DTRD1990
      BSI     NUMEX                           DTRD2000
      MDX     NUMBR-1                         DTRD2010
*                                             DTRD2020
EMPYX LIBF    GMPYX                           DTRD2030
EDIVX LIBF    GDIVX                           DTRD2040
COUNT DC                                      DTRD2050
SWON  DC      /7401       MDX  L   ,1         DTRD2060
ISWOF NOP                                     DTRD2070
ISWON MDX  X  FMTEX-ISW-1                     DTRD2080
EXP   DC      159                             DTRD2090
SWOFF DC      /4C38       BSC  L   ,+Z-       DTRD2100
*                                             DTRD2110
ETYPE LD      ISWOF                           DTRD2120
      BSI     FFIX                            DTRD2130
      BSI     READ                            DTRD2140
      DC      .E                              DTRD2150
      MDX     *+3                             DTRD2160
      BSI     SIGN                            DTRD2170
      NOP                                     DTRD2180
      MDX     *+2                             DTRD2190
```

```
      BSI     SIGN                            DTRD2200
      MDX     SCALE                           DTRD2210
      BSI     NUMBR                           DTRD2220
      MDX     FMTEX                           DTRD2230
      LD      COUNT                           DTRD2240
      S       NUM+1                           DTRD2250
      STO     COUNT                           DTRD2260
      MDX     SCALE                           DTRD2270
*                                             DTRD2280
ITYPE LD      ISWON                           DTRD2290
      MDX     FTYPE+1                         DTRD2300
FTYPE LD      ISWOF                           DTRD2310
      BSI     FFIX                            DTRD2320
      MDX     SCALE                           DTRD2330
*                                             DTRD2340
PLUS  AD   X  DIG-SGN-1                       DTRD2350
MINUS SD   X  DIG-SGN-1                       DTRD2360
      LD      MINUS                           DTRD2370
      STO     SGN                             DTRD2380
      LDX   1 1                               DTRD2390
      BSC  L1 0                               DTRD2400
SIGN  EQU     *-1                             DTRD2410
      LD      PLUS                            DTRD2420
      STO     SGN                             DTRD2430
      LDX   1 0                               DTRD2440
      BSI     READ                            DTRD2450
      DC      .                               DTRD2460
      MDX     *+2                             DTRD2470
      LDX   1 1                               DTRD2480
      MDX     *-5                             DTRD2490
      BSI     READ                            DTRD2500
      DC      .+                              DTRD2510
      MDX     *+1                             DTRD2520
      MDX     SIGN-2                          DTRD2530
      BSI     READ                            DTRD2540
      DC      .+                              DTRD2550
      MDX     *+1                             DTRD2560
      MDX     SIGN-2                          DTRD2570
      BSI     READ                            DTRD2580
      DC      .-                              DTRD2590
      MDX     SIGN-1                          DTRD2600
      MDX     SIGN-4                          DTRD2610
*                                             DTRD2620
FFIX  DC                                      DTRD2630
      STO     ISW                             DTRD2640
      SLA     9                               DTRD2650
      SLT     7                               DTRD2660
      STO     COUNT                           DTRD2670
      BSI     SIGN                            DTRD2680
      NOP                                     DTRD2690
      BSI     NUMBR                           DTRD2700
ISW   MDX     FMTEX                           DTRD2710
      LDD     NUM                             DTRD2720
      STD   3 126                             DTRD2730
      LD      EXP                             DTRD2740
```

```
      STO   3 125
      LIBF    NORM
FFIXX BSC  I  FFIX
*
BRTB  DC      FTYPE
      DC      ETYPE
      DC      MISC
      DC      ITYPE
      DC      XTYPE
      DC      AXT1
      DC      AXT2
      DC      TIX2
      DC      HLT
      DC      INIT
TABLE DC      /0084              10.E01,TRUNCATED
      DC      /5000
      DC      /0000
      DC      /0087              10.E02,TRUNCATED
      DC      /6400
      DC      /0000
      DC      /009E              10.E04,TRUNCATED
      DC      /4E20
      DC      /0000
      DC      /009B              10.E08,TRUNCATED
      DC      /5F5E
      DC      /1000
      DC      /00B6              10.E16,ROUNDED
      DC      /470D
      DC      /E4E0
      DC      /00EB              10.E32,TRUNCATED
      DC      /4EE2
      DC      /D6D4
      END
// DUP
*STORE      WS  UA  DATRD
```

```
// ASM
*EMPY/EMPYX--EXTENDED PRECISION FLOAT MULTIPLY
*EMPY/EMPYX--EXTENDED PRECISION FLOAT MULTIPLY
      LIBR
      ENT      GMPYX
EMPYX STX   1 EMX1+1     SAVE XR1
GMPYX EQU     EMPYX
      LD   L  *-*        LOADER INSERT.
EMC   STO     *+3
      A       MCN+1      =1   SET UP EXIT.
      STO     MX+1
      MDX  I1 *-*        OPND ADDRESS INTO XR1.
      NOP
CMN   LD    3 125        COMPUTE PRODUCT EXPONENT.
      A     1 0
      S       MCN        =128
      STO   3 125
      LD    1 2          PICK UP ARG FRACTION.
      RTE     16
      LD    1 1
      LIBF    XMD        MULTIPLY FRACTIONS.
      STD   3 126
      BSC     +-
      STO   3 125
      SLT     1
      EOR   3 126
      BSC  L  *+5,+
      EOR   3 126
      STD   3 126
      LD    3 125
      S       MCN+1      =1
      STO   3 125
EMX1  LDX  L1            RESTORE XR1.
      LIBF    FARC
MX    BSC  L  *-*        EXIT.
MCN   DC      128        0
      DC      1          1
      END
// DUP
*STORE    02WS  UA  GMPYX
```

```
// ASM
*EDIV/EDIVX--EXTENDED PRECISION FLOAT DIVIDE
*EDIV/EDIVX--EXTENDED PRECISION FLOAT DIVIDE
      LIBR
      ENT       GDIVX
EDIVX STX    1 EDX1+1      SAVE XR1
GDIVX EQU      EDIVX
      LD    L  *-*         LOADER INSERT.
EDC   STO      *+3
      A        ONE+1       =1   SET UP EXIT.
      STO      EDX1+3
      MDX   I1 *-*         OPND ADDRESS INTO XR1.
      NOP
      LD     1 2
      RTE      16
      LD     1 1
      BSC   L  DOVL,+-     CHECK X/0.
      STD      DVR
      LD     3 126
      BSC   L  EDX1,+-     DIVIDEND ZERO TEST.
      EOR    1 1
      AND      EDCN        =/8000
      STO      QSGN        SIGN OF QUOTIENT.
      BSC   L  *+3,+Z
      LDD    3 126         SUBTRACT MAG. OF DIVISOR
      SD       DVR         FROM DIVIDEND MAGNITUDE,
      MDX      *+2         TO ENSURE DIVIDEND SMALLER
      LDD    3 126         THAN DIVISOR.
      AD       DVR
      STD    3 126
      OR     3 127
      BSC   L  *+3,Z
      LDD      DF1
      OR       QSGN
      MDX      X
      LDD      DVR
      LIBF     XDD
      EOR      EDCN        =/8000
      STD    3 126
      EOR      QSGN
      BSC   L  *+9,-
      EOR      QSGN
      BSC      -
      AD       ONE
      SRT      1
      EOR      EDCN        =/8000
X     STD    3 126
      LD     3 125
      A        ONE+1
      STO    3 125
      LD     3 125         COMPUTE QUOTIENT EXPONENT.
      S      1 0
      A        EDCN+1      =128
OVL   STO    3 125
      LIBF     FARC
EDX1  LDX   L1 *-*         RESTORE XR1.
      BSC   L  *-*         EXIT.
DOVL  LD       ONE+1       TURN ON PROGRAM DIVIDE
      STO    3 123         CHECK INDICATOR.
      MDX      EDX1
QSGN  DC       0
ONE   DEC      1
DVR   DEC      0           DIVISOR BUFFER.
DF1   DEC      1.0B1
EDCN  DC       /8000       0
      DC       128         1
      END
// DUP
*STORE    02WS  UA  GDIVX
```

```
// FOR DUMMY SUBROUTINE FOR  TRANSFORMATIONS
*ONE WORD INTEGERS
C      DUMMY SUBROUTINE FOR  TRANSFORMATIONS
       SUBROUTINE TRAN
       RETURN
       END
// DUP
*STORE       WS  UA  TRAN
```

```
// FOR  SUBROUTINE TO READ AND ADD MATRICES
*ONE WORD INTEGERS
C      SUBROUTINE TO READ AND ADD MATRICES
       SUBROUTINE MXRAD
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,IPRED,
      1ISTEP,ICNST,IREAR,KX(1),MX(20),NCD1,NCD2,NCD3,ISEQ,NCASE,NX(10),
      2 EFOUT,EFIN,TOL,FLVB(2),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
  101 FORMAT(I4,3I2,5E14.7)
       IKT=0
    9 READ(ICR,101) IP,MID,IC,IR,(X(I),I=1,5)
       IF(IKT)51,52,51
   51 DO 53 I=1,5
   53 X(I)=-X(I)
   52 IF(IP) 30,30,10
   10 IF(MID-21) 11,15,20
C      STORE MATRIX
   11 J1 = IC
       J2 = IC+4
       IF (J2-N) 14,14,13
   13 J2=N
   14 K=0
       MXT = MID
       DO 12 J=J1,J2
       K = K+1
   12 R(IR,J) = R(IR,J) + X(K)
       GO TO 9
C      STORE NUMBER OF CASES
   15 CASES = CASES + X(1)
       GO TO 9
C      STORE MEANS AND STANDARD DEVIATIONS
   20 SUMY(IR) = SUMY(IR) + X(1)
       SD(IR) = SD(IR) + X(2)
       GO TO 9
   30 IF(MXT-1) 31,31,32
   31 NCASE=MXT
    1 IF(ICNST)50,35,50
   50 ICNST=0
       IKT=1
       GO TO 9
   32 IF (MXT-4) 34,34,35
   34 NCASE=-MXT
       GO TO 1
   35 RETURN
       END
// DUP
*STORE       WS  UA  MXRAD
```

```
// FOR  SUBROUTINE TO COMPUTE CORRELATION COEFFICIENTS
*IOCS(CARD,1132PRINTER,DISK)
*NAME COREL
*ONE WORD INTEGERS
C      SUBROUTINE TO COMPUTE CORRELATION COEFFICIENTS
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,KX(5),
      1MX(20),NX(15),FLVB(5),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,30)
       COMMON HIGH(30),HLOW(30)
       DEFINE FILE 606(500,65,U,IT1)
       DEFINE FILE 5(30,60,U,IT2)
  103 FORMAT(    10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6)
  104 FORMAT(//11X18HSUMMARY STATISTICS10X12HNO.OF CASES=,I6,//16X8HVARI
     1ABLE16X3HLOW9X4HHIGH9X7HAVERAGE7X9HSTD. DEV.6X8HVARIANCE///)
  105 FORMAT(16X,I2,2X,A4,5X,5E15.5)
  106 FORMAT(/1X,24HTHE VARIANCE OF VARIABLE,1XA4,1X7HIS ZERO   )
  107 FORMAT(I4,3I2,5E14.7)
  108 FORMAT(1H )
C       SUM OF SQUARES TAKEN FROM X-PROD MATRIX
    1 ICA=CASES
      ISW = 1
      KON = 22
      KON1 = 1
      DO 10 I=1,N
   10 SD(I) = R(I,I)
      CALL PRNT(1,1,N,N)
      IF(MX(1)-2) 15,91,91
C     COMPUTE RESIDUAL X-PROD MATRIX
   15 DO 20 I=1,N
      DO 20 J=I,N
      R(I,J) =         R(I,J) - SUMY(I)*SUMY(J )/CASES
   20 R(J,I) = R(I,J)
      CALL PRNT(2,0,N,N)
C     COMPUTE COVARIANCE MATRIX
      DO 30 I=1,N
      DO 30 J=I,N
      R(I,J) = R(I,J)/(CASES-1.)
   30 R(J,I) = R(I,J)
      CALL PRNT(3,0,N,N)
C     OUTPUT MEANS AND STANDARD DEVIATIONS
      ISW = 2
      KON = 23
      NPAGE = NPAGE + 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 31,31,32
 31   WRITE(ITW,103)TITLE,IPROB,NPAGE
   32 WRITE(ITW,104)ICA
      DO 40 I=1,N
      SUMY(I) = SUMY(I)/CASES
      SD(I) = SQRT(R(I,I))
   40 WRITE(ITW,105) I,VNAME(I),HLOW(I),HIGH(I),SUMY(I),SD(I),R(I,I)
C     COMPUTE CORRELATION MATRIX
   45 DO 90 I=1,N
      IF(SD(I)) 50,50,60
   50 WRITE(ITW,106) VNAME(I)
   60 DO 90 J=I,N
      IF(SD(J)) 70,70,80
   70 R(I,J) = 0.0
      GO TO 90
   80 R(I,J) = R(I,J)/(SD(I)*SD(J))
   90 R(J,I) = R(I,J)
      CALL PRNT(4,0,N,N)
      IF(MX(4) - 2)  95,91,91
C     PUNCH MEANS, STANDARD DEV. AND NO. OF CASES
   91 READ(ICR,108)
      DO 92 I=1,N
   92 WRITE(ICP,107) IPROB,KON,KON1,I,SUMY(I),SD(I)
      KON = 21
      WRITE(ICP,107) IPROB,KON,KON1,KON1,CASES
      GO TO (15,95),ISW
   95 IF(KNN)150,150,151
  150 CALL LINK(REGR2)
  151 CALL LINK(FCTR1)
      END
// DUP
*STORE      WS  UA  COREL
```

```
// FOR  MATRIX PRINT/PUNCH ROUTINE
*ONE WORD INTEGERS
C      MATRIX PRINT/PUNCH ROUTINE
       SUBROUTINE PRNT(MID,KODE,NR,NC)
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,KX(5),
      1MX(20),NX(15),FLVB(5),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,30)
   101 FORMAT(5XA4,4X8E13.5)
   102 FORMAT(5XA4,4X8E13.5)
   103 FORMAT(   10X18A4,5X3HJOBI7, 5X,4HPAGE I6)
   104 FORMAT(I4,3I2,5E14.7)
   105 FORMAT(/103H READY THE PUNCH WITH BLANK CARDS AND PRESS START ON T
      1HE PUNCH AND CONSOLE.  TURN CONSOLE SWITCH 15 ON.)
   106 FORMAT(1H )
   201 FORMAT(///2X,8HVARIABLE,07X,8(A4,9X)///)
   202 FORMAT(3X8HVARIABLE7X8(I4,8X)//)
   321 FORMAT(/  46X,28HMATRIX OF RAW CROSS-PRODUCTS )
   322 FORMAT(/  43X,33HMATRIX OF RESIDUAL CROSS-PRODUCTS )
   323 FORMAT(/45X28HVARIANCE - COVARIANCE MATRIX   )
   324 FORMAT(/  42X,34HMATRIX OF CORRELATION COEFFICIENTS )
   325 FORMAT(/45X32HMATRIX OF CHARACTERISTIC VECTORS/)
   326 FORMAT(/41X36HNORMALIZED UNROTATED FACTOR LOADINGS)
       KNME=1
       IF(MX(MID)-1)1000,1,100
     1 I = 1
       II = 8
     9 IF(NC-II) 10,11,11
    10 II = NC
    11 NPAGE = NPAGE + 1
       CALL FMAT(IPR,ITW)
    16 IF(IPR) 12,12,13
 12    WRITE(ITW,103)TITLE,IPROB,NPAGE
 13    GO TO (21,22,23,24,25,26),MID
    21 WRITE(ITW,321)
       GO TO 30
    22 WRITE(ITW,322)
       GO TO 30
    23 WRITE(ITW,323)
       GO TO 30
    24 WRITE(ITW,324)
       GO TO 30
    25 WRITE(ITW,325)
       GO TO 30
    26 WRITE(ITW,326)
    30 IF(MID-5)40,41,40
    40 IF(MID-6)42,41,42
    41 KNME=0
       WRITE(ITW,202)(J,J=I,II)
       GO TO 43
    42 WRITE(ITW,201)(VNAME(J),J=I,II)
    43 DO 35 K=1,NR
       IF(KODE) 34,33,34
    33 IF(KNME)44,45,44
    44 KNME=VNAME(K)
    45 WRITE(ITW,101) VNAME(K),(R(K,J),J=I,II)
       GO TO 35
    34 WRITE(ITW,102) VNAME(K),(R(K,J),J=I,II)
    35 CONTINUE
       IF(NC-II) 36,1000,36
    36 I = I+8
       II = II + 8
       GO TO 9
C      PUNCH ROUTINE
   100 I = 1
       II = 5
       READ(ICR,106)
       CALL DATSW(15,JIG)
       IF(JIG-2)151,3,3
     3 WRITE(ITW,105)
       PAUSE
   151 IF(NC-II) 152,153,153
   152 II = NC
   153 DO 154 K = 1,NR
   154 WRITE(ICP,104)IPROB,MID,I ,K,(R(K,J),J=I,II)
       IF(NC-II)  155,156,155
   155 I = I + 5
       II = II + 5
       GO TO 151
   156 IF(MX(MID)-2) 1000,1,1000
  1000 RETURN
       END
// DUP
*STORE      WS  UA  PRNT
```

```
// FOR SUBROUTINE TO COMPUTE COEFFICIENTS OF POLYNOMIAL
*     ONE WORD INTEGERS
C     SUBROUTINE TO COMPUTE COEFFICIENTS OF POLYNOMIAL
      SUBROUTINE PCOEF
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON A(51),TEMP1(51),TEMP2(51),TEMP3(51)
  101 FORMAT(   10X18A4,5X3HJOBI7, 5X,4HPAGE,I6)
  102 FORMAT(20XI5,E20.7)
  103 FORMAT(//20X,33HCOEFFICIENTS OF FITTED POLYNOMIAL/I1)
C     PROGRAM INITIALIZATION
      B = 0.0
      KKD = NF+1
      DO 1 NN = 1,KKD
      A(NN) = C(NN)
      TEMP1(NN) = 0.0
      TEMP2(NN) = 0.0
    1 TEMP3(NN) = 0.0
C      BEGIN COMPUTATION
      DO 6 II=2,KKD
      TEMP2(II) = 1.0
      DO 3 NN=2,II
      TEMP3(NN) = TEMP2(NN-1)-TEMP2(NN)*ALPHA(II-1)-B*TEMP1(NN)
C     COMPUTATION OF A COEFFIEICENT
    3  A(NN-1) =A(NN-1)+C(II)*TEMP3(NN)
      IF(II-KKD) 4,8,8
C     RESETTING THE VECTORS FOR THE NEXT COEFFICIENT
    4 DO 5 NN=1,II
      TEMP1(NN) = TEMP2(NN)
    5 TEMP2(NN) = TEMP3(NN)
    6 B = BETA(II-1)
C     OUTPUT POLYNOMIAL COEFFICIENTS
    8 NPAGE = NPAGE + 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 81,81,82
   81 WRITE(ITW,101) TITLE,IPROB,NPAGE
   82 WRITE(ITW,103)
      DO 9 J = 1,KKD
      L = J-1
    9 WRITE(ITW,102) L,A(J)
   20 RETURN
      END
// DUP
*STORE      WS  UA  PCOEF
```

```
// FOR SUBROUTINE TO INPUT DATA FOR ORTHOGONAL POLYNOMIALS
*     ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME POLY
C     SUBROUTINE TO INPUT DATA FOR ORTHOGONAL POLYNOMIALS
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON MF1(50)
      DEFINE FILE 606(150,8,U,IT1)
  101 FORMAT(6I2)
  102 FORMAT(I4,4X,18A4)
  103 FORMAT(8I2,F10.4,3I2)
  104 FORMAT(///43H THE X VALUES HAVE BEEN TRANSFORMED TO X'=(,E14.7,7H)
     1*X + (,E14.7,2H).)
  105 FORMAT(    10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6//11X,28HMAXIMUM DEGRE
     1E OF POLYNOMIAL,5X,I2/11X,10HINPUT TYPE,23X,I2/11X,23HPOLYNOMIAL C
     3OEFFICIENTS,10X,I2/11X,19HCOMPUTE DERIVATIVES,14X,I2/11X,19HORDER
     4OF DERIVATIVE,14X,I2/11X,16HPREDICTED VALUES,17X,I2/11X,22HPUNCH S
     5OLUTION VECTORS,11X,I2/11X20HSECONDARY INPUT TYPE,13X,I2/11X,18HVA
     6RIANCE CRITERION,2X,F15.9/11X,21HTRANSFORMATION SWITCH,12X,I2)
  106 FORMAT(/11X,7HSCALING26X,I2/11X,24HIGNORE POLYNOMIAL OUTPUT9X,I2)
  107  FORMAT(///,' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED IN COLUMN
     1',I3,' OF THE ABOVE FORMAT CARD.'/' CHANGE CARD AND RERUN JOB.')
  108  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED AT APPROXIMA
     1TELY COLUMN',I3' OF THE ABOVE DATA CARD.'/' CHANGE OR REMOVE CARD
     2AND PRESS START TO CONTINUE')
  109 FORMAT(8XI2,3E14.7)
  110  FORMAT(//' INVALID INPUT OPTION-JOB TERMINATED ')
  111 FORMAT(2E14.7,I2)
  112 FORMAT(//27H X = X' (NO TRANSFORMATION))
C     SUBROUTINE TO READ AND PRINT PARAMETER CARDS (POLYNOMIAL)
      KWS=1
      NPAGE = 0
      READ(2,101) ICR,ICP,IPR,ITW,IT1,IT2
      IF(IPR)701,702,701
  702 ITW=3
      GO TO 703
  701 ITW=1
  703 READ(ICR,102) IPROB,TITLE
      READ(ICR,103) N,INMD,ICOF,IDER,NDER,ISCR,IALP,INMD2,EPS,KX(3)
     1,KX(4),KX(5)
      CALL FMAT(IPR,ITW)
      WRITE(ITW,105) TITLE,IPROB,NPAGE,N,INMD,ICOF,IDER,NDER,ISCR,IALP,
     1INMD2,EPS,KX(3)
      WRITE(ITW,106)KX(4),KX(5)
      IF(INMD - 2) 1,5,1001
 1001 IF(INMD-3)1002,1,1002
 1002 WRITE(ITW,110)
      CALL EXIT
    1 CALL FMTRD(MF1,IRR)
      CALL PRNTB
      IF(IRR) 2,5 ,2
    2 WRITE(ITW,107) IRR
      CALL EXIT
```

```
    5 NF = N
   10 IF(INMD-2) 11,11,30
   11 DO 14 I=1,150
      IF(INMD-1) 16,16,20
C     READ DATA FROM CARD READER
   16 CALL DATRD(MF1,IRR,ID(I),1,IDR,1,        X(I),-1,Y(I),-1,0,0)
      WRITE(606'I)ID(I),IDR,     X(I),Y(I)
      IF(IRR) 17,18,17
   17 CALL PRNTB
      WRITE(ITW,108) IRR
      PAUSE 10
      GO TO 16
   20 READ(606'I)ID(I),IDR,     X(I),Y(I)
   18 IF(ID(I)) 15,15,19
   19 IF(IDR) 13,13,12
   12 ID(I) = -ID(I)
   13 IF(KX(3)) 143, 14,143
  143 CALL TRAN
   14 CONTINUE
   15 NCASE = I-1
      IF(KX(4))35,200,35
  200 WRITE(ITW,112)
      GO TO 100
   35 IF(KWS)356,355,356
  356 XN=1.0E-30
      X1=1.0E+30
      DO 39 I=1,NCASE
      IF(X(I)-XN)37,37,36
   36 XN=X(I)
   37 IF(X(I)-X1)38,39,39
   38 X1=X(I)
   39 CONTINUE
      XB=XN-X1
      X14=4./XB
      XB=-(X1+X1+XN+XN)/XB
  355 DO 40 I=1,NCASE
   40 X(I)=X14*X(I)+XB
      WRITE(ITW,104)X14,XB
      GO TO 100
C     READ ALPHA,BETA,C FROM CARD READER
   30 NP1 = N&1
      KWS=0
      READ(ICR,111)X14,XB,KX(4)
      DO 31   I=1,NP1
      READ(ICR,109) K,T1,T2,T3
      ALPHA(K) = T1
      BETA(K) = T2
   31 C(K) = T3
      INMD = INMD2
      GO TO 11
  100 CALL LINK(POL2)
      END
// DUP
*STORE      WS  UA  POLY
// FOR   SECONDARY MAIN FOR ORTHOGONAL POLYNOMIALS
```

```
*ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME POL2
C     SECONDARY MAIN FOR ORTHOGONAL POLYNOMIALS
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON YA(150),POLY(150),POLYO(150),POL(150,4),SSR(51)
      DEFINE FILE 606(150,8,U,IT1)
  100 FORMAT(/2X13HJOB COMPLETED)
      IF(INMD2) 5,5,6
    5 CALL POLSQ
    6 IF(ICOF) 8,8,7
    7 CALL PCOEF
    8 IF(IDER) 10,10,9
    9 CALL PDER
   10 IF(ISCR) 13,13,11
   11 CALL PFIT
   13 WRITE(ITW,100)
      CALL EXIT
      END
// DUP
*STORE      WS  UA  POL2
```

```
// FOR COMPUTATION OF ORTHOGONAL POLYNOMIALS
*     ONE WORD INTEGERS
C     COMPUTATION OF ORTHOGONAL POLYNOMIALS
      SUBROUTINE POLSQ
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON YA(150),POLY(150),POLYO(150),POL(150,4),SSR(51)
  101 FORMAT(///68H MAX DEGREE OF POLYNOMIAL REACHED.  VARIANCE CRITERIO
     1N NOT SATISFIED    )
  102 FORMAT( /40X,22HORTHOGONAL POLYNOMIALS/ 1X14HIDENTIFICATION7X2HX'9
     1X1HY11X2HY*10X4HY-Y*,I12,3I13,/)
  103 FORMAT(   10X18A4,5X3HJOBI7,5X,4HPAGE,I6)
  104 FORMAT(1H )
  105 FORMAT(/103H READY THE PUNCH WITH BLANK CARDS AND PRESS START ON T
     1HE PUNCH AND CONSOLE.  TURN CONSOLE SWITCH 15 ON.)
  106 FORMAT(1XI3,I7,2X4E13.5,4F13.6)
  107 FORMAT(    60X5HALPHA4E13.5)
  108 FORMAT(    61X4HBETA4E13.5)
  109 FORMAT(    64X1HC4E13.5)
  110 FORMAT(I4,3I2,3E14.7)
  111 FORMAT(2E14.7,I2)
  112 FORMAT(1XI3,I7,2X8E13.5)
  113 FORMAT(//22X20HANALYSIS OF VARIANCE//3X16HVARIATION SOURCE11X4HD.F
     1.6X28HSUM OF SQUARES   MEAN SQUARE/)
  114 FORMAT(3X6HDEGREE,I3,10H COMPONENT,I11,7X,2E14.5)
  115 FORMAT(3X16HRESIDUALS(DEGREE,I3,7H REGR.),I4,7X,2E14.5/)
C     INITIALIZATION
      MN=41
      II=1
      N1=1
      IT = 1
      ISW = 1
      RO = NCASE
      FN = RO
      SY=0.0
      DEL1 = 0.0
      DO 10 I =1,NCASE
      DEL1 = DEL1&Y(I)*Y(I)
      SY=SY+Y(I)
      POLYO(I) = 0.0
      YA(I) = 0.0
   10 POLY(I) = 1.0
      YBAR=SY/RO
      SY=DEL1-SY*YBAR
      B=0.
C  BEGIN COMPUTATION
   11 S = 0.
      SSR(II)=0.0
      SSR(II+1)=0.0
      DO 12 I=1,NCASE
   12 S= S&Y(I)*POLY(I)
C     COMPUTATION OF A COEFFICIENT IN THE POLYNOMIAL EQUATION.
       C(II) = S/RO
C     COMPUTE PREDICTED VALUES
      DO 13 I=1,NCASE
   13 YA(I) = YA(I)&C(II)*POLY(I)
C     DETERMINE IF VARIANCE CRITERION IS SATISFIED
      DEL2 = DEL1-S*S/RO
      VAR2 = DEL2/(FN-II-1.0)
      IF(II-NCASE+1)610,610,611
  610 IF(II-1) 17,17,14
   14 IF(ABS(VAR2-VAR1)-EPS) 15,16,16
   15 II=II-1
      NF=II-1
      IT=IT-1
      ISW=2
      GO TO 45
   16 IF(II-N) 17,17,61
C      COMPUTATION OF ALPHA FOR THE POLYNOMIAL EQUATION
   17 SUMXQ = 0.
      DO 18 I=1,NCASE
   18 SUMXQ = SUMXQ & X(I)*POLY(I)*POLY(I)
       ALPHA(II) = SUMXQ/RO
C      COMPUTATION OF A NEW POLYNOMIAL
      DO 19 I=1,NCASE
      POL(I,IT) = POLY(I)
      POLY(I) = (X(I)-ALPHA(II))*POLY(I)-B*POLYO(I)
   19 POLYO(I) = POL(I,IT)
      DO 191 I=1,NCASE
      SSR(II)=SSR(II)+((Y(I)-YBAR)*POLY(I))
  191 SSR(II+1)=SSR(II+1)+POLY(I)**2
      SSR(II)=SSR(II)**2/SSR(II+1)
C      COMPUTATION OF BETA FOR THE POLYNOMIAL EQUATION
       R = 0.0
      DO 20 I=1,NCASE
   20 R = R & POLY(I)*POLY(I)
      BETA(II) = R/RO
      RO = R
      B = BETA(II)
      GO TO (21,45),ISW
C     OUTPUT SECTION OF ORTHOGONAL POLYNOMIALS
   21 IF(IT-4) 60,45,45
   45 IX = II-1
      IF(KX(5))507,506,507
  506 IL = II-IT
      NPAGE = NPAGE & 1
      DO 52 I=1,NCASE
      IF(I-MN)503,48,503
   48 NPAGE=NPAGE+1
      MN=MN+40
      GO TO 47
  503 IF(I-1)502,47,502
   47 CALL FMAT(IPR,ITW)
      IF(IPR) 461,461,478
  461 WRITE(ITW,103) TITLE,IPROB,NPAGE
  478 WRITE(ITW,102)(J,J=IL,IX)
  502 DIF = Y(I) - YA(I)
      IF(KX(4))528,529,528
  529 WRITE(ITW,112) I,ID(I),X(I),Y(I),YA(I),DIF,(POL(I,J),J=1,IT)
      GO TO 52
  528 WRITE(ITW,106) I,ID(I),X(I),Y(I),YA(I),DIF,(POL(I,J),J=1,IT)
   52 CONTINUE
      IL = IL&1
      WRITE(ITW,107) (ALPHA(I),I=IL,II)
      WRITE(ITW,108) (BETA(I),I=IL,II)
      WRITE(ITW,109) (C(I),I=IL,II)
  507 IT = 0
      GO TO (60,100),ISW
C     CONTINUE THE NEXT ORDER POLYNOMIAL
   60 DEL1 = DEL2
      VAR1 = VAR2
      II = II & 1
      IT = IT & 1
      GO TO 11
```

```
   61 WRITE(ITW,101)
      NF = N
   65 ISW = 2
      GO TO 17
  611 WRITE(ITW,101)
      NF=NCASE-1
      GO TO 65
C     TEST FOR PUNCHING OF ALPHA,BETA,C
  100 IF(IALP) 140,140,125
  125 NFP1 = NF&1
      KON = 24
      KON1 = 1
      READ(ICR,104)
      CALL DATSW(15,JIG)
      IF(JIG-2)151,3,3
    3 WRITE(ITW,105)
      PAUSE
  151 WRITE(ICP,111)X14,XB,KX(4)
      DO 126 I=1,NFP1
  126 WRITE(ICP,110) IPROB,KON,KON1,I,ALPHA(I),BETA(I),C(I)
  140 WRITE(ITW,113)
      DO 192 I=1,NF
      WRITE(ITW,114)I,N1,SSR(I),SSR(I)
      NDF=NCASE-1-I
      ADF=NDF
      SY=SY-SSR(I)
      AMY=SY/ADF
  192 WRITE(ITW,115)I,NDF,SY,AMY
      RETURN
      END
// DUP
*STORE      WS  UA  POLSQ
// FOR SUBROUTINE TO COMPUTE POLYNOMIAL PREDICTED VALUES
*     ONE WORD INTEGERS
C     SUBROUTINE TO COMPUTE POLYNOMIAL PREDICTED VALUES
      SUBROUTINE PFIT
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON YA(150),POLY(150),POLYO(150)
  101  FORMAT(   10X18A4,5X3HJOBI7, 5X,4HPAGE I6/)
  102 FORMAT(10XI3,I7,2X4E13.5)
  103 FORMAT(//8X,14HIDENTIFICATION,9X2HX'12X1HY12X2HY*9X4HY-Y*//)
C     INITIALIZATION
      KAJP1 = NF&1
      B=0.0
      DO 1 I=1,NCASE
      YA(I)=0.0
      POLY(I)=1.0
    1 POLYO(I)=0.0
      DO 6 II =1,KAJP1
C     COMPUTE PREDICTED VALUES
      DO 3 I=1,NCASE
    3 YA(I)=YA(I)&C(II)*POLY(I)
      IF(II-KAJP1)4,8,8
C     COMPUTE NEXT ORDER POLYNOMIAL
    4 DO 5 I=1,NCASE
      TEMP=POLY(I)
      POLY(I)=(X (I)-ALPHA(II))*POLY(I)-B*POLYO(I)
    5 POLYO(I)=TEMP
    6 B=BETA(II)
C     OUTPUT PREDICTED VALUES
    8 LINES = 50
      IF(IPR)7,9,7
    7 WRITE(ITW,103)
    9 DO 12 I=1,NCASE
      IF(LINES-50) 11,10,10
   10 NPAGE = NPAGE & 1
      LINES = 0
      CALL FMAT(IPR,ITW)
      IF(IPR) 13,13,11
 13   WRITE(ITW,101) TITLE,IPROB,NPAGE
      WRITE(ITW,103)
   11 DIF = Y(I) - YA(I)
      LINES = LINES & 1
   12 WRITE(ITW,102) I,ID(I),X(I),Y(I),YA(I),DIF
      RETURN
      END
// DUP
*STORE      WS  UA  PFIT
```

```
// FOR SUBROUTINE TO COMPUTE POLYNOMIAL DERIVATIVES
*     ONE WORD INTEGERS
C     SUBROUTINE TO COMPUTE POLYNOMIAL DERIVATIVES
      SUBROUTINE PDER
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1NCASE,ICOF,IDER,NDER,IALP,INMD2,KX(5),EPS,FLVB(4),XB,X14
      COMMON TITLE(18),ID(150),X(150),Y(150),C(51),ALPHA(51),BETA(51)
      COMMON DPOLY(51),DERIV(51),DOPOL(51)
  101 FORMAT(   10X18A4,5X3HJOBI7, 5X,4HPAGE,I6)
 102  FORMAT(/  5X,I9,7X,2F15.5,I12,7X,F15.5)
  103 FORMAT(61XI2,7XF15.5)
  104 FORMAT(//5X,14HIDENTIFICATION,14X2HX'14X2HY*8X12HDERIV. ORDER7X12H
     1DERIV. VALUE/)
      IF(NDER)17,16,17
   17 LINES = 50
      IF(NF-NDER)30,31,31
   30 NDP1=NF+1
      GO TO 32
   31 NDP1 = NDER + 1
   32 KBJP1 = NF + 1
      DO 25 IL1=1,NCASE
      IF(ID(IL1)) 1,25,25
    1 XB = X(IL1)
      DO 2 II = 1,KBJP1
      DPOLY(II) = 0.0
    2 DOPOL(II) = 0.0
      DPOLY(KBJP1 + 1) = 0.0
      DPOL = 1.0
      NN = 1
      GO TO 4
C     BEGIN COMPUTATION.
    3 DPOL = DPOLY(NN)
    4 DPOLO = 0.0
      DERIV(NN)=0.0
      B = 0.0
      II = 1
C     COMPUTATION OF FITTED VALUE AND DERIVATIVE.
    5 DERIV(NN) = DERIV(NN) + C(II) * DPOL
      IF(II-KBJP1) 6,7,7
C     COMPUTATION OF A NEW POLYNOMIAL DERIVATIVE.
    6 TEMP = DPOL
      DPOL = (XB - ALPHA(II))*DPOL +       (NN - 1) * DOPOL(II) -B*DPOLO
      DPOLO = TEMP
      DOPOL(II)=DPOLO
      B = BETA(II)
      II = II + 1
      GO TO 5
C     COMPUTATION OF THE NEXT DERIVATIVE.
    7 IF(NN - NDP1) 8,9,8
    8 NN = NN + 1
      GO TO 3
C     OUTPUT DERIVATIVE
    9 IF(LINES-50) 12,11,11
   11 NPAGE=NPAGE + 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 111,111,112
  111 WRITE(ITW,101) TITLE,IPROB,NPAGE
  112 WRITE(ITW,104)
      LINES = 0
   12 L=1
      LINES = LINES + 2
      WRITE(ITW,102) ID(IL1),XB,DERIV(1),L,DERIV(2)
      IF(NDP1-3) 25,13,13
   13 DO 14 J=3,NDP1
      L = J-1
      LINES = LINES + 1
   14 WRITE(ITW,103) L,DERIV(J)
   25 CONTINUE
   16 RETURN
      END
// DUP
*STORE      WS  UA  PDER
```

```
// FOR INPUT DATA SUBROUTINE
*ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME REGR
C      INPUT DATA SUBROUTINE
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,IPRED,
      1ISTEP,ICNST,IREAR,KX(1),MX(20),NCD(3),        ISEQ,NCASE,NX(10),
      2 EFOUT,EFIN,TOL,FLVB(2),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
       COMMON HIGH(30),HLOW(30),MF(50,3)
       DEFINE FILE 606(500,65,U,IT1)
  101 FORMAT(6I2)
  102 FORMAT(I4,4X,18A4)
  103 FORMAT(15I2,      2F4.3,F6.5)
  104 FORMAT(20A4)
  105 FORMAT(    10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6//11X,19HNUMBER OF VAR
      1IABLES,16X,I2/11X,10HINPUT TYPE , 25X,I2/11X,14HSEQUENCE CHECK 21X
      2I2/11X,19HVARIABLES ON CARD 1 16X,I2/11X19HVARIABLES ON CARD 2 16X
      3I2/11X,19HVARIABLES ON CARD 3 16X,I2/11X,21HTRANSFORMATION SWITCH,
      414X,I2/11X,25HOUTPUT RAW CROSS PRODUCTS 10X,I2/11X,30HOUTPUT RESID
      5UAL CROSS PRODUCTS 5X,I2)
  106 FORMAT(11X,22HPRINT PREDICTED VALUES,13X,I2/11X11HPRINT STEPS,24X,
      1I2/11X,14HPOOLING OPTION,21X,I2      /11X,18HDEPENDENT VARIABLE 17
      2X,I2/11X,27HF-LEVEL TO REMOVE VARIABLES,F10.3/11X,26HF-LEVEL TO EN
      3TER VARIABLES,F11.3/11X,15HTOLERANCE VALUE,11X,F11.5/11X,28HOUTPUT
      4 VARIANCE - COVARIANCE 7X,I2/11X,18HOUTPUT CORRELATION 17X,I2)
 107  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED IN COLUMN',
      1I3,' OF THE ABOVE FORMAT CARD.'/' CHANGE CARD AND RERUN JOB.')
 108  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED AT APPROXIMA
      1TELY COLUMN',I3,' OF THE ABOVE DATA CARD.'/' CHANGE OR REMOVE CARD
      2 AND PRESS START TO CONTINUE.')
  109 FORMAT(//5X4HCARDI10,4H NO.I4,1X 30HIS OUT OF SEQUENCE. RERUN JOB.
      1  )
 110  FORMAT(//' INVALID INPUT OPTION-JOB TERMINATED ')
       KNN=0
       NPAGE = 0
       READ(2,101) ICR,ICP,IPR,ITW,IT1,IT2
       IF(IPR)701,702,701
  702 ITW=3
       GO TO 703
  701 ITW=1
  703 READ(ICR,102) IPROB,TITLE
       READ(ICR,103) N,INMD,ISEQ,(NCD(I),I=1,3),MX(20),
      1(MX(I),I=1,4),IPRED,ISTEP,ICNST,IREAR,EFOUT,EFIN,TOL
       CALL FMAT(IPR,ITW)
       WRITE(ITW,105) TITLE,IPROB,NPAGE,N,INMD,ISEQ,(NCD(I),I=1,3),MX(20)
      1,(MX(I),I=1,2)
       WRITE(ITW,106) IPRED,ISTEP,ICNST,IREAR,EFOUT,EFIN,TOL,
      1(MX(I),I=3,4)
       READ(ICR,104) (VNAME(I),I=1,N)
       IF(INMD-1) 1002,1,1004
 1004 IF(INMD-4) 5,1002,1002
    1 DO 4 I=1,3
       CALL FMTRD(MF(1,I),IRR)
       CALL PRNTB
       IF(IRR) 2,3,2
    2 WRITE(ITW,107) IRR
       CALL EXIT
 1002 WRITE(ITW,110)
       CALL EXIT
    3 IF(NCD(I+1)) 5,5,4
    4 CONTINUE
C      SUBROUTINE TO READ SOURCE DATA
C      INITIALIZATION
    5 DO 8 I=1,N
       HIGH(I) = 0.
       SD(I)=0.
       HLOW(I) = 1.0E+36
       SUMY(I) = 0
       DO 8 J=1,N
    8 R(I,J) = 0.0
       KOUNT = 0
       CASES = 0.
       NCASE = 0
    9 IT1 = 1
  10   GO TO (11,41,51),INMD
C      CARD READER INPUT
   11 IST = 1
       I=1
       IF(NCD(1)) 12,12,13
   12 NCD(1) = N
   13 CALL DATRD(MF(1,1),IRR,ID,1,NC,1,     X,-NCD(1),0,0)
       IF(IRR) 14,15,14
   14 CALL PRNTB
       WRITE(ITW,108) IRR
       PAUSE 10
       GO TO (13,18,18),I
   15 IF(ID) 100,16,16
   16 DO 22 I=2,3
       IF(NCD(I)) 23,23,17
   17 IST = NCD(I-1) + IST
   18 CALL DATRD(MF(1,I),IRR,ID1,1,NC1,1,X(IST),-NCD(I),0,0)
       IF(IRR) 14,19,14
   19 IF(ISEQ) 22,22,20
   20 IF(ID-ID1) 60,21,60
   21 IF(NC1-NC) 60,60,6
    6 ID = ID1
       NC = NC1
   22 CONTINUE
       GO TO 23
   60 WRITE(ITW,109) ID1,NC1
       CALL EXIT
   23 IF(MX(20)) 230,231,230
  230 CALL TRAN
  231 IF(INMD-1) 1002,27,30
   27 WRITE(606'IT1) ID ,  (X(I),I=1,N)
C      COMPUTE CROSS PRODUCT MATRIX
   30 CASES = CASES + 1.
       NCASE = NCASE + 1
       DO 35 I = 1,N
```

```
      SUMY(I) = SUMY(I) + X(I)
      DO 35 J=I,N
      R(I,J) = R(I,J) + X(I)*X(J)
   35 R(J,I) = R(I,J)
C     DETERMINE HIGH AND LOW VALUES
      DO 39 I=1,N
      IF(X(I) - HIGH(I)) 37,37,36
   36 HIGH(I) = X(I)
   37 IF(X(I) - HLOW(I))38,39,39
   38 HLOW(I) = X(I)
   39 CONTINUE
      GO TO 10
   41 READ(606'IT1) ID    ,(X(I),I=1,N)
      IF(ID) 100,100,23
C     READ A MATRIX FROM CARDS
 51   CALL MXRAD
 100  IF(INMD-1) 1002,150,151
 150  WRITE(606'IT1) ID   ,(X(I),I=1,N)
 151  IT1 = 1
  200 IF(NCASE)571,571,572
  572 CALL LINK(COREL)
  571 CALL LINK(REGR2)
      END
// DUP
*STORE      WS  UA  REGR
```

```
// FOR CALLING PROGRAM FOR CORRELATION AND REGRESSION
*IOCS(CARD,1132PRINTER,DISK)
*ONE WORD INTEGERS
*NAME REGR2
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,IPRED,
     1ISTEP,ICNST,IREAR,KX(1),MX(20),NCD1,NCD2,NCD3,ISEQ,NCASE,NX(10),
     2 EFOUT,EFIN,TOL,FLVB(2),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
      COMMON HIGH(30),HLOW(30)
      DEFINE FILE 606(500,65,U,IT1)
  100 FORMAT(//2X13HJOB COMPLETED)
      IF(ISTEP)4,4,2
    2 IF(IREAR) 4,4,3
    3 CALL REGRE
    4 WRITE(ITW,100)
      CALL EXIT
      END
// DUP
*STORE      WS  UA  REGR2
```

```
// FOR SUBROUTINE FOR STEPWISE REGRESSION
*ONE WORD INTEGERS
C     SUBROUTINE FOR STEPWISE REGRESSION
      SUBROUTINE REGRE
      DIMENSION INDEX(30)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,IPRED,
     1ISTEP,ICNST,IREAR,KX(1),MX(20),NX(15),EFOUT,EFIN,TOL,FLVB(2),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,30)
      COMMON CORRY(30),COEN(30)
  101 FORMAT(   10X18A4,5X3HJOBI7, 5X,4HPAGE I6)
  102 FORMAT(//5X,19HREGRESSION ANALYSIS //5X,18HDEPENDENT VARIABLE ,16X
     1,A4/5X,27HRESIDUAL STANDARD DEVIATIONF11.4/5X,26HSTANDARD ERROR OF
     2 THE MEAN F12.4/5X,10HMULTIPLE R 18X,F10.4/5X,13HMULTIPLE RSQR 15X
     3,F10.4)
  103 FORMAT(///4X,16HVARIABLE REMOVED 18X,A4//)
  104 FORMAT(///4X,16HVARIABLE ENTERED 18X,A4//)
  105 FORMAT(//3X,8HVARIABLE11X,8HB - COEF 4X,14HSTD ERROR OF B 9X,
     19HPARTIAL-R,8X,9HBETA-COEF 4X17HSTD ERROR OF BETA/)
  106 FORMAT(///30X26HANALYSIS OF VARIANCE TABLE)
  107 FORMAT(5X,A4,6X,2F15.4,13X,F7.4,2X,F15.4,3X,F15.4)
  108 FORMAT(/// 2X,8HCONSTANT 4X,F15.4)
  109 FORMAT(/// 33X,16HPREDICTED VALUES//11X,4HCASE,12X,6HACTUAL,11X,
     19HPREDICTED,16X,8HRESIDUAL///)
  110 FORMAT(10X,I5,3(5X,E15.4))
  111 FORMAT(/15X6HSOURCE13X4HD.F.05X14HSUM OF SQUARES3X11HMEAN SQUARE08
     1X1HF)
  112 FORMAT(/// 42HMEAN SQUARE NON-POSITIVE. JOB TERMINATED. )
  113 FORMAT (///41H NO MORE DEGREES FREEDOM. JOB TERMINATED.  )
  114 FORMAT( 15X10HREGRESSION5X,I6,5X,E14.5,E16.5,E15.5)
  115 FORMAT( /// 62HNO MORE VARIABLES SATISFY VARIANCE CRITERION. JOB T
     1ERMINATED. )
  116 FORMAT(15X5HERROR10X,I6,5X,E14.5,E16.5)
  117 FORMAT(/15X4HMEAN11X,I6,5X,E14.5,E16.5)
      PLACE DEPENDENT VARIABLE AT END OF MATRIX
      IK1=1
      IKT=0
      IF(IREAR) 6,6,2
    2 DO 3 I = 1,N
      T = R(I,N)
      R(I,N) = R(I,IREAR)
    3 R(I,IREAR) = T
      DO 4 I=1,N
      T = R(N,I)
      R(N,I) = R(IREAR,I)
    4 R(IREAR,I) = T
      T = SUMY(N)
      SUMY(N) = SUMY(IREAR)
      SUMY(IREAR) = T
      T = SD(N)
      SD(N) = SD(IREAR)
      SD(IREAR) = T
      T = VNAME(N)
      VNAME(N) = VNAME(IREAR)
      VNAME(IREAR) = T
      INITIALIZE COMPUTATIONAL PARAMETERS
    6 NOVMI = N-1
      DO 7 I = 1,NOVMI
    7 CORRY(I) = R(I,N)
      DEFR = CASES - 1.
      SSM=CASES*SUMY(N)**2
      SSY=DEFR      *SD(N)**2
      ANODA=SQRT(DEFR/CASES)
      NOENT = 0
      NOMIN = 0
      NOMAX = 0
C     START OF MAIN ITERATION FOR A VARIABLE
C     COMPUTE STANDARD ERROR OF MEAN AND ESTIMATE
   11 SMEAN = SD(N)*SQRT(R(N,N)/DEFR)*ANODA
      SEST = SMEAN * SQRT(CASES)
C     COMPUTE MULTIPLE R AND MULTIPLE R**2
      R2M1 = 1.0-R(N,N)
      IF(R2M1) 31,31,30
   30 RMLT = SQRT(R2M1)
      GO TO 32
   31 RMLT = 0.0
   32 RSQ = RMLT**2
C     INITIALIZE VARIABLE ENTRY PARAMETERS
      VMIN = 1.0E20
      VMAX = 0.0
      VAR = 0.0
      NOIN = 0
C     DETERMINE ENTRY VARIABLES AND COMPUTE COEFFICIENTS
C     AND THEIR STANDARD ERRORS
   35 DO 56 I=1,NOVMI
   41 IF(R(I,I) - TOL) 56,42,42
   42 VAR = R(I,N)*R(N,I)/R(I,I)
      IF(VAR) 44,56,53
   44 NOIN = NOIN & 1
      INDEX(NOIN) = I
      IF (ABS(VAR) - ABS(VMIN)) 50, 50, 56
   50 VMIN = VAR
      NOMIN = I
   52 GO TO 56
   53 IF(VAR-VMAX) 56,56,54
   54 VMAX = VAR
      NOMAX = I
   56 CONTINUE
C     IF NO VARIABLES ENTERED  GO TO NEXT ITERATION
      IF(NOIN) 82,82,66
C     OUTPUT REGRESSION EQUATION FOR THIS STEP
   66 IF(ISTEP)400,401,400
  400 IF(ISTEP-NOIN) 68,68,78
   68 NPAGE = NPAGE & 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 681,681,682
  681 WRITE(ITW,101) TITLE,IPROB,NPAGE
  682 WRITE (ITW,102) VNAME(N),SEST,SMEAN,RMLT,RSQ
      IF(NOENT) 69,69,71
   69 WRITE(ITW,103) VNAME(K)
      GO TO 72
```

```
   71 WRITE(ITW,104) VNAME(K)
   72 WRITE(ITW,105)
   63 CNST = SUMY(N)
   65 DO 76 I=1,NOIN
      IL = INDEX(I)
      PARTL = 0.0
C     COMPUTE COEFFICIENTS AND THEIR STANDARD ERRORS
      BETA = R(IL,N)
      COEN(I) = BETA*SD(N)/SD(IL)
      BER = SQRT(R(N,N)*R(IL,IL)/DEFR)
      SIGM = BER*SD(N)/SD(IL)
C     COMPUTE PARTIAL CORRELATION COEFFICIENTS
      DO 58 J = 1,NOIN
      JL = INDEX(J)
   58 PARTL=PARTL+(R(JL,N)-R(IL,N)  *R(JL,IL)/R(IL,IL))*CORRY(JL)
      PARTL     = SIGN(SQRT(1.0-R(N,N)/(1.0-PARTL    )),COEN(I))
      WRITE(ITW,107) VNAME(IL),COEN(I),SIGM,PARTL,BETA,BER
C     COMPUTE CONSTANT TERM
   76 CNST = CNST-(COEN(I)*SUMY(IL))
      WRITE(ITW,108) CNST
      WRITE(ITW,106)
      WRITE(ITW,111)
      IDF=CASES-DEFR-1.
      IEDF=DEFR
      AMSE=SEST**2
      SSE=AMSE*DEFR
      SSR=SSY-SSE
      AMSR=SSR/(CASES-DEFR-1.)
      AF=AMSR/AMSE
      WRITE(ITW,117)IK1,SSM,SSM
      WRITE(ITW,114)IDF,SSR,AMSR,AF
      WRITE(ITW,116)IEDF,SSE,AMSE
C     PRINT PREDICTED VALUES AND RESIDUALS
   78 IF(INMD-3)178,79,178
  178 IF(IPRED)79 ,79,251
  251 IF(IPRED-ISTEP)79,151,151
  151 IF(NOIN-IPRED)  79,77,77
   77 LIZ = 40
      IT1 = 1
   16 READ(606'IT1) ID  ,(DATA(I),I=1,N)
      IF (ID) 79,79,17
   17 IF(IREAR) 19,19,18
   18 T = DATA(N)
      DATA(N) = DATA(IREAR)
      DATA(IREAR) = T
   19 YPRED = CNST
      DO  22 I=1,NOIN
      KK = INDEX(I)
   22 YPRED = YPRED & COEN(I) * DATA(KK)
      DEV = DATA(N) - YPRED
      IF(LIZ-40) 25,24,24
   24 LIZ = 0
      NPAGE = NPAGE & 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 241,241,242
  241 WRITE(ITW,101) TITLE,IPROB,NPAGE
  242 WRITE(ITW,109)
   25 LIZ = LIZ & 1
   26 WRITE(ITW,110) ID,DATA(N),YPRED,DEV
      GO TO 16
C     IF VARIANCE CONTRIBUTION INSIGNIFICANT - REMOVE VARIABLE K
   79 IF(IKT)179,179,401
  179 FLEV = ABS(VMIN) * DEFR / R(N,N)
      IF (FLEV - EFOUT) 80,82,82
   80 K =  NOMIN
      DEFR = DEFR &1.0
      NOENT = 0
      GO TO 89
C     IF VARIANCE CONTRIBUTION SIGNIFICANT - ENTER VARIABLE K
   82 DENOM = R(N,N)  - VMAX
      IF(DENOM) 210,210,84
   84 FLEV   = VMAX * DEFR / DENOM
      IF (FLEV - EFIN) 402,402,87
   87 K = NOMAX
      NOENT =  K
      DEFR=DEFR-1.0
C     IF DEGREES OF FREEDOM NON-POSITIVE,TERMINATE JOB
      IF(DEFR) 34,89,89
   34 WRITE(ITW,113)
  401 RETURN
C     IF VARIANCE CRITERION NOT SATISFIED - TERMINATE JOB
   89 IF(K) 90,90,92
   90 WRITE(ITW,115)
      GO TO 401
C     REARRANGE INVERSE FOR ENTERING OR DELETING A VARIABLE
   92 DO 98 I=1,N
      IF(I-K) 94,98,94
   94 DO 97 J=1,N
      IF(J-K)  96,97,96
   96 R(I,J) = R(I,J) - (R(I,K)*R(K,J)/R(K,K))
   97 CONTINUE
   98 CONTINUE
      DO 202 I=1,N
      IF(I-K) 201,202,201
  201 R(I,K) = -R(I,K)/R(K,K)
  202 CONTINUE
      DO 206 J=1,N
      IF(J-K) 205,206,205
  205 R(K,J) = R(K,J)/R(K,K)
  206 CONTINUE
      R(K,K) = 1.0/R(K,K)
C     TEST FOR POSITIVE MEAN SQUARE
      IF(R(N,N)) 210,210,11
  210 WRITE(ITW,112)
      GO TO 401
  402 IKT=1
      IF(INMD-3)404,401,404
  404 IF(IPRED)77,401,401
      END
// DUP
```

```
*STORE      WS  UA  REGRE                                          RGRE2200
// FOR   SUBROUTINE TO READ PARAMETER CARDS  AND DATA
*ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME ANOVA
C     SUBROUTINE TO READ PARAMETER CARDS  AND DATA
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
     1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
      DEFINE FILE 606(500,6,U,IT1)
      DEFINE FILE 607(1000,2,U,IT2)
  101 FORMAT (7I2)
  102 FORMAT (    10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6///10X,17HNUMBER OF F
     1ACTORS,15X,I2/ 10X,10HINPUT MODE,22X,I2/ 10X,21HTRANSFORMATION SWI
     2TCH,11X,I2/ 10X,27HNUMBER OF LEVELS - FACTOR 1,5X,I2/ 10X,27HNUMBE
     3R OF LEVELS - FACTOR 2,5X,I2/ 10X,27HNUMBER OF LEVELS - FACTOR 3,
     45X,I2/ 10X,27HNUMBER OF LEVELS - FACTOR 4,5X,I2)
  103 FORMAT(6I2)
  104 FORMAT(I4,4X,18A4)
 107  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED IN COLUMN',
     1I3,' OF THE ABOVE FORMAT CARD.'/' CHANGE CARD AND RERUN JOB.')
 108  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED AT APPROXIMA
     1TELY COLUMN',I3,' OF THE ABOVE DATA CARD.'/' CHANGE OR REMOVE CARD
     2 AND PRESS START TO CONTINUE.')
  109 FORMAT(//' INVALID INPUT OPTION-JOB TERMINATED ')
C     READ PARAMETER CARDS
      NPAGE=0
      READ(2,103) ICR,ICP,IPR,ITW,IT1,IT2
      IF(IPR)701,702,701
  702 ITW=3
      GO TO 703
  701 ITW=1
  703 READ(ICR,104) IPROB,TITLE
      READ (ICR,101) NF,INMD,ITRN,(NX(I),I=1,4)
      CALL FMAT(IPR,ITW)
      WRITE (ITW,102) TITLE,IPROB,NPAGE,NF,INMD,ITRN,(NX(I),I=1,4)
      IF(INMD-1) 1,1,3
    1 CALL FMTRD(NDIV,IRR)
      CALL PRNTB
      IF(IRR) 2,5,2
    2 WRITE(ITW,107) IRR
      CALL EXIT
C     SUBROUTINE TO READ SOURCE DATA (ANALYSIS OF VARIANCE)
    3 IF(INMD-2)5,5,4
    4 WRITE(ITW,109)
      CALL EXIT
    5 NA = NX(1) + 1
      NB = NX(2) + 1
      NC = NX(3) + 1
      ND = NX(4) + 1
      LS(1)=1
      LS(2) = NA
      LS(3) = LS(2) * NB
      LS(4) = LS(3) * NC
      LS(5)=LS(4)*ND
      J=1
  189 GO TO (10,40),INMD
```

```
C       READ DATA FROM CARD READER
   10 CALL DATRD(NDIV,IRR,IN(1),4,DATA,-1,0,0)
      WRITE(606'J)   (IN(I),I=1,4),DATA
      J=J+1
      IF(IRR) 17,18,17
   17 CALL PRNTB
      WRITE(ITW,108) IRR
      PAUSE 10
      GO TO 10
   18 IF(ITRN) 19,20,19
   19 CALL TRAN
   20 IF (IN(1)) 50,50,21
   21 IS=IN(1)
      DO 30 I=2,NF
   30 IS=IS+LS(I)*(IN(I)-1)
   32 CALL STORE (DATA,IS)
      GO TO 189
C       READ DATA FROM DISC OR TAPE
   40 READ(606'J)    (IN(I),I=1,4),DATA
      J=J+1
      GO TO 18
   50 CALL LINK(ANOV2)
      END
// DUP
*STORE       WS   UA   ANOVA
```

```
// FOR    SUBROUTINE TO STORE A DATUMIN CORE OR DISC
*ONE WORD INTEGERS
C       SUBROUTINE TO STORE A DATUMIN CORE OR DISC
        SUBROUTINE STORE (DATA,IS)
        COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
       1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
        IF (IS-1500) 10,10,20
     10 X(IS)=DATA
        GO TO 30
C       WRITE DATA ON DISC AT LOCATION IS-1500
     20 IST=IS-1500
        WRITE(607'IST) DATA
     30 RETURN
        END
// DUP
*STORE       WS   UA   STORE
```

```
// FOR SUBROUTINE TO GET A DATUM FROM CORE OR DISC
*ONE WORD INTEGERS
C       SUBROUTINE TO GET A DATUM FROM CORE OR DISC
        SUBROUTINE GET (DATA,IS)
        COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
       1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
        IF (IS-1500) 10,10,20
   10   DATA=X(IS)
        GO TO 30
C       GET DATA FROM DISC AT LOCATION IS-1500
   20   IST=IS-1500
        READ(607'IT2) DATA
   30   RETURN
        END
// DUP
*STORE      WS  UA  GET
```

```
// FOR  SECONDARY MAIN PROGRAM - ANALYSIS OF VARIANCE
*ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME ANOV2
C       SECONDARY MAIN PROGRAM - ANALYSIS OF VARIANCE
        COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
       1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(2000)
        DEFINE FILE 606(500,6,U,IT1)
        DEFINE FILE 607(1000,2,U,IT2)
  100   FORMAT(/2X13HJOB COMPLETED)
        CALL SDOP
        CALL MNSQ
        CALL REPRT
        WRITE(ITW,100)
        CALL EXIT
        END
// DUP
*STORE      WS  UA  ANOV2
```

```
// FOR  SUBROUTINE TO PERFORM SIGMA AND DELTA OPERATIONS
*ONE WORD INTEGERS
C      SUBROUTINE TO PERFORM SIGMA AND DELTA OPERATIONS
       SUBROUTINE SDOP
       COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
      1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
 60    NFP1=NF+1
       DO 130 K=1,NF
       NN=NX(K)
       FN=NN
       IS=1
       ISPM=1
   70 SUMX=0.
       DO 80 I=1,NN
       CALL GET (DATA,IS)
       SUMX=SUMX+DATA
   80 IS=IS+LS(K)
       CALL STORE (SUMX,IS)
       DO 90 I=1,NN
       CALL GET (DATA,ISPM)
       DATA=FN*DATA-SUMX
       CALL STORE (DATA,ISPM)
   90 ISPM=ISPM+LS(K)
       ITEST= IS-LS(NF+1)
       IF (ITEST) 100,130,130
  100 IF (ITEST+LS(K)) 110,110,120
  110 IS=IS+LS(K)
       ISPM=ISPM+LS(K)
       GO TO 70
  120 IS=ITEST+LS(K)+1
       ISPM=ISPM+LS(K)+1-LS(NF+1)
       GO TO 70
  130 CONTINUE
       RETURN
       END
// DUP
*STORE      WS  UA  SDOP
```

```
// FOR  SUBROUTINE TO COMPUTE MEAN SQUARE SUMMARYS
*ONE WORD INTEGERS
C      SUBROUTINE TO COMPUTE MEAN SQUARE SUMMARYS
       SUBROUTINE MNSQ
       COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
      1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
C      CLEAR SUMMARY TABLE
       DO 140 I=1,15
       NDIV(I)=0
  140 SMQR(I)=0.0
       IA=1
       IB=1
       IC=1
       ID=1
       I=0
       GO TO 160
  150 CALL GET (DATA,I)
       SMQR(K)=SMQR(K)+DATA**2
       XDEV(K)=DATA
       NDIV(K)=NDIV(K)+1
  160 I=I+1
       IF (IA-NA) 170,320,320
170    IA=IA+1
       IF (IB-NB) 180,250,250
  180 IF (IC-NC) 190,220,220
  190 IF (ID-ND) 200,210,210
  200 K=15
       GO TO 150
  210 K=11
       GO TO 150
  220 IF (ID-ND) 230,240,240
  230 K=12
       GO TO 150
  240 K=5
       GO TO 150
  250 IF (IC-NC) 260,290,290
  260 IF (ID-ND) 270,280,280
  270 K=13
       GO TO 150
  280 K=6
       GO TO 150
  290 IF (ID-ND) 300,310,310
  300 K=7
       GO TO 150
  310 K=1
       GO TO 150
  320 IA=1
       IF (IB-NB) 330,400,400
  330 IB=IB+1
       IF (IC-NC) 340,370,370
  340 IF (ID-ND) 350,360,360
  350 K=14
       GO TO 150
  360 K=8
       GO TO 150
```

```
  370 IF (ID-ND) 380,390,390
  380 K=9
      GO TO 150
  390 K=2
      GO TO 150
  400 IB=1
      IF (IC-NC) 410,440,440
  410 IC=IC+1
      IF (ID-ND) 420,430,430
  420 K=10
      GO TO 150
  430 K=3
      GO TO 150
  440 IC=1
      IF (ID-ND) 450,460,460
  450 ID=ID+1
      K=4
      GO TO 150
  460 CALL GET (DATA,I)
      SMQR(16)=DATA**2
      XDEV(16)=DATA
      RETURN
      END
// DUP
*STORE      WS  UA  MNSQ
```

```
// FOR  SUBROUTINE TO GENERATE ANALYSIS OF VARIANCE TABLES
*ONE WORD INTEGERS
C       SUBROUTINE TO GENERATE ANALYSIS OF VARIANCE TABLES
        SUBROUTINE REPRT
        COMMON ICR,IRP,IPR,ITW,IT1,IT2,IPROB,NPAGE,INMD,NF,ITRN,NA,NB,NC,
      1ND,TITLE(18),NX(5),LS(5),IN(4),NDIV(20),SMQR(20),XDEV(20),X(1500)
        COMMON NDF(15),HEAD(4),INX(15)
  101 FORMAT (      9X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6)
  102 FORMAT(/// 10X,
      130HANALYSIS OF VARIANCE TABLE FOR,1X,I3,3H X ,I3,3H X ,I3,
      23H X ,I3,11H EXPERIMENT,/// 39X,6HSUM OF,5X,10HDEGREES OF,14X,
      34HMEAN/13X,9HCOMPONENT,17X,7HSQUARES,5X,7HFREEDOM,15X,6HSQUARE///)
  103 FORMAT (4A4,I4,15I2)
  104 FORMAT (10X,4A4,4X,F15.2,6X,I5,7X,F15.2)
  105 FORMAT(/18X,8HRESIDUAL,4X,F15.2,6X,I5,7X,F15.2)
  106 FORMAT(/21X, 5HTOTAL,4X,F15.2,6X,I5)
C       FORM DEGREES OF FREEDOM VECTOR FOR COMPONENT MEAN SQUARE
        NDF(1)=NX(1)-1
        NDF(2)=NX(2)-1
        NDF(3)=NX(3)-1
        NDF(4)=NX(4)-1
        NDF(5)= NDF(1)*NDF(2)
        NDF(6)= NDF(1)*NDF(3)
        NDF(7)= NDF(1)*NDF(4)
        NDF(8)= NDF(2)* NDF(3)
        NDF(9)= NDF(2)*NDF(4)
        NDF(10)= NDF(3)*NDF(4)
        NDF(11)= NDF(5)*NDF(3)
        NDF(12)= NDF(5)*NDF(4)
        NDF(13)= NDF(6)*NDF(4)
        NDF(14)= NDF(8)*NDF(4)
        NDF(15)= NDF(11)*NDF(4)
C       COMPUTE DIVISOR AND INITIALIZE COUNTERS
        NN = 1
        DO 6 I = 1,NF
    6   NN = NN *NX(I)
        FN = NN
        KTDFR = NN - 1
        TOTL = 0.0
        NDFRT = 0
C       COMPUTE TOTAL SUM OF SQUARES FOR ALL COMPONENTS
        TOT = 0.0
        DO 9 I = 1,15
        IF (NDIV(I)) 9,9,85
   85   SMQR(I) = SMQR(I) / (NDIV(I)*FN)
        TOT = TOT + SMQR(I)
    9   CONTINUE
C       READ LINE CARD AND PRINT COMPONENT
        KSW=0
    8   READ (ICR,103) (HEAD(I),I=1,4),INDI,(INX(I),I=1,15)
        SMSQ = 0.0
        NDF1 = 0
C       COMPUTE COMPONENT SUM OF SQUARES AND MEAN SQUARE
        DO 20 I = 1,15
        IF (INX(I)) 30,30,10
```

```
   10 K=INX(I)
      SMSQ = SMSQ + SMQR(K)
   20 NDF1=NDF1+NDF(K)
   30 SMSQM=SMSQ/NDF1
C     WRITE TITLE LINE AND COLUMN HEADINGS
      IF(INDI) 40,40,31
   31 NPAGE=NPAGE+1
      CALL FMAT(IPR,ITW)
      IF(IPR) 32,32,40
 32   WRITE (ITW,101) TITLE,IPROB,NPAGE
   40 IF(KSW)401,402,401
  402 WRITE(ITW,102)(NX(I),I=1,4)
      KSW=1
  401 WRITE(ITW,104)(HEAD(I),I=1,4),SMSQ,NDF1,SMSQM
      TOTL = TOTL + SMSQ
      NDFRT = NDFRT + NDF1
      IF(INDI) 50,8,8
C     PRINT RESISUAL AND/OR TITLE LINE
   50 IDIF = KTDFR - NDFRT
      IF (IDIF) 51,52,51
   51 SMSQ = TOT - TOTL
      SMSQM = SMSQ / IDIF
      WRITE (ITW,105) SMSQ,IDIF,SMSQM
   52 WRITE (ITW,106) TOT,KTDFR
      RETURN
      END
// DUP
*STORE      WS  UA  REPRT
```



```
// FOR INPUT DATA SUBROUTINE
*IOCS(CARD,1132PRINTER,DISK)
*ONE WORD INTEGERS
*NAME FCTR
C     INPUT DATA SUBROUTINE
      DEFINE FILE 606(500,65,U,IT1)
      DEFINE FILE 5(30,60,U,IT2)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,IPRED,
     1ICOM,IROT,NFRT,KX(1),MX(20),NCD(3),        ISEQ,NCASE,KCNT,NX(9),
     1TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
      COMMON HIGH(30),HLOW(30),MF(50,3)
  101 FORMAT(6I2)
  102 FORMAT(I4,4X,18A4)
  103 FORMAT(31I2)
  104 FORMAT(20A4)
  105 FORMAT(     10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6//11X,19HNUMBER OF VAR
     1IABLES,46X,I2/11X,10HINPUT TYPE  , 55X,I2/11X,14HSEQUENCE CHECK 51X
     2I2/11X,19HVARIABLES ON CARD 1 46X,I2/11X19HVARIABLES ON CARD 2 46X
     3I2/11X,19HVARIABLES ON CARD 3 46X,I2/11X,21HTRANSFORMATION SWITCH,
     444X,I2)
  106 FORMAT(11X,13HFACTOR SCORES 52X,I2/11X,24HNUMBER OF FACTORS OPTION
     1 41X,I2/11X,37HNUMBER OF FACTORS OR PERCENT OF TRACE 28X,I2/11X,18
     2HCOMMUNALITY OPTION 47X,I2/11X,15HROTATION OPTION 50X,I2/11X,27HNU
     3MBER OF FACTORS TO ROTATE 38X,I2/11X,14HPOOLING OPTION
     451X,I2/11X,14HLATENT VECTORS 51X,I2/11X,23HUNROTATED FACTOR MATRIX
     5 42X,I2/11X,32HORTHOGONAL TRANSFORMATION MATRIX 33X,I2)
  107 FORMAT(11X,24HORTHOGONAL FACTOR MATRIX 41X,I2/11X,59HTRANSFORMATIO
     1N MATRIX TO OBLIQUE REFERENCE VECTOR STRUCTURE 6X,I2/11X,41HOBLIQU
     2E REFERENCE VECTOR STRUCTURE MATRIX 24X,I2/11X,44HCORRELATIONS AMO
     3NG OBLIQUE REFERENCE VECTORS 21X,I2/11X,39HOBLIQUE REFERENCE VECTO
     4R PATTERN MATRIX 26X,I2/11X,58HCORRELATIONS BETWEEN REFERENCE VECT
     5ORS AND PRIMARY FACTORS 7X,I2)
 108  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED AT APPROXIMA
     1TELY COLUMN',I3,' OF THE ABOVE DATA CARD.'/' CHANGE OR REMOVE CARD
     2 AND PRESS START TO CONTINUE.')
  109 FORMAT (                      11X,39HOBLIQUE PRIMARY FACTOR STRUCT
     1URE MATRIX 26X,I2/11X,42HCORRELATIONS AMONG OBLIQUE PRIMARY FACTOR
     2S 23X,I2/11X,37HOBLIQUE PRIMARY FACTOR PATTERN MATRIX 28X,I2/11X,3
     36HFACTOR SCORE REGRESSION COEFFICIENTS 29X,I2)
  110 FORMAT (11X,25HOUTPUT RAW CROSS PRODUCTS 40X,I2/11X 30HOUTPUT RESI
     1DUAL CROSS PRODUCTS35X,I2/11X,28HOUTPUT VARIANCE - COVARIANCE37X,I
     22/11X,18HOUTPUT CORRELATION 47X,I2)
  111 FORMAT(/// 5X 4HCARD I10,  I4,1X 30HIS OUT OF SEQUENCE. RERUN JOB.
     1  )
 112  FORMAT(///' AN ILLEGAL CHARACTER HAS BEEN ENCOUNTERED IN COLUMN',
     1I3,' OF THE ABOVE FORMAT CARD.'/' CHANGE CARD AND RERUN JOB.')
  113 FORMAT(//' INVALID INPUT OPTION-JOB TERMINATED ')
C     SUBROUTINE TO READ PARAMETER CARDS (FACTOR ANALYSIS)
      NPAGE = 0
      READ(2,101) ICR,ICP,IPR,ITW,IT1,IT2
      IF(IPR)701,702,701
  702 ITW=3
      GO TO 703
  701 ITW=1
```

```
  703 READ(ICR,102) IPROB,TITLE
      READ(ICR,103) N,INMD,ISEQ,(NCD(I),I=1,3),MX(20),(MX(I),I=1,4),
     1IPRED,NF,KCNT,ICOM,IROT,NFRT,NX(1),(MX(I),I=5,17)
      CALL FMAT(IPR,ITW)
      WRITE(ITW,105) TITLE,IPROB,NPAGE,N,INMD,ISEQ,(NCD(I),I=1,3),
     1MX(20)
      WRITE(ITW,110) (MX(I),I=1,4)
      WRITE(ITW,106)IPRED,NF,KCNT,ICOM,IROT,NFRT,NX(1),(MX(I),I=5,7)
      WRITE(ITW,107) (MX(I),I=8,13)
      WRITE(ITW,109) (MX(I),I=14,17)
      READ(ICR,104) (VNAME(I),I=1,N)
      IF(INMD-1) 1,1,1001
 1001 IF(INMD-4) 5,1002,1002
 1002 WRITE(ITW,113)
      CALL EXIT
    1 DO 4 I=1,3
      CALL FMTRD(MF(1,I),IRR)
      CALL PRNTB
      IF(IRR) 2,3,2
    2 WRITE(ITW,112) IRR
      CALL EXIT
    3 IF(NCD(I+1)) 5,5,4
    4 CONTINUE
C     INITIALIZATION
    5 DO 8 I=1,N
      HIGH(I) = 0.
      HLOW(I) = 1.0E+36
      SUMY(I) = 0.
      SD(I)=0.0
      DO 8 J=1,N
    8 R(I,J) = 0.0
      KOUNT = 0
      CASES = 0.
      NCASE = 0
    9 IT1 = 1
   10 GO TO (11,41,51),INMD
C     CARD READER INPUT
   11 IST = 1
      I=1
      IF(NCD(1)) 12,12,13
   12 NCD(1) = N
   13 CALL DATRD(MF(1,1),IRR,ID,1,NC,1      ,X,-NCD(1),0,0)
      IF(IRR) 14,15,14
   14 CALL PRNTB
      WRITE(ITW,108) IRR
      PAUSE 10
      GO TO (13,18,18),I
   15 IF(ID) 150,16,16
   16 DO 22 I=2,3
      IF(NCD(I)) 23,23,17
   17 IST = NCD(I-1) + IST
   18 CALL DATRD(MF(1,I),IRR,ID1,1,NC1,1,X(IST),-NCD(I),0,0)
      IF(IRR) 14,19,14
   19 IF(ISEQ) 22,22,20
   20 IF(ID-ID1) 60,21,60
   21 IF(NC1-NC) 60,60,6
    6 ID = ID1
      NC = NC1
   22 CONTINUE
      GO TO 23
   60 WRITE(ITW,111) ID1,NC1
      CALL EXIT
   23 IF(MX(20)) 230,231,230
  230 CALL TRAN
  231 IF(INMD-2) 27,30,27
   27 WRITE(606'IT1) ID ,  (X(I),I=1,N)
C     COMPUTE CROSS PRODUCT MATRIX
   30 CASES = CASES + 1.
      NCASE = NCASE + 1
      DO 35 I = 1,N
      SUMY(I) = SUMY(I) + X(I)
      DO 35 J=I,N
      R(I,J) = R(I,J) + X(I)*X(J)
   35 R(J,I) = R(I,J)
C     DETERMINE HIGH AND LOW VALUES
      DO 39 I=1,N
      IF(X(I) - HIGH(I)) 37,37,36
   36 HIGH(I) = X(I)
   37 IF(X(I) - HLOW(I))38,39,39
   38 HLOW(I) = X(I)
   39 CONTINUE
      GO TO 10
C     READ DATA FROM DISK OR TAPE(360)
   41 READ(606'IT1) ID   ,(X(I),I=1,N)
      IF(ID ) 43,43,23
   43 IT1=1
      GO TO 200
C     READ A MATRIX FROM CARDS
   51 IXOT=IROT
      IROT=NX(1)
      CALL MXRAD
      IROT=IXOT
      IF(INMD-2)150,151,151
  150 WRITE(606'IT1) ID   ,(X(I),I=1,N)
  151 IT1 = 1
  200 IF(NCASE)400,400,300
  300 KNN=1
      CALL LINK(COREL)
  400 CALL LINK(FCTR1)
      END
// DUP
*STORE      WS  UA  FCTR
```

```
// FOR  FACTOR ANALYSIS SETUP PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132PRINTER,DISK)
*NAME FCTR1
C     FACTOR ANALYSIS SETUP PROGRAM
      DEFINE FILE 606(500,65,U,IT1)
      DEFINE FILE 5(30,60,U,IT2)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NCD1,NCD2,NCD3,ISEQ,NCASE,KCNT,NX(9),
     1TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,30)
      COMMON Y(30),B0000(30),B0002
    9 IF(ICOM - 1) 30,10,20
C     MAXIMUM ROW ELEMENT AS COMMUNALITY
   10 R(N,N) = 0.
      DO 12 I=1,N
      R(I,I) = ABS(R(I,N))
      DO 12 J=1,N
      IF(ABS(R(I,J)) - R(I,I)) 12,12,11
   11 R(I,I) = ABS(R(I,J))
   12 CONTINUE
      GO TO 30
C     SQUARED MULTIPLE CORRELATION AS COMMUNALITY
   20 CALL INVRS(R,DATA,N,IERR)
      DO 21 I=1,N
      R(I,I) =1.-1./R(I,I)
      DO 21  J=I,N
   21 R(I,J) =R(J,I)
C     COMPUTE TRACE OF THE MATRIX TO BE FACTORED
   30 TRC = 0.
      DO 31 I=1,N
   31 TRC = TRC + R(I,I)
C       COMPUTE EIGENVALUES.
      CALL TRIDI
      CALL QR
      CALL LINK (FCTR2)
      END
// DUP
*STORE      WS  UA  FCTR1
```

```
// FOR
*ONE WORD INTEGERS
      SUBROUTINE INVRS(R,X,NROW,IERR)
      DIMENSION R(30,30),X(30)
      IERR = 0
      DO 10 K=2,NROW
      M=K-1
      DO 3 KK=1,M
      X(KK) =0.0
      DO 3 J=1,M
      IF(KK-J) 4,4,5
    5 X(KK) = X(KK) +R(J,KK)*R(J,K)
      GO TO 3
    4 X(KK) =X(KK) +R(KK,J)*R(J,K)
    3 CONTINUE
      ALPHA = R(K,K)
      DO 6 I=1,M
    6 ALPHA = ALPHA -X(I)*R(I,K)
      IF(ABS(ALPHA)-1.0E-8) 7,7,8
    7 IERR = 1
      GO TO 20
C     CALCULATE LAST COLUMN  OF NEXT INVERSE
    8 DO 9 I=1,M
      R(I,K) =-X(I)/ALPHA
C     RECALCULATE PREVIOUS INVERSE
      DO 9 J=I,M
    9 R(I,J) =R(I,J) +(X(I)*X(J))/ALPHA
C     CALCULATE R(K,K) ELEMENT OF NEXT INVERSE
      R(K,K) =1.0/ALPHA
   10 CONTINUE
   20 RETURN
      END
// DUP
*STORE      WS  UA  INVRS
```

```
// FOR    XMAX
*ONE WORD INTEGERS
      FUNCTION XMAX(A,B)
      XMAX = A
      IF(A-B)2,1,1
    2 XMAX = B
    1 RETURN
      END
// DUP
*STORE      WS  UA   XMAX
```

```
// FOR   TRANSFORM MATRIX TO TRIDIAGONAL FORM
*ONE WORD INTEGERS
C     REDUCES A REAL SYMMETRIC N BY N MATRIX TO TRIDIAGONAL FORM USING
C     N - 2 ELEMENTARY ORTHOGONAL TRANSFORMATIONS.  THE DIAGONAL
C     ELEMENTS AND THE SUBDIAGONAL ELEMENTS ARE STORED IN ARRAYS
C     ALPHA AND BETA RESPECTIVELY
      SUBROUTINE TRIDI
      DIMENSION GAM(30),V(30)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),P(30),R(30,30)
      COMMON ALPHA(30),BETA(30),ANORM
      ANORM=0.0
      ABSB=0.0
      L=N-2
      DO 4 K=1,L
      ALPHA(K) = R(K,K)
      SIGMA =0.0
      LL=K+1
      DO 5 I=LL,N
    5 SIGMA = R(I,K)*R(I,K)+SIGMA
      ABSB = SQRT(SIGMA)
      T = ABS(ALPHA(K)) + ABSB
      ANORM = XMAX(ANORM,T+ABSB)
      A = R(K+1,K)
      B = SIGN(ABSB,-A)
      BETA(K) = B
      IF(SIGMA) 8,4,8
    8 GAMMA = 1.0 / (SIGMA - A*B)
      GAM(K)=GAMMA
      R(K+1,K)=A - B
      T=0.
      DO 13 I=LL,N
      P(I) = 0.
      DO 11 J=LL,I
   11 P(I) = P(I) + R(I,J)*R(J,K)
      IF(I-N)110,10,10
  110 L1 = I + 1
      DO 12 J=L1,N
   12 P(I) = P(I) + R(J,I)*R(J,K)
 10   P(I) =P(I)*GAMMA
   13 T = T + P(I) * R(I,K)
      T = .5*GAMMA *T
      DO 14 I=LL,N
      P(I) = P(I) - T*R(I,K)
      DO 14 J=LL,I
   14 R(I,J)=R(I,J) - R(I,K)*P(J) - R(J,K)*P(I)
      WRITE(5'K)(R(J,K),J=1,N)
 4    CONTINUE
      ALPHA(N-1)= R(N-1,N-1)
      BETA(N-1) = R(N,N-1)
      ALPHA(N) = R(N,N)
      BETA(N)=0
      T = ABS(BETA(N-1))
      ANORM=XMAX(ANORM , XMAX(ABSB+T+ABS(ALPHA(N-1)) , T+ABS(ALPHA(N))))
```

```
C      FORM TRANSFORMATION MATRIX BY APPLYING THE TRIDIAGONALIZING
C      ROTATIONS TO AN IDENTITY MATRIX.
       DO 402 I=1,N
       DO 403 J=1,N
  403  R(I,J)=0.0
  402  R(I,I)=1.0
       DO 409 I=1,L
       READ(5'I)(V(J),J=1,N)
       II = I + 1
       DO 409 J=2,N
       P(J)=0.0
       DO 408 K=II,N
  408  P(J) = P(J) + R(J,K)*V(K)
       P(J) = P(J) * GAM(I)
       DO 409 K=II,N
  409  R(J,K) = R(J,K) - P(J)*V(K)
       RETURN
       END
// DUP
*STORE      WS  UA  TRIDI
```



```
// FOR  FIND EIGENVALUES OF TRIDIAGONAL MATRIX
*ONE WORD INTEGERS
C      FINDS THE EIGENVALUES OF A TRIDIAGONAL MATRIX BY THE QR METHOD
       SUBROUTINE QR
       DIMENSION A(30),B(30)
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
      1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
       COMMON ALPHA(30),BETA(30),ANORM
       EPSQ = ANORM*ANORM*7.5E-14
C      SET INTERNAL ARRAYS A AND B TO ALPHA AND BETA**2 RESPECTIVELY.
       DO 542 I=1,N
       A(I)=ALPHA(I)
  542  B(I) = BETA(I) * BETA(I)
       AMU = 0.0
        M = N
    1  IF(M-1)100,100,2
    2  I = M - 1
       K = I
       M1 = K
       IF(B(K)-EPSQ)3,3,4
    3  X(M) = A(M)
       AMU = 0.0
       M = K
       GO TO 1
    4  I = I - 1
       IF(I)7,7,5
    5  IF(B(I)-EPSQ)7,7,6
    6  K = I
       GO TO 4
    7  IF(K-M1)9,8,9
C      HANDLE 2 BY 2 BLOCK SEPARATELY.
    8  AMU = A(M1)*A(M) - B(M1)
       SQ1 = A(M1)+A(M)
       SQ2 = A(M1)-A(M)
       SQ2 = SQRT(SQ2*SQ2 + 4.0*B(M1))
       ALAMB = .5*(SQ1+SIGN(SQ2,SQ1)  )
       X(M1)=ALAMB
       X(M) = AMU/ALAMB
       AMU=0.0
       M = M - 2
       IF(M)101,101,1
C      SHORTCUT SINGLE QR ITERATION.
    9  ALAMB = 0.0
       IF(ABS(A(M)-AMU) - .5*ABS(A(M)))10,10,11
   10  ALAMB = A(M) + .5*SQRT(B(M1))
   11  AMU = A(M)
       SQ1=0.0
       SQ2=0.0
         U=0.0
       DO 20 I=K,M1
       GAMMA = A(I)-ALAMB-U
       IF(SQ1-1.0)12,13,12
   12  PQ = GAMMA*GAMMA/(1.0-SQ1)
       GO TO 15
```

```
   13 PQ = 0.0
      IF(I-1)15,15,14
   14 PQ = (1.0-SQ2)*B(I-1)
   15 T = PQ + B(I)
      IF(I-1)17,17,16
   16 B(I-1)=SQ1*T
   17 SQ2 = SQ1
      SQ1 = B(I)/T
      U = SQ1 * (GAMMA + A(I+1) - ALAMB)
      A(I) = GAMMA + U + ALAMB
   20 CONTINUE
      GAMMA = A(M) -ALAMB-U
      IF(SQ1 - 1.0)21,22,21
   21 B(M1)=SQ1*GAMMA*GAMMA/(1.0-SQ1)
      GO TO 23
   22 B(M1) = SQ1*B(M1)*(1.0-SQ2)
   23 A(M) = GAMMA + ALAMB
      GO TO 1
  100 X(1)=A(1)
C     PLACE EIGENVALUES IN ORDER OF DESCENDING VALUE
  101 DO 110 K=1,N
      XMX = -1000.
      DO 105 J=K,N
      IF(X(J) - XMX)105,105,103
  103 XMX = X(J)
      JJ = J
  105 CONTINUE
      X(JJ)=X(K)
      X(K)=XMX
  110 CONTINUE
      RETURN
      END
// DUP
*STORE      WS  UA  QR
```

```
// FOR  FACTOR ANALYSIS OUTPUT PROGRAM
*IOCS(CARD,1132PRINTER,DISK)
*ONE WORD INTEGERS
*NAME FCTR2
C     FACTOR ANALYSIS OUTPUT PROGRAM
      DEFINE FILE 606(500,65,U,IT1)
      DEFINE FILE 5(30,60,U,IT2)
      DIMENSION Y(30)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NCD1,NCD2,NCD3,ISEQ,NCASE,KCNT,NX(9),
     1TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),   X(30),R(30,30)
      COMMON B1(30),B2(30),B3
  100 FORMAT(/2X13HJOB COMPLETED)
  101 FORMAT(    10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6)
  102 FORMAT(//40X,5HTRACE,F15.4,/)
  103 FORMAT(38X,F15.4,15X,F15.4)
  104 FORMAT(38X,20HCHARACTERISTIC ROOTS 10X,23HCUMUL. PERCENT OF TRACE
    3  /,'  ' /)
  105 FORMAT( /10X13HCOMMUNALITIES/)
  106 FORMAT(2X,A4,5X,E15.7)
C     DETERMINE NUMBER OF FACTORS TO COMPUTE
      DO 21 I=1,N
   21 Y(I) = 0.0
      IF(NF-2)32,33,40
   32 NF = 0
      KCNT = 0
      GO TO 50
   33 NF = KCNT
      KCNT = 0
      GO TO 50
   40 NF = 0
   50 SUM = 0.
      PCNT = KCNT
      DO 8 J=1,N
      IF(X(J)) 1,1,2
    1 NF = J-1
      GO TO 9
    2 IF(NF) 4,4,3
    3 IF(NF-J) 9,7,7
    4 IF(KCNT) 5,5,6
    5 IF(X(J) - 1.0) 1,7,7
    6 IF(Y(J) - PCNT) 7,1,1
    7 SUM = SUM + X(J)
    8 Y(J) = SUM*100./TRC
C     COMPUTE CHARACTERISTIC VECTOR FOR FIRST NF CHARACTERISTIC VALUES.
    9 CALL VECTR
C     OUTPUT CHARACTERISTIC VECTORS
      CALL PRNT(5,0,N,NF)
C     COMPUTE FACTOR COEFFICIENTS
      DO 80 J=1,NF
      SQTXJ = SQRT(X(J))
      DO 80 I=1,N
   80 R(I,J) = R(I,J)*SQTXJ
C     OUTPUT CHARACTERISTIC VALUES
```

```
      DIFE = TRC - SUM
      NPAGE = NPAGE + 1
      CALL FMAT(IPR,ITW)
      IF(IPR) 81,81,82
   81 WRITE(ITW,101) TITLE,IPROB,NPAGE
   82 WRITE(ITW,102)TRC
      WRITE(ITW,104)
      DO 90 I=1,N
   90 WRITE(ITW,103)          X(I),Y(I)
C     OUTPUT FACTOR MATRIX
      CALL PRNT(6,0,N,NF)
      WRITE(ITW,105)
      DO 52 I=1,N
      COM=0.
      DO 51 J=1,NF
   51 COM=COM+R(I,J)**2
      WRITE(ITW,106)VNAME(I),COM
   52 CONTINUE
      IF(IROT)16,17,16
16    CALL LINK(FCTR3)
   17 WRITE(ITW,100)
      CALL EXIT
      END
// DUP
*STORE      WS  UA  FCTR2
```



```
// FOR   SUBROUTINE TO OUTPUT RESULTS OF ROTATION
*ONE WORD INTEGERS
C     SUBROUTINE TO OUTPUT RESULTS OF ROTATION
      SUBROUTINE RFOUT
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),H(30),A(30,10)
      COMMON B(10,10),E(10,10)
C     IF KX(1) = 0 ENTRY FROM VARMX
C     IF KX(1) = 1 ENTRY FROM PROMX
      IF(KX(1)) 1,1,10
    1 CALL RPRNT(B,7,1,NFRT,NFRT)
C     OUTPUT ORTHOGONAL FACTOR MATRIX
      CALL RPRNT (A,8,0,N,NFRT)
C     SET B-MATRIX TO IDENTITY FOR FACTOR SCORES
      DO 4 I=1,NFRT
      DO 4 J=1,NFRT
      IF(I-J) 3,2,3
    2 B(I,J) = 1.0
      GO TO 4
    3 B(I,J) = 0.
    4 CONTINUE
      GO TO 100
C     OUTPUT OBLIQUE TRANSFORMATION MATRIX
   10 CALL RPRNT(B,9,1,NFRT,NFRT)
C     OUTPUT CORRELATIONS AMONG OBLIQUE REFERENCE VECTORS
      CALL RPRNT(E,11,1,NFRT,NFRT)
C     OUTPUT OBLIQUE REFERENCE VECTOR STRUCTURE MATRIX
      CALL RPRNT(A,10,0,N,NFRT)
C     COMPUTE INVERSE OF REFERENCE VECTOR CORRELATIONS
      CALL MATIN(E,NFRT)
C     COMPUTE REFERENCE VECTOR PATTERN MATRIX (W)
      DO 5 I=1,NFRT
    5 WRITE(5'I) (A(J,I),J=1,N)
      DO 12 I=1,N
      DO 11 J = 1,NFRT
      H(J) = 0.0
      DO 11 K = 1,NFRT
   11 H(J) = H(J) + A(I,K) * E(K,J)
      DO 12 J = 1,NFRT
   12 A(I,J) = H(J)
      CALL RPRNT(B,12,0,N,NFRT)
C     COMPUTE COR. AMONG REFERENCE VECTORS AND PRIMARY FACTORS
      DO 15 I = 1,NFRT
      DO 15 J = 1,NFRT
      IF (I-J) 14,13,14
   13 B(I,I) = 1. / SQRT(E(I,I))
      GO TO 15
   14 B(I,J) = 0.0
   15 CONTINUE
      CALL RPRNT(B,13,1,NFRT,NFRT)
C     COMPUTE COR. AMONG PRIMARY FACTORS
      DO 21 I = 1,NFRT
   21 H(I) = B(I,I)
      DO 20 I=1,NFRT
```

```
      DO 20 J = 1,NFRT
      IF (I-J) 16,17,16
   16 B(I,J) = E(I,J) *H(I)*H(J)
      GO TO 20
   17 B(I,I) = 1.0
   20 CONTINUE
      CALL RPRNT(B,15,1,NFRT,NFRT)
C      COMPUTE PRIMARY FACTOR STRUCTURE MATRIX  (S)
      DO 30 I = 1,NFRT
      DO 30 J = 1,N
   30 A(J,I) = A(J,I) * H(I)
      CALL RPRNT (B,14,0,N,NFRT)
C     COMPUTE PRIMARY FACTOR PATTERN MATRIX
      DO 40 I = 1,NFRT
      READ(5'I) (A(J,I),J=1,N)
      DO 40 J = 1,N
   40 A(J,I) = A(J,I)/H(I)
      CALL RPRNT(B,16,0,N,NFRT)
  100 RETURN
      END
// DUP
*STORE      WS  UA  RFOUT
```

```
// FOR  SUBROUTINE FOR OBLIQUE ROTATION (PROMAX)
*ONE WORD INTEGERS
C     SUBROUTINE FOR OBLIQUE ROTATION (PROMAX)
      SUBROUTINE PROMX
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),H(30),A(30,10)
      COMMON B(10,10),E(10,10),G(10)
C     COMPUTE A-TRANSPOSE * A
   21 DO 1 I=1,NFRT
      DO 1 J=1,NFRT
      B(I,J) = 0.
      DO 1 K=1,N
    1 B(I,J) = B(I,J) + A(K,I) * A(K,J)
      CALL MATIN(B,NFRT)
      DO 2 I=1,NFRT
      DO 2 J=1,NFRT
      E(I,J)=0.
      DO 2 K=1,N
    2 E(I,J)=E(I,J)+A(K,I)*SIGN((ABS(A(K,J)))**4),A(K,J))
      DO 8 I=1,NFRT
      DO 7 J=1,NFRT
      G(J) = 0.
      DO 7 K=1,NFRT
    7 G(J) = G(J) + B(I,K)*E(K,J)
      DO 8 J=1,NFRT
    8 B(I,J) = G(J)
      DO 10 J=1,NFRT
      T=0.
      DO 9 I=1,NFRT
    9 T = T + B(I,J)**2
      T=SQRT(T)
      DO 10 I=1,NFRT
   10 B(I,J) = B(I,J)/T
C     APPLY TRANSFORMATION MATRIX TO FORM REFERENCE VECTOR STRUCTURE
C     MATRIX
      DO 12 I=1,N
      DO 11 J=1,NFRT
      G(J) = 0.
      DO 11 K=1,NFRT
   11 G(J) = G(J) + A(I,K)*B(K,J)
      DO 12 J=1,NFRT
   12 A(I,J) = G(J)
C     COMPUTE CORRELATIONS AMONG REFERENCE VECTORS
      DO 13 I=1,NFRT
      DO 13 J=1,NFRT
      E(I,J) = 0.
      DO 13 K=1,NFRT
   13 E(I,J) = E(I,J) + B(K,I)*B(K,J)
      KX(1) = 1
      RETURN
      END
// DUP
*STORE      WS  UA  PROMX
```

```
// FOR  SUBROUTINE FOR ORTHOGONAL ROTATION  (VARIMAX)
*ONE WORD INTEGERS
C      SUBROUTINE FOR ORTHOGONAL ROTATION (VARIMAX)
       SUBROUTINE VARMX
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
      1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),H(30),A(30,10)
       COMMON B(10,10)
   101 FORMAT(     10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6)
   102 FORMAT(26XI3,1X2F15.8)
   103 FORMAT(//42X,37HNORMAL VARIMAX CRITERION (NORMALIZED)//25X5HCYCLE6
      1X9HCRITERION5X10HDIFFERENCE5X18HEPSILON CRITERION=,F14.8)
C       INITIALIZE VARIABLES.
       PREV=0.
       IF (NFRT) 50,50,52
    50 IF (NF-10) 51,51,53
    51 NFRT = NF
       GO TO 54
    52 IF (NFRT-10) 54,54,53
    53 NFRT = 10
    54 EPS=0.00116
C      FORM IDENTITY MATRIX FOR TRANSFORMATION
       DO 3 I=1,NFRT
       B(I,I) = 1.0
       DO 3 J=1,NFRT
       IF(I-J) 1,3,1
     1 B(I,J) = 0.
     3 CONTINUE
       LL = NFRT - 1
       NV = 0
       FN=N
       CONS = .7071068
C      NORMALIZE INPUT MATRIX
       DO 5 I = 1,N
       H(I)=0.
       DO 4 J = 1,NFRT
     4 H(I) = H(I) + A(I,J)*A(I,J)
       H(I) = SQRT(H(I))
       DO 5 J = 1,NFRT
     5 A(I,J) = A(I,J) / H(I)
       NPAGE = NPAGE + 1
       CALL FMAT(IPR,ITW)
       IF(IPR) 501,501,6
  501  WRITE(ITW,101) TITLE,IPROB,NPAGE
     6 WRITE(ITW,103)EPS
C      COMPUTE VARIANCE OF SQUARES FOR TO TEST CONVERGENCE.
    61 TV = 0.
       NV=NV+1
       DO 8 J=1,NFRT
       SA = 0.
       SA2 = 0.
       DO 7 I=1,N
        ECCH = A(I,J) * A(I,J)
       SA = SA + ECCH
     7 SA2 = SA2 + ECCH * ECCH
       V = (FN*SA2-SA*SA)/(FN*FN)
     8 TV = TV + V
       DIFFR = TV - PREV
       PREV = TV
       WRITE(ITW,102) NV,TV,DIFFR
       IF(NV - 50)9,999,999
C      IS THE VARIANCE ON THIS CYCLE EQUAL(APPROXIMATELY) TO LAST CYCLES'
     9 IF(ABS(DIFFR)-.000001) 999,999,13
C      BEGIN NEXT ITERATION CYCLE
    13 DO 40 J=1,LL
       II=J+1
       DO 40 K=II,NFRT
C      COMPUTE THE NUMERATOR AND DENOMINATOR OF THE TANGENT OF THETA.
       AA=0.0
       BB=0.0
       CC=0.0
       DD=0.0
       DO 15 I = 1,N
       XX=A(I,J)
       YY=A(I,K)
       UU = (XX + YY) * (XX - YY)
       VV = 2.0 * XX * YY
       CC = CC + ( UU + VV)*(UU - VV)
       DD = DD + UU * VV
       AA = AA + UU
    15 BB = BB + VV
       T = 2.0 * (DD - AA * BB/FN)
       Z = CC - (AA * AA - BB*BB)/FN
       IF(T - Z)18,16,22
    16 IF((T+Z)-EPS) 40,17,17
    17 COST = .9807853
       SINT = .1950903
C       THE SIN AND COSINE OF  11 DEGREES, 15 MINUTES  ( 45/4 DEGREES )
       GO TO 26
    18 TAN4T = ABS(T/Z)
       IF(TAN4T-EPS) 20,19,19
    19 COS4T=1.0/SQRT(1.0+TAN4T**2)
       SIN4T=TAN4T*COS4T
       GO TO 25
    20 IF(Z)  21,40,40
    21 SINP=CONS
       COSP=CONS
       GO TO 31
C        IF THE NUMERATOR IS MORE THAN THE DENOMINATOR,REVERSE THE TWO.
    22 CTN4T = ABS(Z/T)
       IF(CTN4T - EPS) 24,23,23
C    COMPUTE SUCCESIVELY COS 2T,SIN 2T, COS T, SIN T.
    23 SIN4T = 1.0/SQRT(1.0+CTN4T**2)
       COS4T = CTN4T*SIN4T
       GO TO 25
    24 COS4T = 0.0
       SIN4T = 1.0
    25 COS2T = CONS* SQRT(1. + COS4T)
       SIN2T=SIN4T/(2.0*COS2T)
       COST = CONS * SQRT(1. + COS2T)
```

```
      SINT=SIN2T/(2.0*COST)
   26 IF(Z) 28,28,27
   27 COSP=COST
      SINP=SINT
      GO TO 29
C      IF DENOMINATOR IS NEGATIVE,  SUBTRACT 45 DEGREES FROM THE ANGLE.
   28 COSP = CONS * (COST + SINT)
      SINP = ABS(CONS * (COST - SINT))
   29 IF(T) 30,30,31
C   IF NUMERATOR WAS NEGATIVE, SUBTRACT 90 DEGREES FROM THE ANGLE.
   30 SINP=-SINP
C     MULTIPLY THE TWO COLUMNS TO BE ROTATED BY THE MATRIX OF SINES AND
C    COSINES
   31 DO 32 I=1,N
      AIJ = A(I,J) * COSP + A(I,K) * SINP
      AIK = -A(I,J)*SINP + A(I,K)*COSP
      A(I,J)=AIJ
   32 A(I,K)=AIK
C    ROTATE THE CORRESPONDING COLUMNS OF THE IDENTITY MATRIX TO OBTAIN
C    THE TRANSFORMATION MATRIX.
      DO 33 I=1,NFRT
      COST = B(I,J) * COSP + B(I,K) * SINP
      SINT = B(I,K) * COSP - B(I,J) * SINP
      B(I,J) = COST
   33 B(I,K) = SINT
   40 CONTINUE
      GO TO 61
  999 KX(1) = 0
      DO 2  I=1,N
      DO 2 J=1,NFRT
    2 A(I,J)=A(I,J)*H(I)
      RETURN
      END
// DUP
*STORE      WS  UA  VARMX
```

```
// FOR  FIND EIGENVECTORS OF TRIDIAGONAL MATRIX
*ONE WORD INTEGERS
C     FIND THE EIGENVECTORS OF THE TRIDIAGONAL MATRIX BY
C     THE METHOD OF J. H. WILKINSON
      SUBROUTINE VECTR
      DIMENSION CONS(30),VECT(30)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,30)
      COMMON ALPHA(30),BETA(30),XX
      DO 500 K=1,NF
      XX = X(K)
C     INITIALIZE RIGHT SIDE OF EQUATIONS TO BE SOLVED TO ONES.
      DO 1 I=1,N
    1 CONS(I) = 1.0
      OLDH = 1.0
      DO 100 IJK = 1,10
      CALL COVEC(CONS,VECT)
      H = 0.0
      DO 2 I=1,N
      IF(ABS(H) - ABS(VECT(I)))11,2,2
   11 H = VECT(I)
    2 CONTINUE
      DO 4 I=1,N
      IF(ABS(CONS(I)/OLDH - VECT(I)/H) - 5.0E-2)4,45,45
    4 CONTINUE
      GO TO 200
C     IF RESULTS DO NOT CONVERGE,SET RIGHT-HAND SIDE TO LAST APPROX.
C                   AND LOOP
   45 OLDH = H
      DO 100 J=1,N
  100 CONS(J) = VECT(J)
C     ONCE APPROX. SOLUTION HAS BEEN FOUND, REFINE IT TO FIVE PLACES.
  200 CONS(1)=CONS(1)-VECT(1)*(ALPHA(1)-XX) - VECT(2)*BETA(1)
      DO 201 I=2,N
  201 CONS(I) = CONS(I)  -  VECT(I-1) * BETA(I-1)  -  VECT(I+1) * BETA(I
     1)  -  VECT(I) * (ALPHA(I)-XX)
      CALL COVEC(CONS,CONS)
      H = 0.0
      DO 212 I=1,N
      VECT(I) = VECT(I) + CONS(I)
      IF(ABS(H)-ABS(VECT(I)))211,212,212
  211 H = VECT(I)
  212 CONTINUE
C     REDUCE MAGNITUDE OF EIGENVECTOR TO PREVENT POSSIBLE OVERFLOWS.
      DO 3 I=1,N
    3 CONS(I) = VECT(I) / H
C     TRANSFORM EIGENVECTOR TO CORRESPONDING VECTOR OF ORIGINAL MATRIX
C     AND NORMALIZE
      H = 0.0
      DO 206 I=1,N
      VECT(I) = 0.0
      DO 205 J=1,N
  205 VECT(I) = VECT(I) + CONS(J)*R(I,J)
  206 H = H + VECT(I)*VECT(I)
```

```
      H = SQRT(H)
      DO 210 I=1,N
  210 VECT(I) = VECT(I) / H
      WRITE(5'K)(VECT(I),I=1,N)
  500 CONTINUE
      DO 600 K=1,NF
      READ(5'K)(R(I,K),I=1,N)
  600 CONTINUE
      RETURN
      END
// DUP
*STORE      WS  UA  VECTR
```

```
// FOR  SOLVE SIMULTANEOUS TRIDIAGONAL EQUATIONS
*ONE WORD INTEGERS
C     PERFORM A SINGLE ITERATION OF WILKINSONS METHOD
      SUBROUTINE COVEC(CONS,VECT)
C     SOLVES THE SYSTEM OF EQUATIONS WHOSE GENERAL FORM IS-
C     BETA(I)*X(I-1) +(ALPHA(I)-XX)*X(I) + BETA(I)*X(I+1) = CONS(I)
C     FOR X(1 - N) , WHERE BETA(0)=BETA(N+1)=0. AND XX IS AN EIGENVALUE
C     OF THE TRIDIAGONAL MATRIX DETERMINED BY THE ALPHAS AND BETAS.
      DIMENSION CONS(30),VECT(30)
      DIMENSION U(29),V(29),W(29)
      COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
     1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
      COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,30)
      COMMON ALPHA(30),BETA(30),XX
      C = CONS(1)
      P = ALPHA(1) - XX
      Q = BETA(1)
      I = 2
   10 PP = BETA(I-1)
      QQ = ALPHA(I) - XX
      RR = BETA(I)
C     SELECT MAXIMUM COEFFICIENT OF X(I) AS I TH PIVOT
    4 IF(ABS(PP)-ABS(P))2,3,3
    3 U(I-1)= CONS(I)/PP
      V(I-1)=-QQ/PP
      W(I-1)=-RR/PP
      XP = P
      P = XP * V(I-1) + Q
      Q = XP * W(I-1)
      C = C - XP * U ( I - 1 )
      GO TO 5
    2 U(I-1)= C/P
      V(I-1) = -Q/P
      W(I-1)=0.0
      P = QQ + PP*V(I-1)
      Q = RR
      C = CONS(I) - PP*U(I-1)
    5 I = I + 1
      IF(I - N)10,11,12
   11 PP = BETA(N-1)
      QQ = ALPHA(N) - XX
      RR = 0.0
      GO TO 4
   12 VECT(N) = C/P
C       BACK SUBSTITUTION
   14 DO 20 I=2,N
      J = N+1-I
   20 VECT(J) = U(J)+V(J)*VECT(J+1)+W(J)*VECT(J+2)
   40 RETURN
      END
// DUP
*STORE      WS  UA  COVEC
```

```
// FOR   ROTATIONS PACKAGE FOR FACTOR ANALYSIS                              FCT3   0
*ONE WORD INTEGERS                                                          FCT3  10
*IOCS(CARD,1132PRINTER,DISK)                                                FCT3  20
*NAME FCTR3                                                                 FCT3  30
C      ROTATIONS PACKAGE FOR FACTOR ANALYSIS                                FCT3  40
       DEFINE FILE 606(500,65,U,IT1)                                        FCT3  50
       DEFINE FILE 5(30,60,U,IT2)                                           FCT3  60
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,     FCT3  70
      1ICOM,IROT,NFRT,KX(1),MX(20),NCD1,NCD2,NCD3,ISEQ,NCASE,NX(10),TRC,    FCT3  80
      2FLVB(4),KNN                                                          FCT3  90
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),R(30,10)            FCT3 100
       COMMON B(10,10),E(10,10),G(10)                                       FCT3 110
  100 FORMAT(/2X13HJOB COMPLETED)                                           FCT3 120
    4 IF(IROT-1) 9,5,5                                                      FCT3 130
    5 CALL VARMX                                                            FCT3 140
      CALL RFOUT                                                            FCT3 150
      IF(IROT-2) 7,6,6                                                      FCT3 160
    6 CALL PROMX                                                            FCT3 170
      CALL RFOUT                                                            FCT3 180
    7 IF(ISCR) 9,9,8                                                        FCT3 190
    8 CALL SCORE                                                            FCT3 200
    9 WRITE(ITW,100)                                                        FCT3 210
      CALL EXIT                                                             FCT3 220
      END                                                                   FCT3 230
// DUP                                                                      FCT3 240
*STORE      WS  UA  FCTR3                                                   FCT3 250
```

```
// FOR  MATRIX PRINT/PUNCH ROUTINE FOR ROTATION                             RPNT   0
*      ONE WORD INTEGERS                                                    RPNT  10
C      MATRIX PRINT/PUNCH ROUTINE FOR ROTATION                              RPNT  20
       SUBROUTINE RPRNT(B,MID,KODE,NR,NC)                                   RPNT  30
       DIMENSION B(10,10)                                                   RPNT  40
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,KX(5),    RPNT  50
      1MX(20),NX(15),FLVB(5),KNN                                            RPNT  60
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),DATA(30),R(30,10)         RPNT  70
  101 FORMAT(5XA4,4X8F12.4)                                                 RPNT  80
  102 FORMAT(3XI4,6X8F12.4)                                                 RPNT  90
  103 FORMAT(     10X,18A4,5X,3HJOB,I7,5X,4HPAGE,I6)                        RPNT 100
  104 FORMAT(I4,3I2,5E14.7)                                                 RPNT 110
  105 FORMAT(/103H READY THE PUNCH WITH BLANK CARDS AND PRESS START ON TRPNT 120
     1HE PUNCH AND CONSOLE.  TURN CONSOLE SWITCH 15 ON.)                    RPNT 130
  106 FORMAT(1H )                                                           RPNT 140
  201 FORMAT(3X     ,8HVARIABLE,7X,8(I4,8X)///)                             RPNT 150
  327 FORMAT(/45X 32HORTHOGONAL TRANSFORMATION MATRIX    )                  RPNT 160
  328 FORMAT(/41X 33HORTHOGONAL FACTOR MATRIX(VARIMAX))                     RPNT 170
  329 FORMAT(/45X 50HTRANSFORMATION TO OBLIQUE REFERENCE VECTOR STRCTR.)RPNT 180
  330 FORMAT(/45X 41HOBLIQUE REFERENCE VECTOR STRUCTURE MATRIX )            RPNT 190
  331 FORMAT(/45X 44HCORRELATIONS AMONG OBLIQUE REFERENCE VECTORS)          RPNT 200
  332 FORMAT(/45X 39HOBLIQUE REFERENCE VECTOR PATTERN MATRIX )              RPNT 210
  333 FORMAT(/45X 48HCORR. BET. REFERENCE VECTORS AND PRIMARY FACTORS ) RPNT 220
  334 FORMAT(/45X 39HOBLIQUE PRIMARY FACTOR STRUCTURE MATRIX )              RPNT 230
  335 FORMAT(/45X 35HCORR. AMONG OBLIQUE PRIMARY FACTORS   )                RPNT 240
  336 FORMAT(/45X 31HOBLIQUE PRIMARY FACTOR LOADINGS)                       RPNT 250
  337 FORMAT(/45X 36HFACTOR SCORE REGRESSION COEFFICIENTS )                 RPNT 260
      IF(MX(MID)-1)1000,1,100                                               RPNT 270
    1 I = 1                                                                 RPNT 280
      II = 8                                                                RPNT 290
      ISW = MID-6                                                           RPNT 300
    9 IF(NC-II) 10,11,11                                                    RPNT 310
   10 II = NC                                                               RPNT 320
   11 NPAGE = NPAGE + 1                                                     RPNT 330
      CALL FMAT(IPR,ITW)                                                    RPNT 340
      IF(IPR) 111,111,112                                                   RPNT 350
  111 WRITE(ITW,103)TITLE,IPROB,NPAGE                                       RPNT 360
  112 GO TO (21,22,23,24,25,26,27,28,29,30,31),ISW                          RPNT 370
   21 WRITE(ITW,327)                                                        RPNT 380
      GO TO 32                                                              RPNT 390
   22 WRITE(ITW,328)                                                        RPNT 400
      GO TO 32                                                              RPNT 410
   23 WRITE(ITW,329)                                                        RPNT 420
      GO TO 32                                                              RPNT 430
   24 WRITE(ITW,330)                                                        RPNT 440
      GO TO 32                                                              RPNT 450
   25 WRITE(ITW,331)                                                        RPNT 460
      GO TO 32                                                              RPNT 470
   26 WRITE(ITW,332)                                                        RPNT 480
      GO TO 32                                                              RPNT 490
   27 WRITE(ITW,333)                                                        RPNT 500
      GO TO 32                                                              RPNT 510
   28 WRITE(ITW,334)                                                        RPNT 520
      GO TO 32                                                              RPNT 530
   29 WRITE(ITW,335)                                                        RPNT 540
```

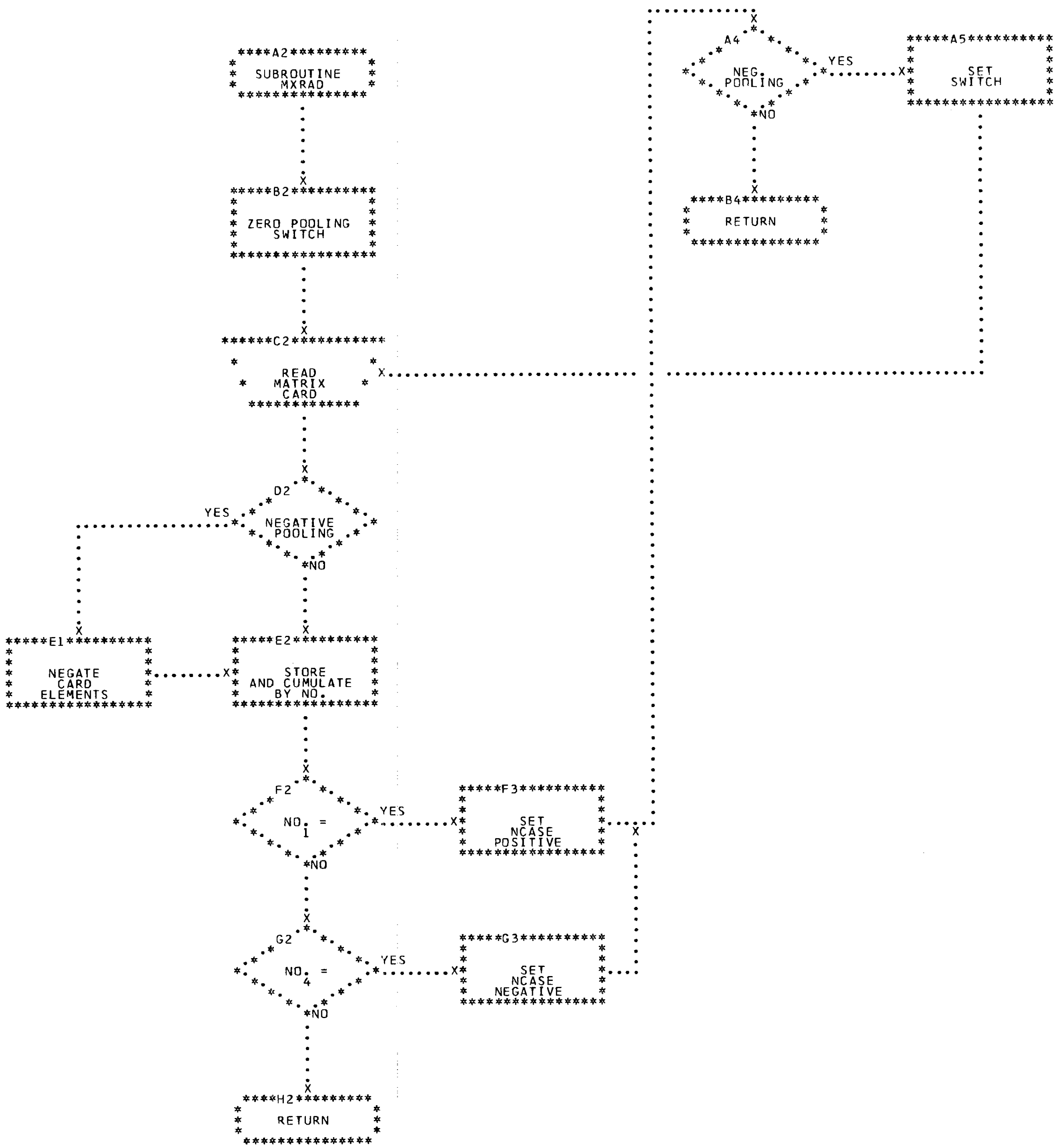
```
      GO TO 32
   30 WRITE(ITW,336)
      GO TO 32
   31 WRITE(ITW,337)
   32 WRITE(ITW,201)(J,J=I,II)
      DO 35 K=1,NR
      IF(KODE) 34,33,34
   33 WRITE(ITW,101) VNAME(K),(R(K,J),J=I,II)
      GO TO 35
   34 WRITE(ITW,102) K,(B(K,J),J=I,II)
   35 CONTINUE
      IF(NC-II) 36,1000,36
   36 I = I+8
      II = II + 8
      GO TO 9
C     PUNCH ROUTINE
  100 I = 1
      II = 5
      READ(ICR,106)
      CALL DATSW(15,JIG)
      IF(JIG-2)151,3,3
    3 WRITE(ITW,105)
      PAUSE
  151 IF(NC-II) 152,153,153
  152 II = NC
  153 DO 156 K = 1,NR
      IF(KODE)     154,154,155
  154 WRITE(ICP,104)IPROB,MID,I  ,K,(R(K,J),J=I,II)
      GO TO 156
  155 WRITE(ICP,104)IPROB,MID,I  ,K,(B(K,J),J=I,II)
  156 CONTINUE
      IF(NC-II)157,158,157
  157 I = I + 5
      II = II + 5
      GO TO 151
  158 IF(MX(MID)-2) 1000,1,1000
 1000 RETURN
      END
// DUP
*STORE      WS  UA  RPRNT
```

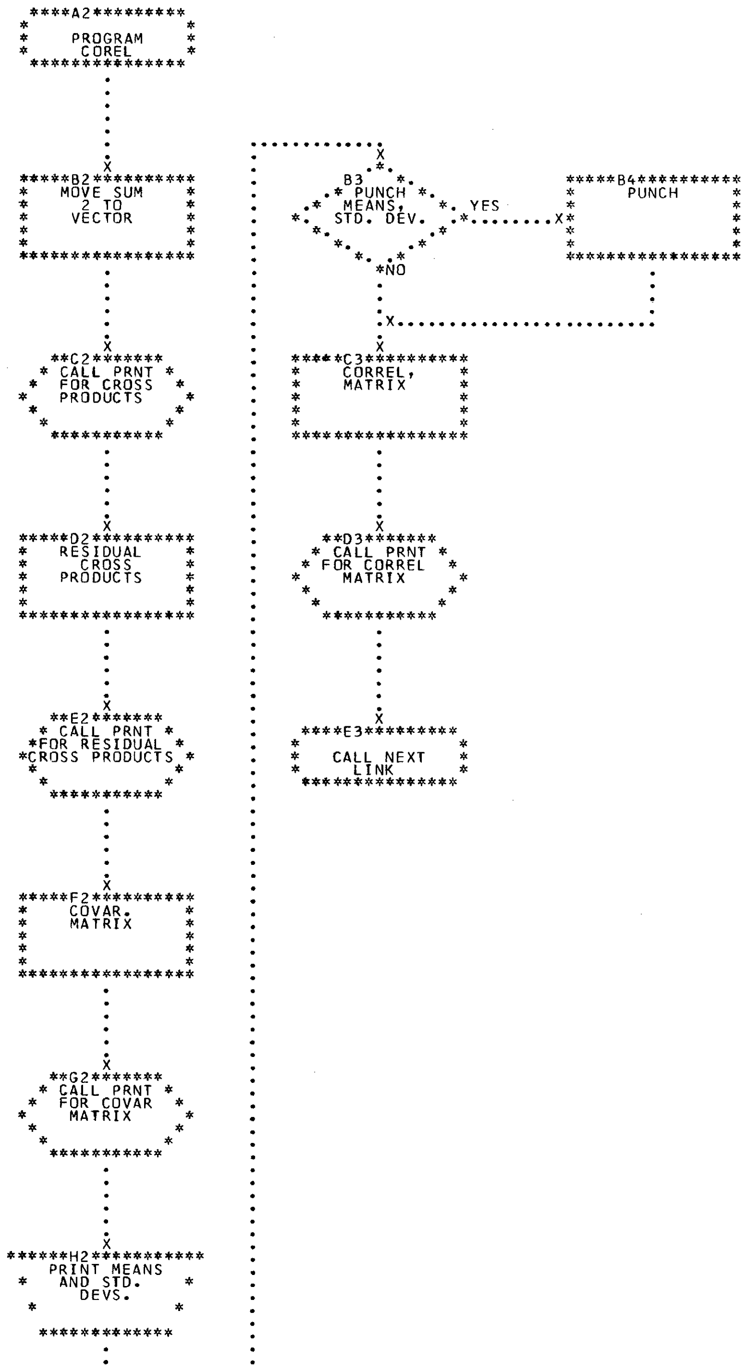
```
// FOR   SUBROUTINE TO INVERT A MATRIX
*ONE WORD INTEGERS
C     SUBROUTINE TO INVERT A MATRIX
      SUBROUTINE MATIN(A,N)
      DIMENSION IPIV(10),A(10,10),INDEX(10,2),PIVOT(10)
      DO 20 J = 1,N
   20 IPIV(J) = 0
      DO 550 I = 1,N
      AMAX = 0.0
      DO 105 J = 1,N
      IF(IPIV(J) - 1)60,105,60
   60 DO 100 K = 1,N
      IF(IPIV(K) - 1 )80,100,740
   80 IF(ABS(AMAX) - ABS(A(J,K)))85,100,100
   85 IROW = J
      ICOLM = K
      AMAX = A(J,K)
  100 CONTINUE
  105 CONTINUE
      IPIV(ICOLM) = IPIV(ICOLM) + 1
      IF(IROW - ICOLM)150,260,150
  150 DO 200 L = 1,N
      SWAP = A(IROW,L)
      A(IROW,L) = A(ICOLM,L)
  200 A(ICOLM,L) = SWAP
  260 INDEX(I,1) = IROW
      INDEX(I,2) = ICOLM
      PIVOT(I) = A(ICOLM,ICOLM)
      A(ICOLM,ICOLM) = 1.0
      DO 350 L = 1,N
  350 A(ICOLM,L) = A(ICOLM,L) / PIVOT(I)
      DO 550 L1 = 1,N
      IF(L1 - ICOLM)400,550,400
  400 T = A(L1,ICOLM)
      A(L1,ICOLM) = 0.0
      DO 450 L = 1,N
  450 A(L1,L) = A(L1,L) - A(ICOLM,L) * T
  550 CONTINUE
      DO 710 I = 1,N
      L = N + 1 - I
      IF(INDEX(L,1) - INDEX(L,2))630,710,630
  630 JROW = INDEX(L,1)
      JCOLM = INDEX(L,2)
      DO 705 K = 1,N
      SWAP = A(K,JROW)
      A(K,JROW) = A(K,JCOLM)
  705 A(K,JCOLM) = SWAP
  710 CONTINUE
  740 RETURN
      END
// DUP
*STORE      WS  UA  MATIN
```

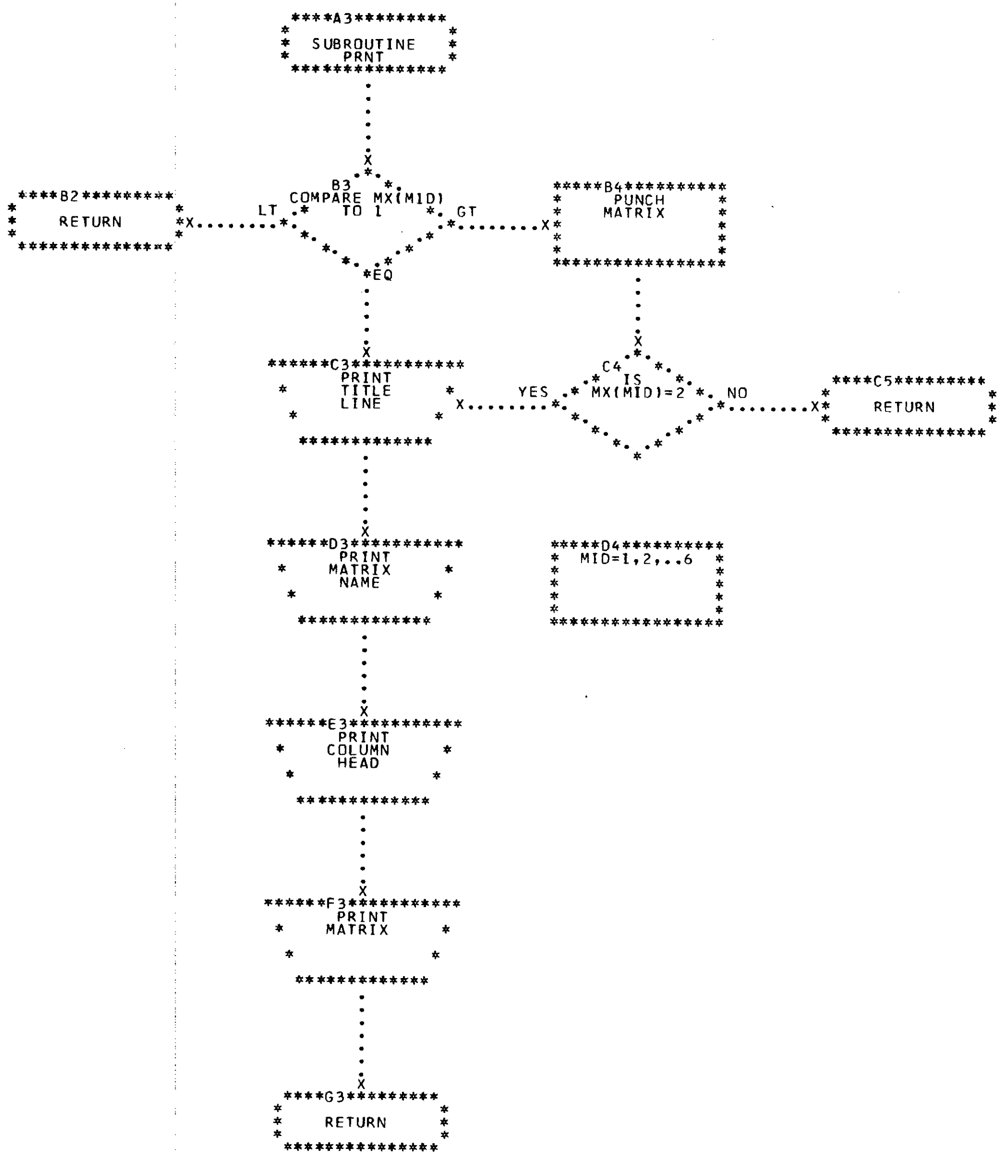
```
// FOR   SUBROUTINE TO COMPUTE FACTOR SCORES
*ONE WORD INTEGERS
C      SUBROUTINE TO COMPUTE FACTOR SCORES
       SUBROUTINE SCORE
       COMMON ICR,ICP,IPR,ITW,IT1,IT2,IPROB,N,NF,CASES,NPAGE,INMD,ISCR,
      1ICOM,IROT,NFRT,KX(1),MX(20),NX(15),TRC,FLVB(4),KNN
       COMMON TITLE(18),VNAME(30),SUMY(30),SD(30),X(30),A(30,10)
       COMMON B(10,10),Z(10)
  101 FORMAT(     10X,18A4,5X,3HJOB,I7,5X4HPAGE,I6)
  102 FORMAT(//45X,13HFACTOR SCORES,// 2X14HIDENTIFICATION2X,5I14,/18X5I
      114//)
  103 FORMAT( 1XI5,1X,I7,6X,5E14.5/20X,5E14.5)
  104 FORMAT(I4,I2,5E14.7/5E14.7)
  105 FORMAT(1H )
  106 FORMAT(/103H READY THE PUNCH WITH BLANK CARDS AND PRESS START ON T
      1HE PUNCH AND CONSOLE.  TURN CONSOLE SWITCH 15 ON.)
       I25=25
C      COMPUTE COMMUNALITIES
       DO 1 I=1,N
       X(I) =0.
       DO 10 J=1,NFRT
   10 X(I) = X(I) - A(I,J)**2
    1 X(I)=1.+X(I)
       DO 8 J=1,N
       DO 11 I=1,NFRT
   11 Z(I)=A(J,I)/X(J)
    8 WRITE(5'J) (Z(K),K=1,NFRT)
       CALL MATIN(B,NFRT)
C      A-TRANSPOSE * UNIQUENESS * A, ZDDED TO PHI
       DO 13 I=1,NFRT
       DO 12 J=1,NFRT
       Z(J) = 0.0
       DO 12 K=1,N
   12 Z(J) = Z(J) + A(K,J)*A(K,I)/X(K)
       DO 13 J=1,NFRT
   13 B(I,J) = B(I,J) + Z(J)
       CALL MATIN(B,NFRT)
C      COMPUTE FACTOR SCORE COEFFICIENTS
       DO 14 I=1,N
       READ(5'I) (Z(L),L=1,NFRT)
       DO 14 J=1,NFRT
       A(I,J) = 0.
       DO 14 K=1,NFRT
   14 A(I,J) = A(I,J) + B(J,K)*Z(K)
       CALL RPRNT(B,17,0,N,NFRT)
C      COMPUTE FACTOR SCORES
       IT1 = 1
       IF(ISCR-2)302,303,302
  303 READ(ICR,105)
       CALL DATSW(15,JIG)
       IF(JIG-2)302,3,3
    3 WRITE(ITW,106)
       PAUSE
  302 LINES = 79
       II=0
   18 READ(606'IT1)ID,   (X(I),I=1,N)
       IF(ID) 50,50,19
   19 DO 20 I=1,N
   20 X(I) = (X(I)-SUMY(I))/SD(I)
       DO 21 J=1,NFRT
       Z(J) = 0.
       DO 21 I=1,N
   21 Z(J) = Z(J) + A(I,J)*X(I)
C      OUTPUT FACTOR SCORES
       IF(LINES-79) 26,25,25
   25 NPAGE = NPAGE + 1
       LINES = 0
       CALL FMAT(IPR,ITW)
       IF(IPR) 251,251,26
  251  WRITE(ITW,101) TITLE,IPROB,NPAGE
   26 IF(LINES)41,42,41
   42 WRITE(ITW,102)(K,K=1,NFRT)
   41 LINES = 2+LINES + (NFRT-1)/10
       II = II + 1
   30 WRITE(ITW,103) II, ID, (Z(J),J=1,NFRT)
       IF(ISCR-2)18,301,18
  301 WRITE(ICP,104)II,I25,(Z(J),J=1,NFRT)
       GO TO 18
   50 RETURN
       END
// DUP
*STORE      WS  UA  SCORE
```

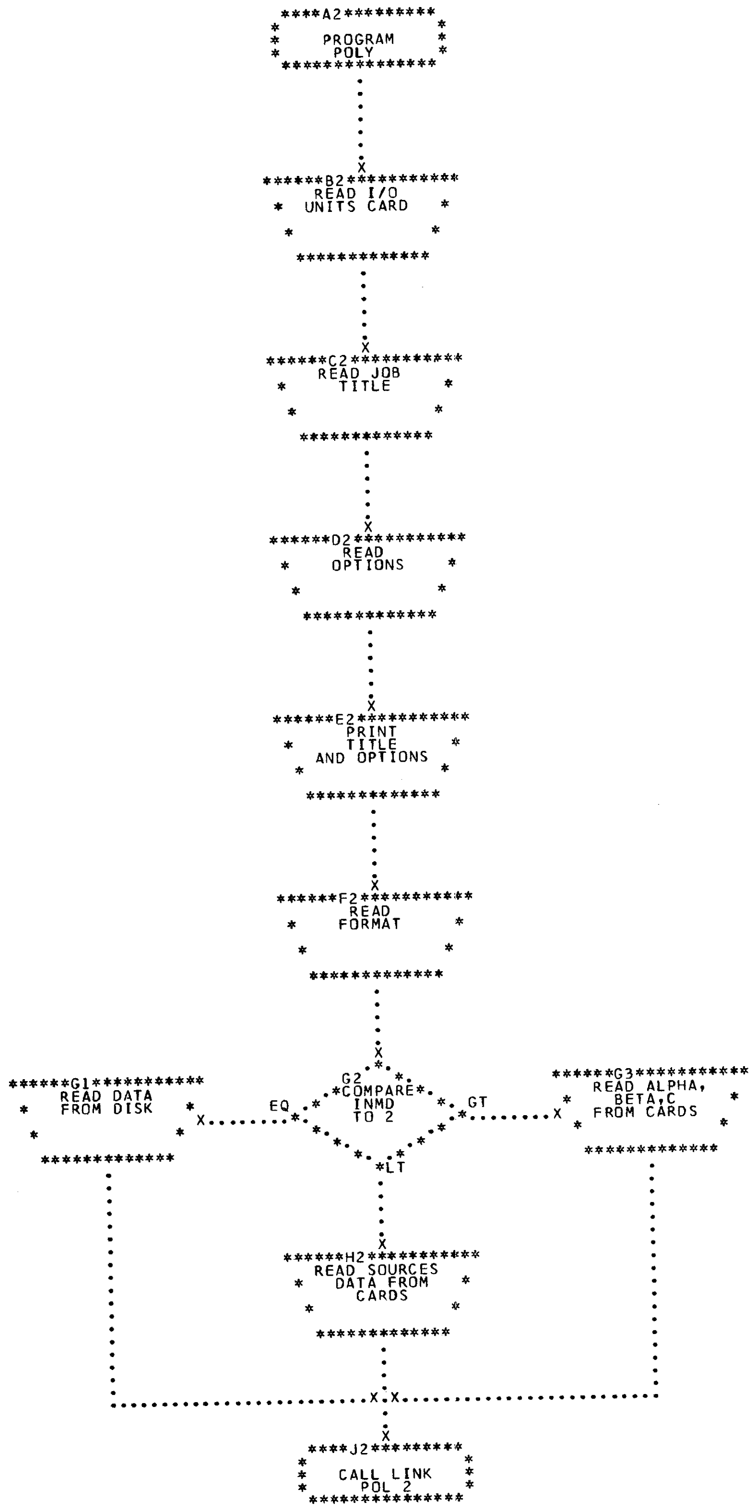
```
// FOR
*ONE WORD INTEGERS
      SUBROUTINE FMAT(IPR,ITW)
      IF (IPR) 1,1,2
    1 WRITE(ITW,100)
      GO TO 3
    2 WRITE(ITW,101)
    3 RETURN
  100 FORMAT(1H1)
  101 FORMAT(//'  '//)
      END
// DUP
*STORE      WS  UA  FMAT

1F0031217
```

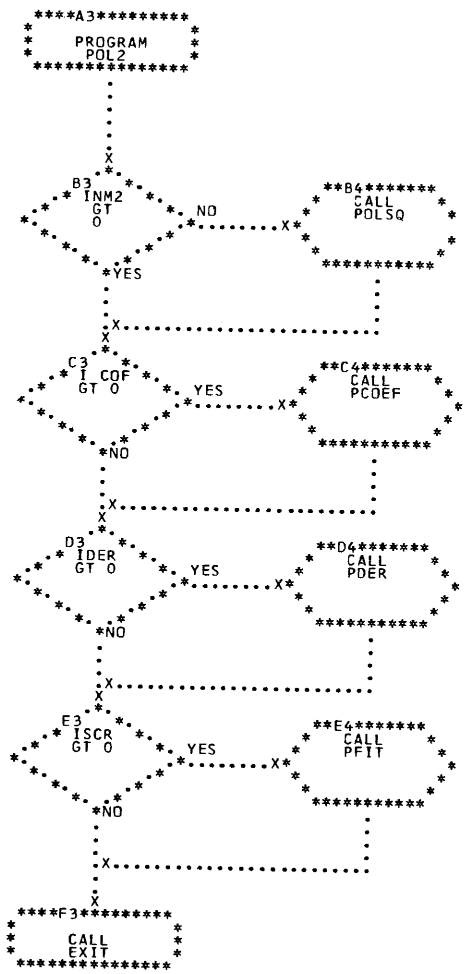
CHART DA

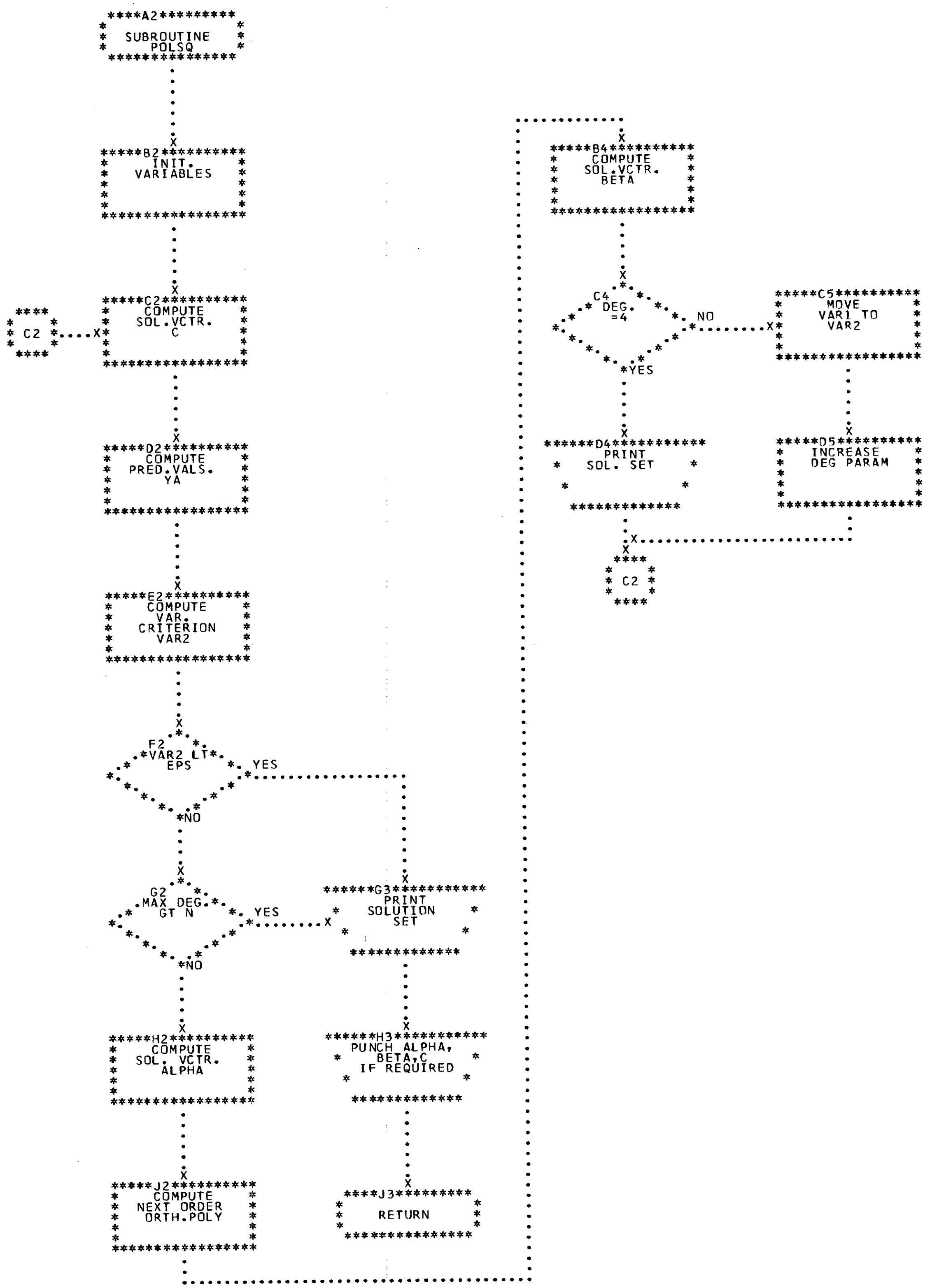

A2
SUBROUTINE MXRAD
B2
ZERO POOLING SWITCH
C2
READ MATRIX CARD
D2
NEGATIVE POOLING
YES
NO
E1
NEGATE CARD ELEMENTS
E2
STORE AND CUMULATE BY NO.
F2
NO. = 1
YES
NO
F3
SET NCASE POSITIVE
G2
NO. = 4
YES
NO
G3
SET NCASE NEGATIVE
H2
RETURN
A4
NEG. POOLING
YES
NO
A5
SET SWITCH
B4
RETURN

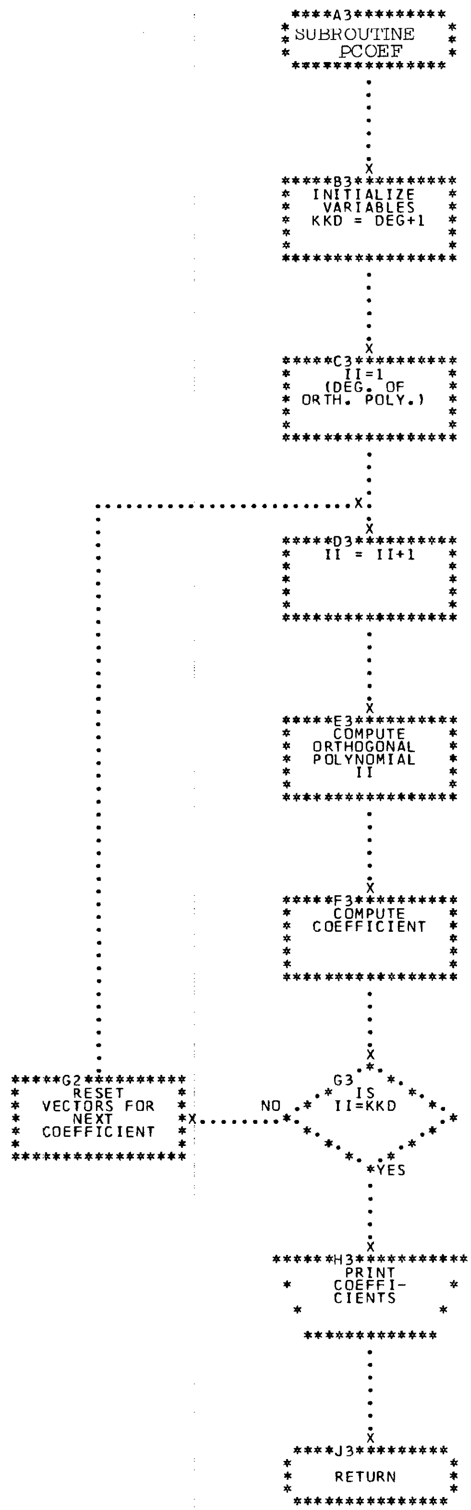

A2
PROGRAM COREL
B2
MOVE SUM 2 TO VECTOR
C2
CALL PRNT FOR CROSS PRODUCTS
D2
RESIDUAL CROSS PRODUCTS
E2
CALL PRNT FOR RESIDUAL CROSS PRODUCTS
F2
COVAR. MATRIX
G2
CALL PRNT FOR COVAR MATRIX
H2
PRINT MEANS AND STD. DEVS.
B3
PUNCH MEANS, STD. DEV.
YES
NO
B4
PUNCH
C3
CORREL, MATRIX
D3
CALL PRNT FOR CORREL MATRIX
E3
CALL NEXT LINK

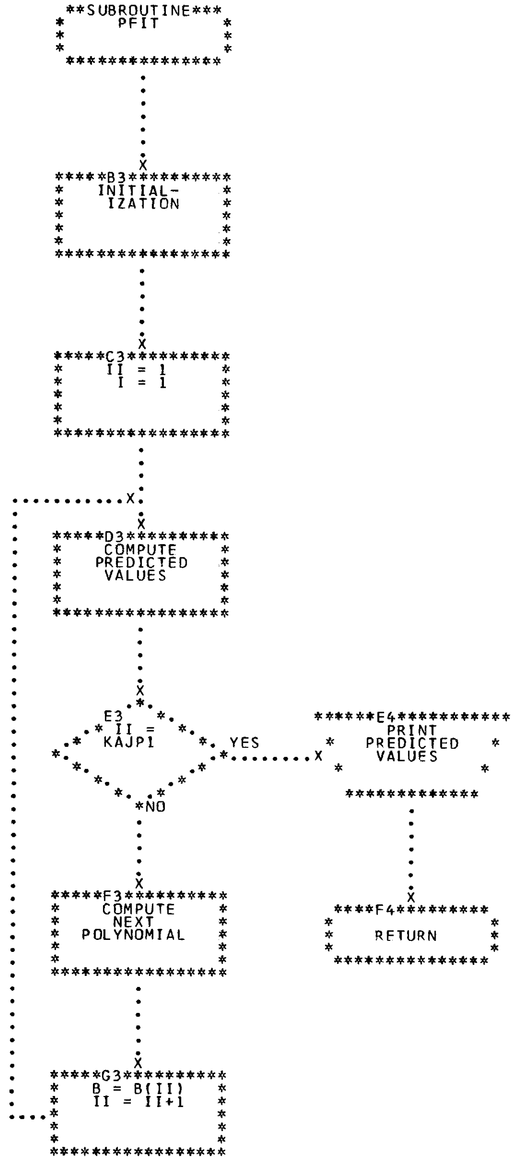

A3
SUBROUTINE PRNT
B3
COMPARE MX(MID) TO 1
LT
GT
EQ
B2
RETURN
B4
PUNCH MATRIX
C3
PRINT TITLE LINE
C4
IS MX(MID)=2
YES
NO
C5
RETURN
D3
PRINT MATRIX NAME
D4
MID=1,2,..6
E3
PRINT COLUMN HEAD
F3
PRINT MATRIX
G3
RETURN

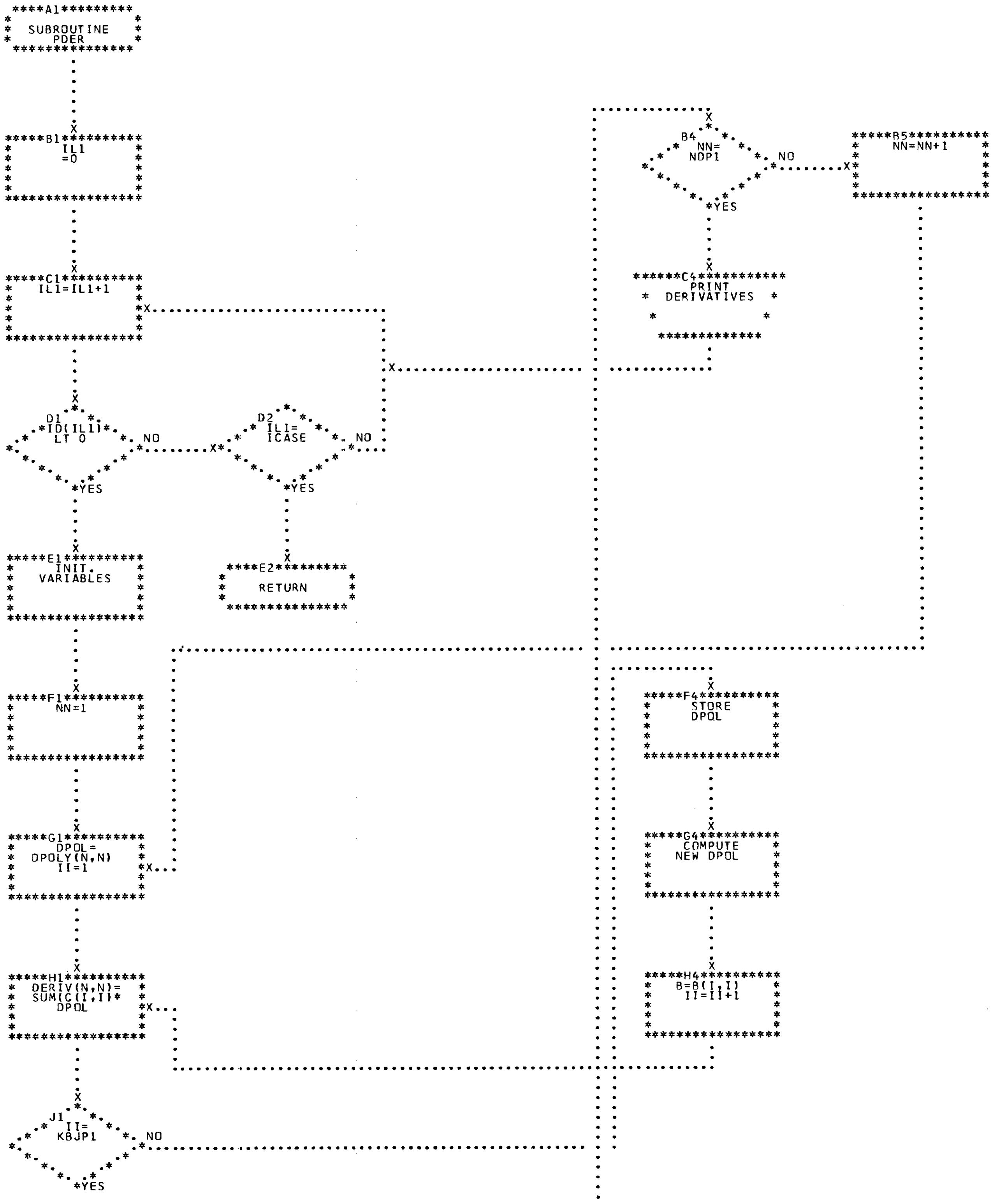

A2
PROGRAM POLY
B2
READ I/O UNITS CARD
C2
READ JOB TITLE
D2
READ OPTIONS
E2
PRINT TITLE AND OPTIONS
F2
READ FORMAT
G1
READ DATA FROM DISK
G2
COMPARE INMD TO 2
EQ
GT
LT
G3
READ ALPHA, BETA,C FROM CARDS
H2
READ SOURCES DATA FROM CARDS
J2
CALL LINK POL 2

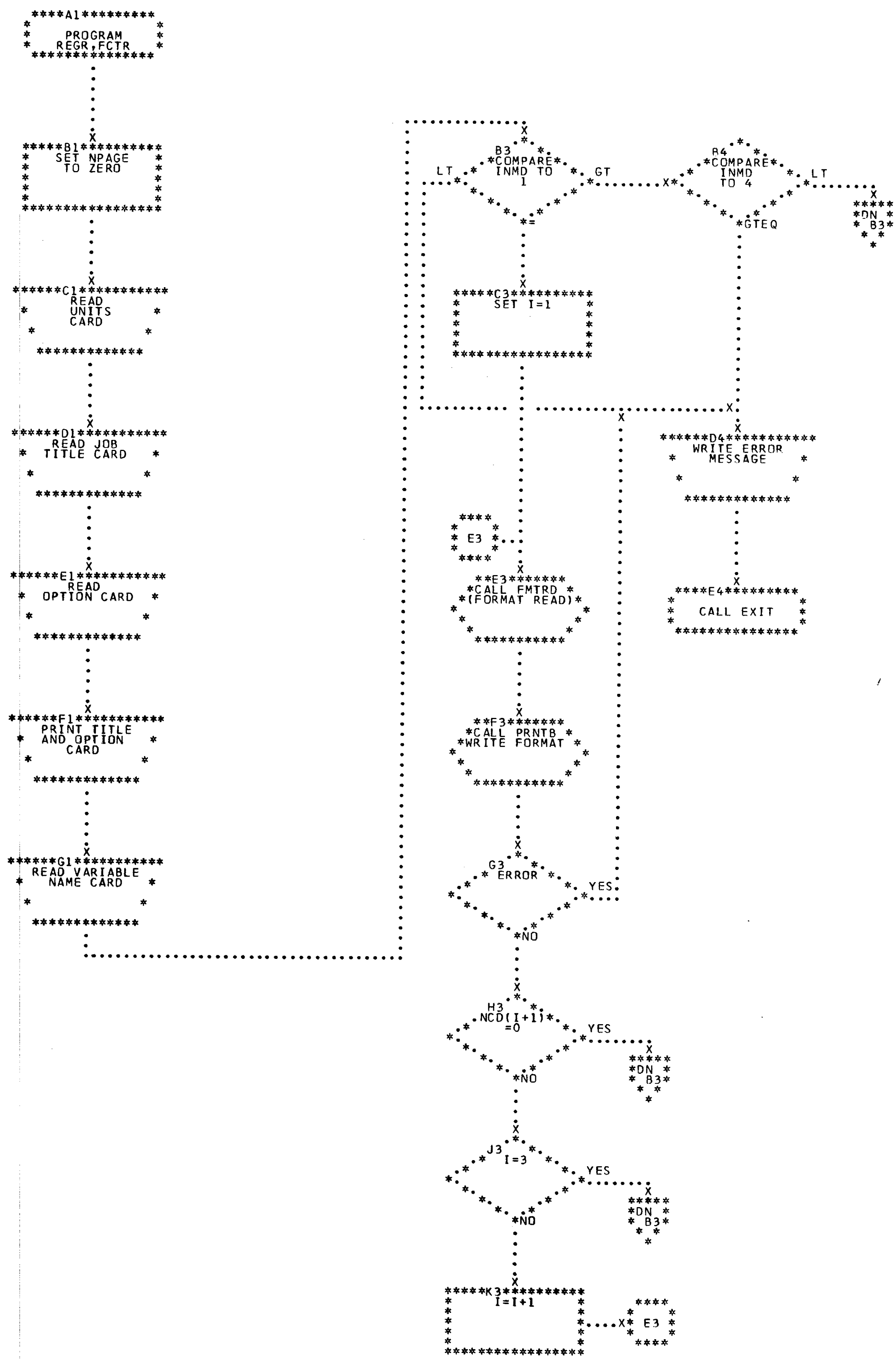

A3
PROGRAM POL2
B3
INM2 GT 0
NO
B4
CALL POLSQ
YES
C3
I COF GT 0
YES
C4
CALL PCOEF
NO
D3
IDER GT 0
YES
D4
CALL PDER
NO
E3
ISCR GT 0
YES
E4
CALL PFIT
NO
F3
CALL EXIT

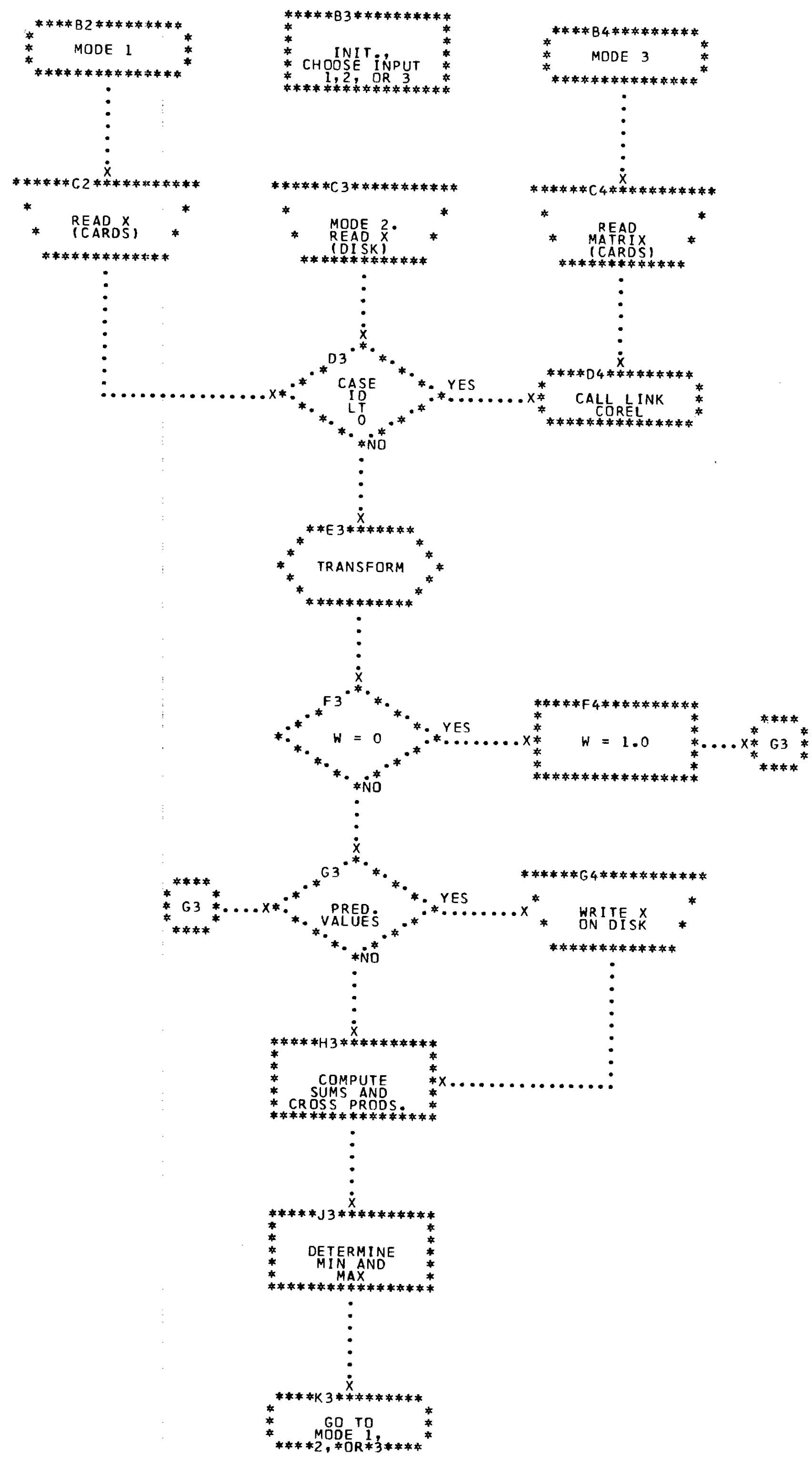

A2
SUBROUTINE POLSQ
B2
INIT. VARIABLES
C2
COMPUTE SOL.VCTR. C
C2
D2
COMPUTE PRED.VALS. YA
E2
COMPUTE VAR. CRITERION VAR2
F2
VAR2 LT EPS
YES
NO
G2
MAX DEG. GT N
YES
NO
G3
PRINT SOLUTION SET
H2
COMPUTE SOL. VCTR. ALPHA
H3
PUNCH ALPHA, BETA,C IF REQUIRED
J2
COMPUTE NEXT ORDER ORTH.POLY
J3
RETURN
B4
COMPUTE SOL.VCTR. BETA
C4
DEG. =4
NO
YES
C5
MOVE VAR1 TO VAR2
D4
PRINT SOL. SET
D5
INCREASE DEG PARAM
C2

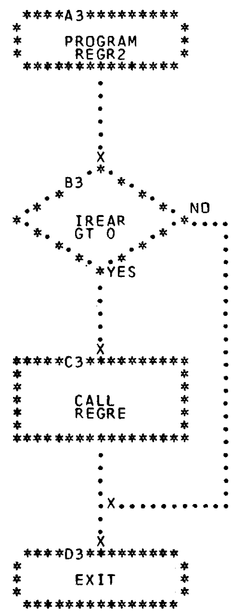

A3
SUBROUTINE PCOEF
B3
INITIALIZE VARIABLES KKD = DEG+1
C3
II=1 (DEG. OF ORTH. POLY.)
D3
II = II+1
E3
COMPUTE ORTHOGONAL POLYNOMIAL II
F3
COMPUTE COEFFICIENT
G3
IS II=KKD
NO
YES
G2
RESET VECTORS FOR NEXT COEFFICIENT
H3
PRINT COEFFI- CIENTS
J3
RETURN

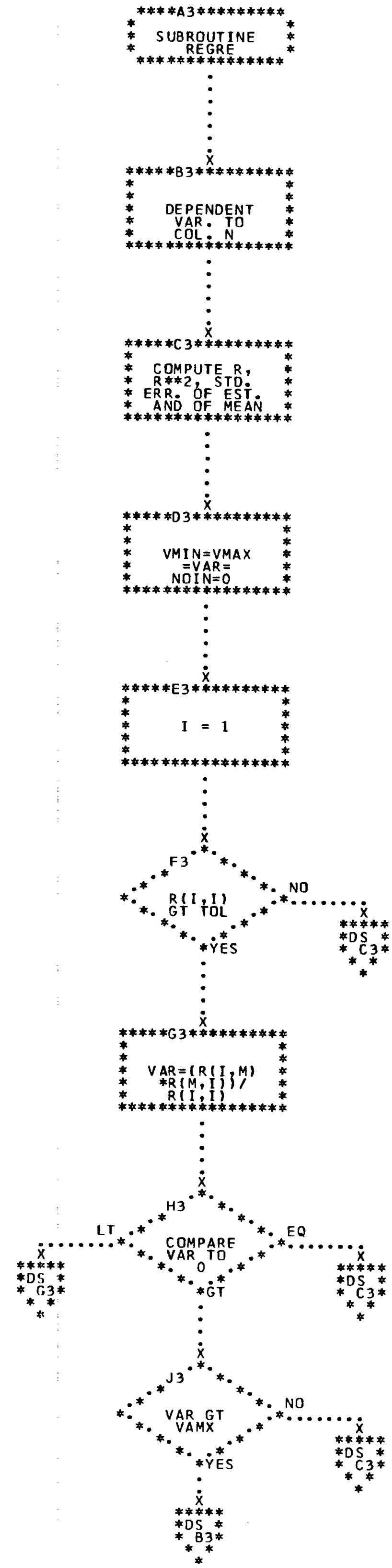

SUBROUTINE PFIT
B3 INITIAL- IZATION
C3 II = 1 I = 1
D3 COMPUTE PREDICTED VALUES
E3 II = KAJPI
YES
NO
E4 PRINT PREDICTED VALUES
F3 COMPUTE NEXT POLYNOMIAL
F4 RETURN
G3 B = B(II) II = II+1

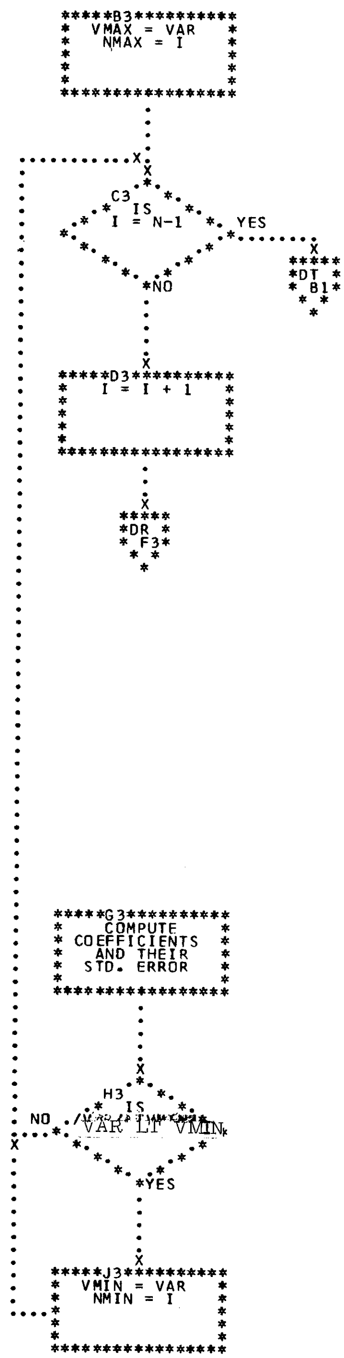

A1
SUBROUTINE PDER
B1
IL1 =0
C1
IL1=IL1+1
D1
ID(IL1) LT 0
NO
YES
D2
IL1= ICASE
NO
YES
E1
INIT. VARIABLES
E2
RETURN
F1
NN=1
G1
DPOL= DPOLY(N,N) II=1
H1
DERIV(N,N)= SUM(C(I,I)* DPOL
J1
II= KBJP1
NO
YES
B4
NN= NDP1
NO
YES
B5
NN=NN+1
C4
PRINT DERIVATIVES
F4
STORE DPOL
G4
COMPUTE NEW DPOL
H4
B=B(I,I) II=II+1

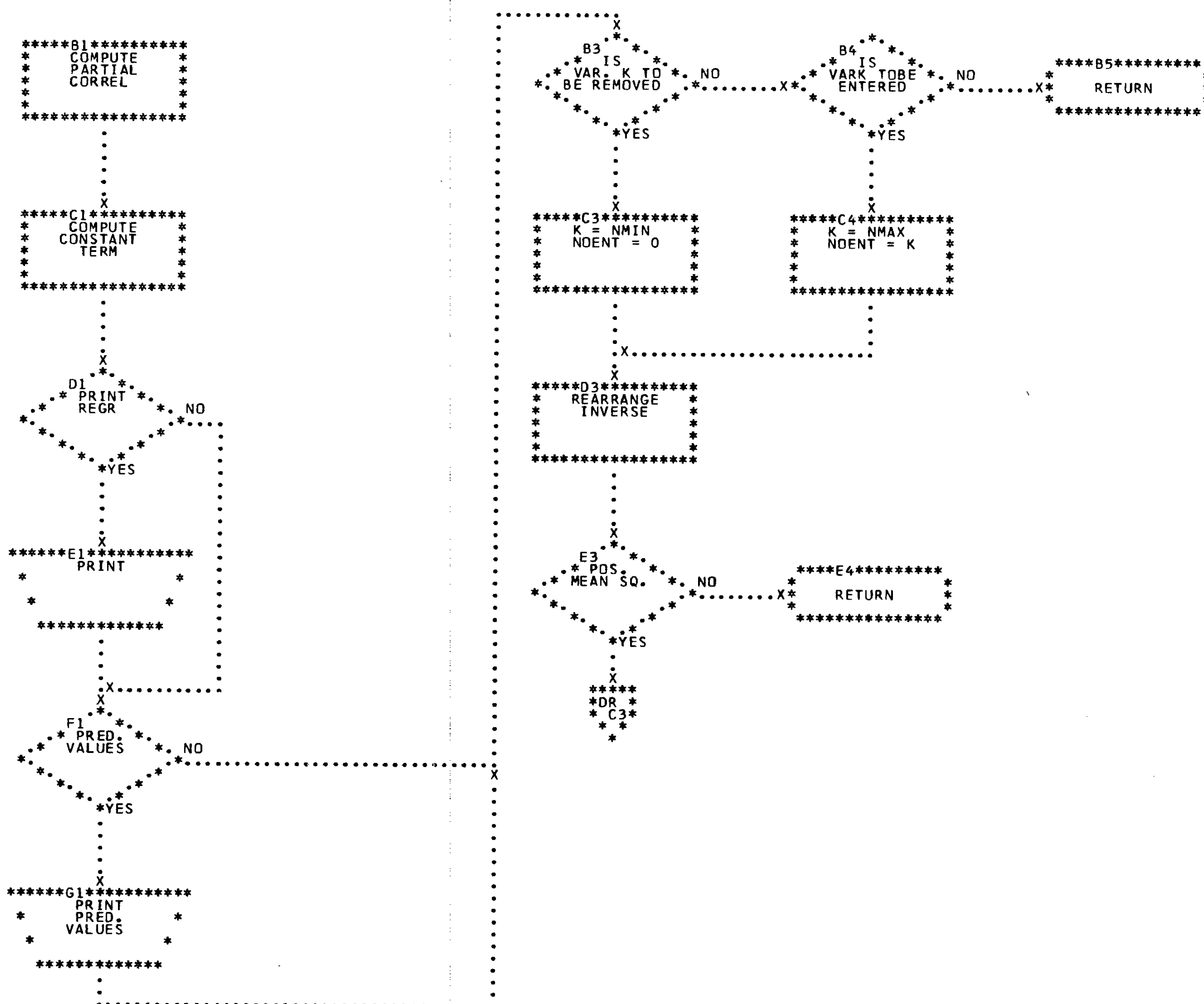

A1
PROGRAM REGR,FCTR
B1
SET NPAGE TO ZERO
C1
READ UNITS CARD
D1
READ JOB TITLE CARD
E1
READ OPTION CARD
F1
PRINT TITLE AND OPTION CARD
G1
READ VARIABLE NAME CARD
B3
COMPARE INMD TO 1
LT
GT
=
B4
COMPARE INMD TO 4
LT
GTEQ
DN B3
C3
SET I=1
D4
WRITE ERROR MESSAGE
E4
CALL EXIT
E3
E3
CALL FMTRD (FORMAT READ)
F3
CALL PRNTB WRITE FORMAT
G3
ERROR
YES
NO
H3
NCD(I+1) =0
YES
NO
DN B3
J3
I=3
YES
NO
DN B3
K3
I=I+1
E3

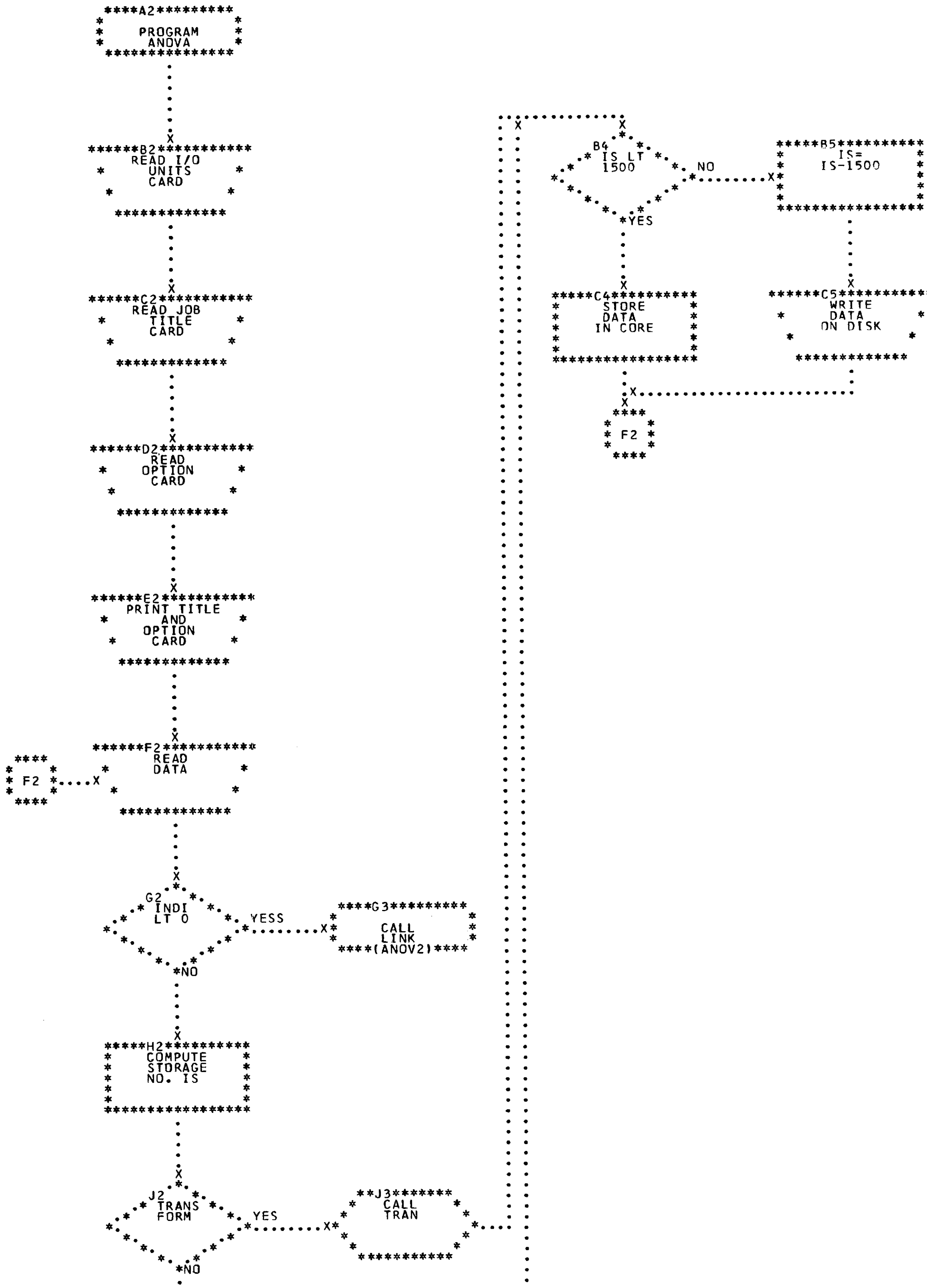

B2
MODE 1
B3
INIT., CHOOSE INPUT 1,2, OR 3
B4
MODE 3
C2
READ X (CARDS)
C3
MODE 2. READ X (DISK)
C4
READ MATRIX (CARDS)
D3
CASE ID LT 0
YES
NO
D4
CALL LINK COREL
E3
TRANSFORM
F3
W = 0
YES
NO
F4
W = 1.0
G3
G3
PRED. VALUES
YES
NO
G3
G4
WRITE X ON DISK
H3
COMPUTE SUMS AND CROSS PRODS.
J3
DETERMINE MIN AND MAX
K3
GO TO MODE 1, 2, OR 3

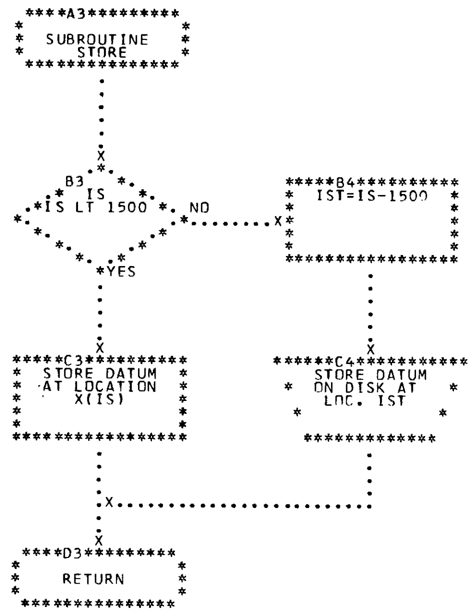

A3
PROGRAM
REGR2
B3
IREAR
GT 0
NO
YES
C3
CALL
REGRE
D3
EXIT

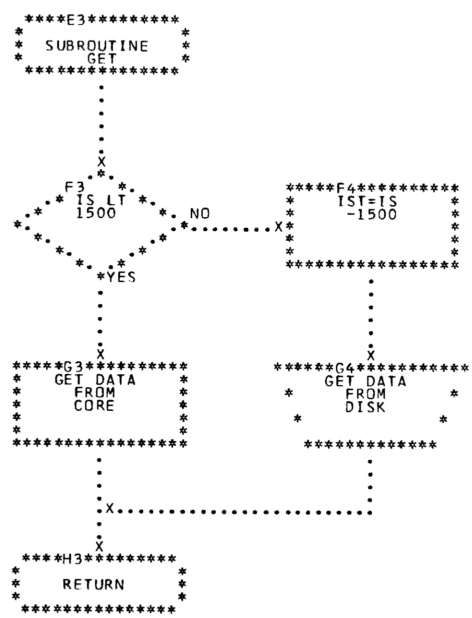

A3
SUBROUTINE REGRE
B3
DEPENDENT VAR. TO COL. N
C3
COMPUTE R, R**2, STD. ERR. OF EST. AND OF MEAN
D3
VMIN=VMAX =VAR= NOIN=0
E3
I = 1
F3
R(I,I) GT TOL
NO
DS C3
YES
G3
VAR=(R(I,M) *R(M,I))/ R(I,I)
H3
COMPARE VAR TO 0
LT
DS G3
EQ
DS C3
GT
J3
VAR GT VAMX
NO
DS C3
YES
DS B3


B3
VMAX = VAR
NMAX = I
C3
IS
I = N-1
YES
NO
DT
B1
D3
I = I + 1
DR
F3
G3
COMPUTE
COEFFICIENTS
AND THEIR
STD. ERROR
H3
IS
VAR LT VMIN
NO
YES
J3
VMIN = VAR
NMIN = I


B1
COMPUTE PARTIAL CORREL
C1
COMPUTE CONSTANT TERM
D1
PRINT REGR
NO
YES
E1
PRINT
F1
PRED. VALUES
NO
YES
G1
PRINT PRED. VALUES
B3
IS VAR. K TO BE REMOVED
NO
YES
B4
IS VARK TOBE ENTERED
NO
YES
B5
RETURN
C3
K = NMIN
NOENT = 0
C4
K = NMAX
NOENT = K
D3
REARRANGE INVERSE
E3
POS. MEAN SQ.
NO
YES
E4
RETURN
DR
C3


A2
PROGRAM ANOVA
B2
READ I/O UNITS CARD
C2
READ JOB TITLE CARD
D2
READ OPTION CARD
E2
PRINT TITLE AND OPTION CARD
F2
READ DATA
F2
G2
INDI LT 0
YESS
NO
G3
CALL LINK (ANOV2)
H2
COMPUTE STORAGE NO. IS
J2
TRANS FORM
YES
NO
J3
CALL TRAN
B4
IS LT 1500
NO
YES
B5
IS= IS-1500
C4
STORE DATA IN CORE
C5
WRITE DATA ON DISK
F2


A3
SUBROUTINE STORE
B3
IS LT 1500
NO
B4
IST=IS-1500
YES
C3
STORE DATUM AT LOCATION X(IS)
C4
STORE DATUM ON DISK AT LOC. IST
D3
RETURN


E3
SUBROUTINE GET
F3
IS LT 1500
NO
F4
IST=IS -1500
YES
G3
GET DATA FROM CORE
G4
GET DATA FROM DISK
H3
RETURN

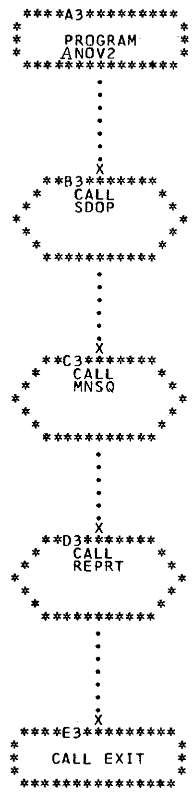
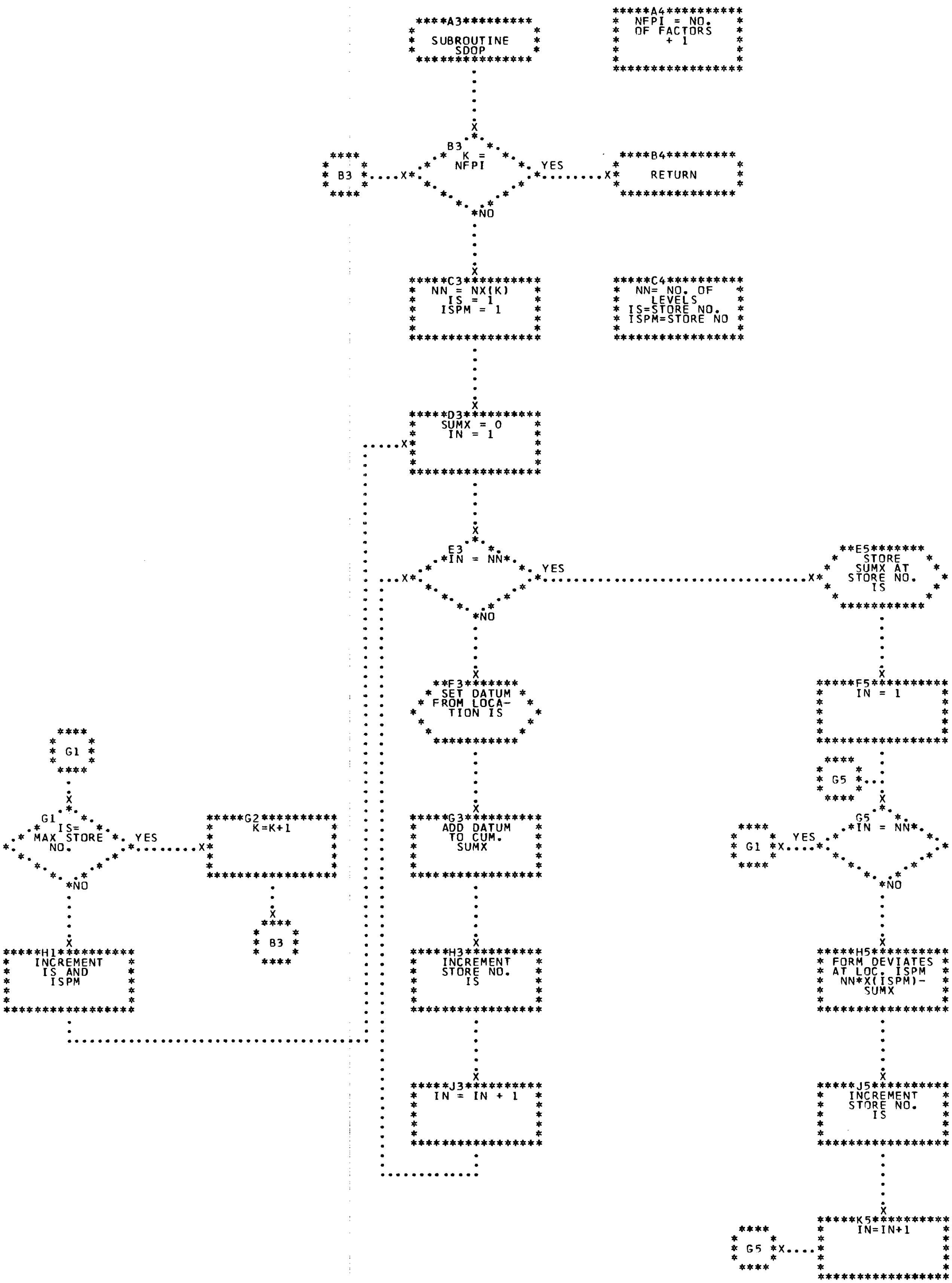
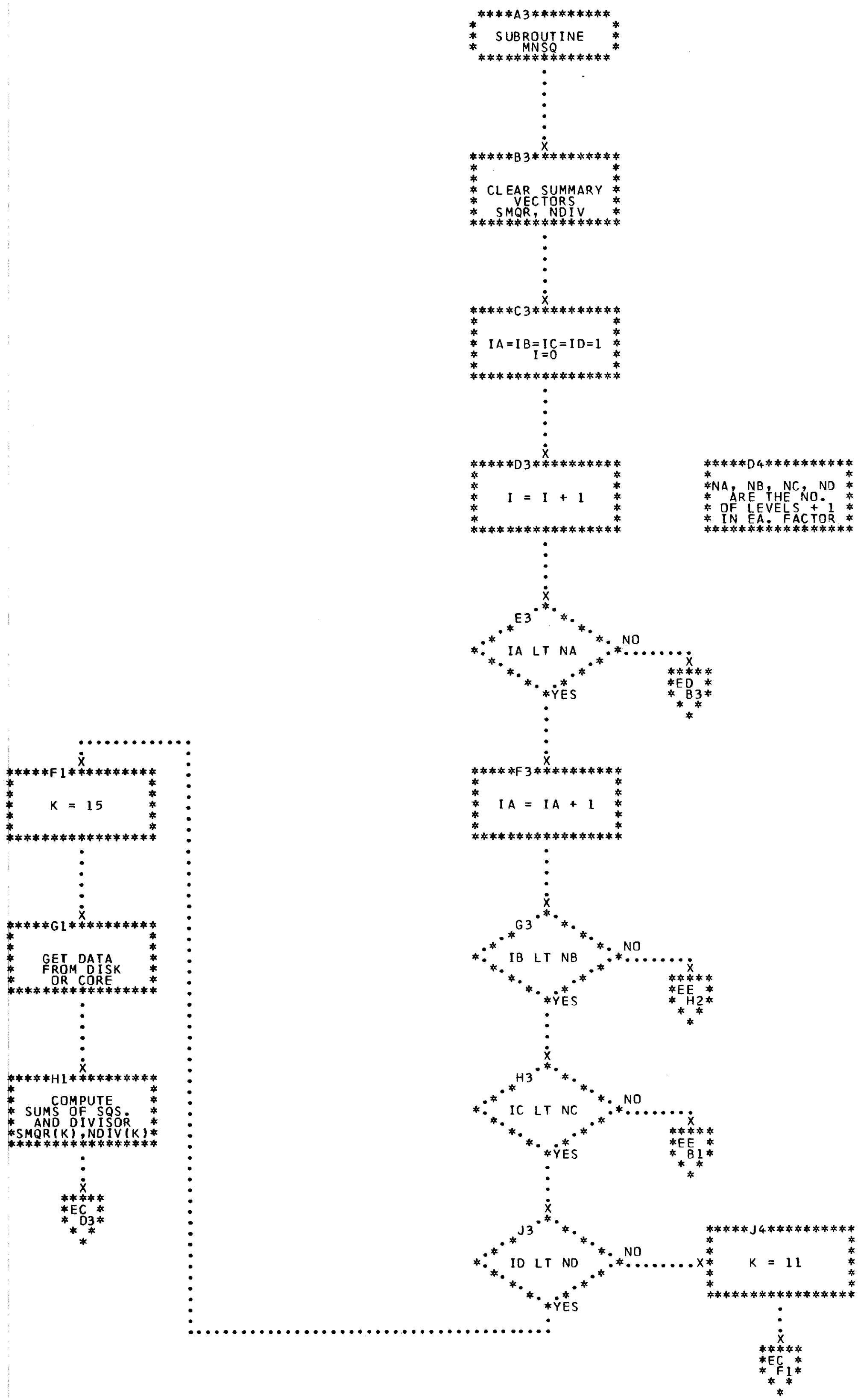
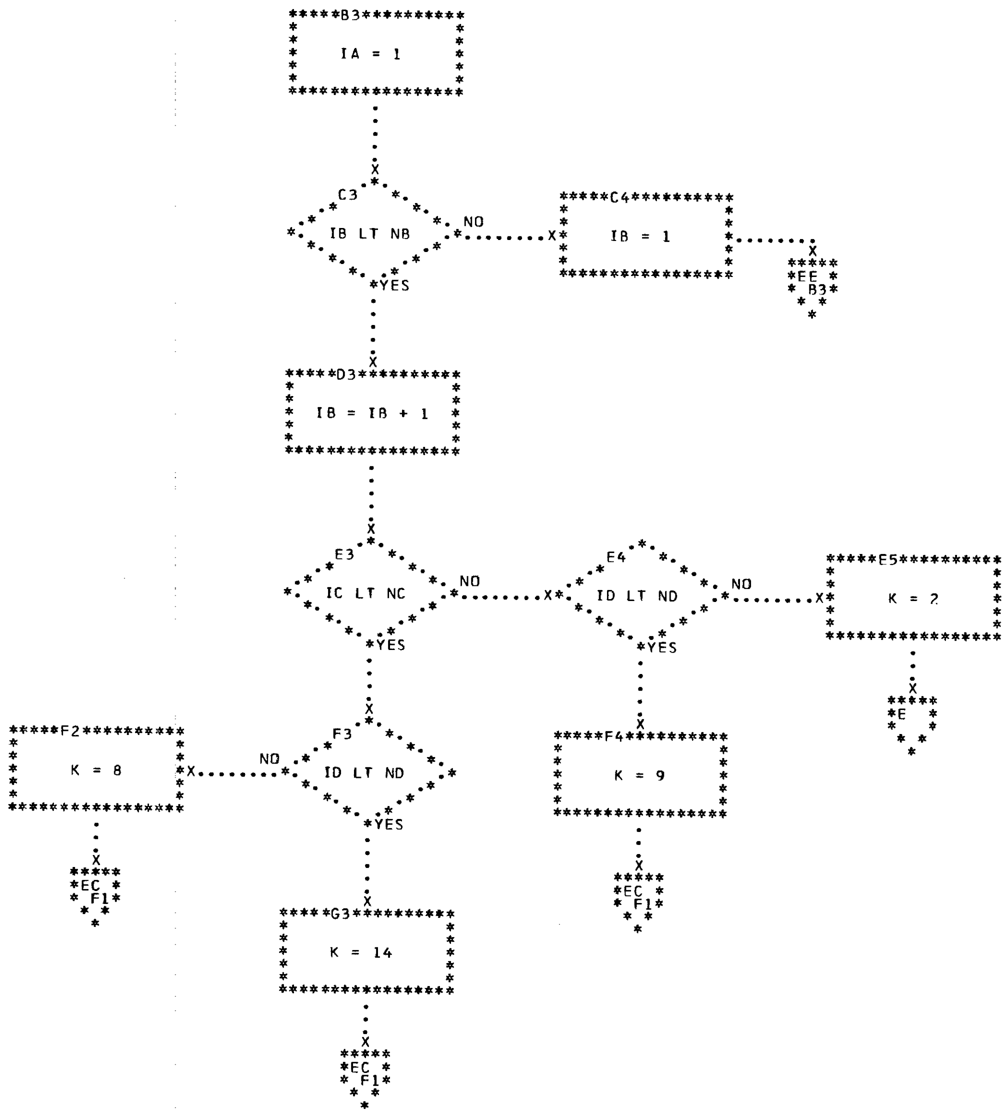
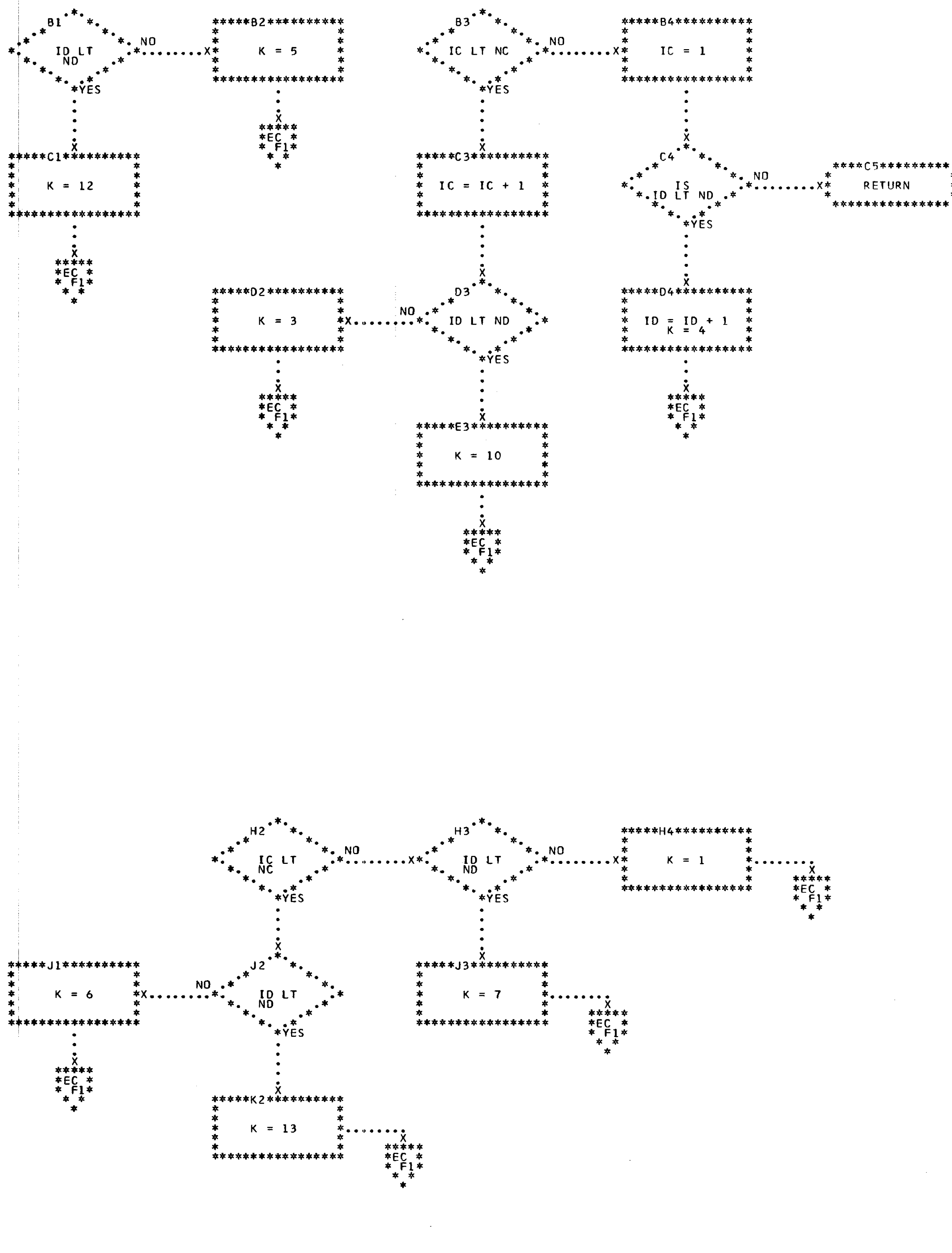

A3
PROGRAM ANOV2
B3
CALL SDOP
C3
CALL MNSQ
D3
CALL REPRT
E3
CALL EXIT

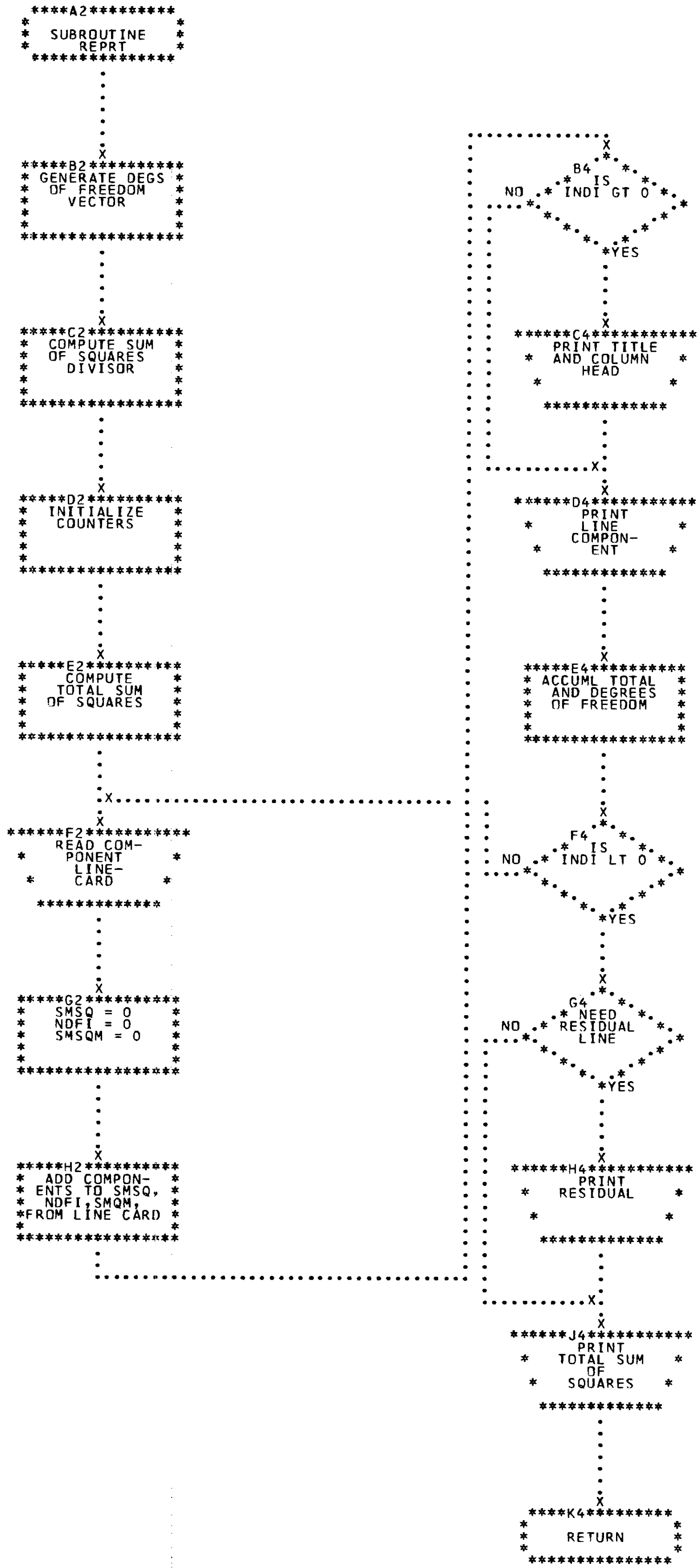
CHART EB

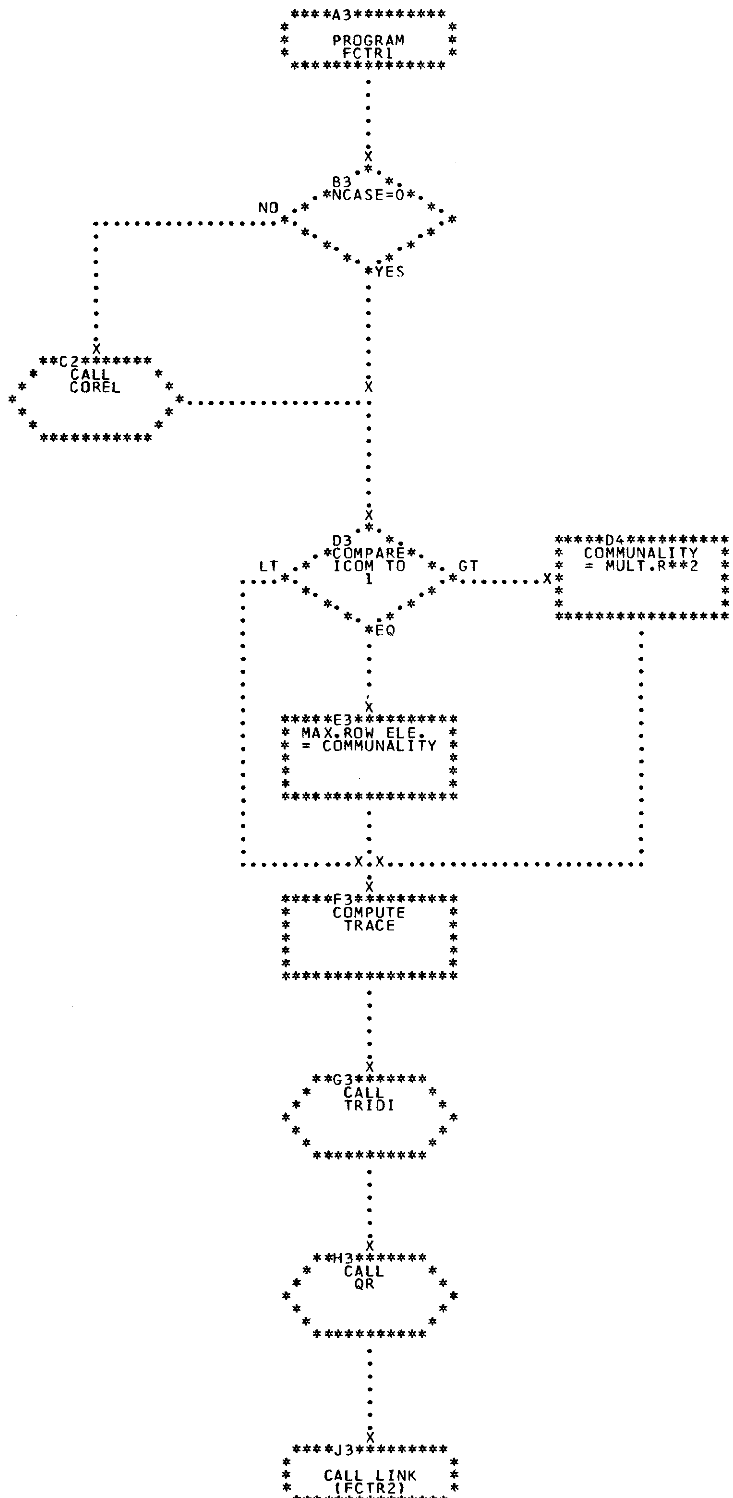

A3 SUBROUTINE SDOP
A4 NFPI = NO. OF FACTORS + 1
B3
B3 K = NFPI
YES
NO
B4 RETURN
C3 NN = NX(K) IS = 1 ISPM = 1
C4 NN= NO. OF LEVELS IS=STORE NO. ISPM=STORE NO
D3 SUMX = 0 IN = 1
E3 IN = NN
YES
NO
E5 STORE SUMX AT STORE NO. IS
F3 SET DATUM FROM LOCA- TION IS
F5 IN = 1
G1
G1 IS= MAX STORE NO.
YES
NO
G2 K=K+1
B3
G3 ADD DATUM TO CUM. SUMX
G5
G5 IN = NN
G1
YES
NO
H1 INCREMENT IS AND ISPM
H3 INCREMENT STORE NO. IS
H5 FORM DEVIATES AT LOC. ISPM NN*X(ISPM)- SUMX
J3 IN = IN + 1
J5 INCREMENT STORE NO. IS
K5 IN=IN+1
G5


A3
SUBROUTINE MNSQ
B3
CLEAR SUMMARY VECTORS SMQR, NDIV
C3
IA=IB=IC=ID=1
I=0
D3
I = I + 1
D4
NA, NB, NC, ND ARE THE NO. OF LEVELS + 1 IN EA. FACTOR
E3
IA LT NA
NO
ED B3
YES
F3
IA = IA + 1
G3
IB LT NB
NO
EE H2
YES
H3
IC LT NC
NO
EE B1
YES
J3
ID LT ND
NO
J4
K = 11
EC F1
YES
F1
K = 15
G1
GET DATA FROM DISK OR CORE
H1
COMPUTE SUMS OF SQS. AND DIVISOR SMQR(K),NDIV(K)
EC D3


B3
IA = 1
C3
IB LT NB
NO
YES
C4
IB = 1
EE
B3
D3
IB = IB + 1
E3
IC LT NC
NO
YES
E4
ID LT ND
NO
YES
E5
K = 2
E
F2
K = 8
EC
F1
F3
ID LT ND
NO
YES
F4
K = 9
EC
F1
G3
K = 14
EC
F1


B1
ID LT ND
NO
YES
B2
K = 5
B3
IC LT NC
NO
YES
B4
IC = 1
C1
K = 12
C3
IC = IC + 1
C4
IS ID LT ND
NO
YES
C5
RETURN
D2
K = 3
D3
ID LT ND
NO
YES
D4
ID = ID + 1
K = 4
E3
K = 10
EC F1
H2
IC LT NC
NO
YES
H3
ID LT ND
NO
YES
H4
K = 1
J1
K = 6
J2
ID LT ND
NO
YES
J3
K = 7
K2
K = 13


A2
SUBROUTINE REPRT
B2
GENERATE DEGS OF FREEDOM VECTOR
C2
COMPUTE SUM OF SQUARES DIVISOR
D2
INITIALIZE COUNTERS
E2
COMPUTE TOTAL SUM OF SQUARES
F2
READ COMPONENT LINE-CARD
G2
SMSQ = 0
NDFI = 0
SMSQM = 0
H2
ADD COMPONENTS TO SMSQ, NDFI,SMQM, FROM LINE CARD
B4
IS INDI GT 0
NO
YES
C4
PRINT TITLE AND COLUMN HEAD
D4
PRINT LINE COMPONENT
E4
ACCUML TOTAL AND DEGREES OF FREEDOM
F4
IS INDI LT 0
NO
YES
G4
NEED RESIDUAL LINE
NO
YES
H4
PRINT RESIDUAL
J4
PRINT TOTAL SUM OF SQUARES
K4
RETURN


A3
PROGRAM FCTR1
B3
NCASE=0
NO
YES
C2
CALL COREL
D3
COMPARE ICOM TO 1
LT
GT
EQ
D4
COMMUNALITY = MULT.R**2
E3
MAX.ROW ELE. = COMMUNALITY
F3
COMPUTE TRACE
G3
CALL TRIDI
H3
CALL QR
J3
CALL LINK (FCTR2)

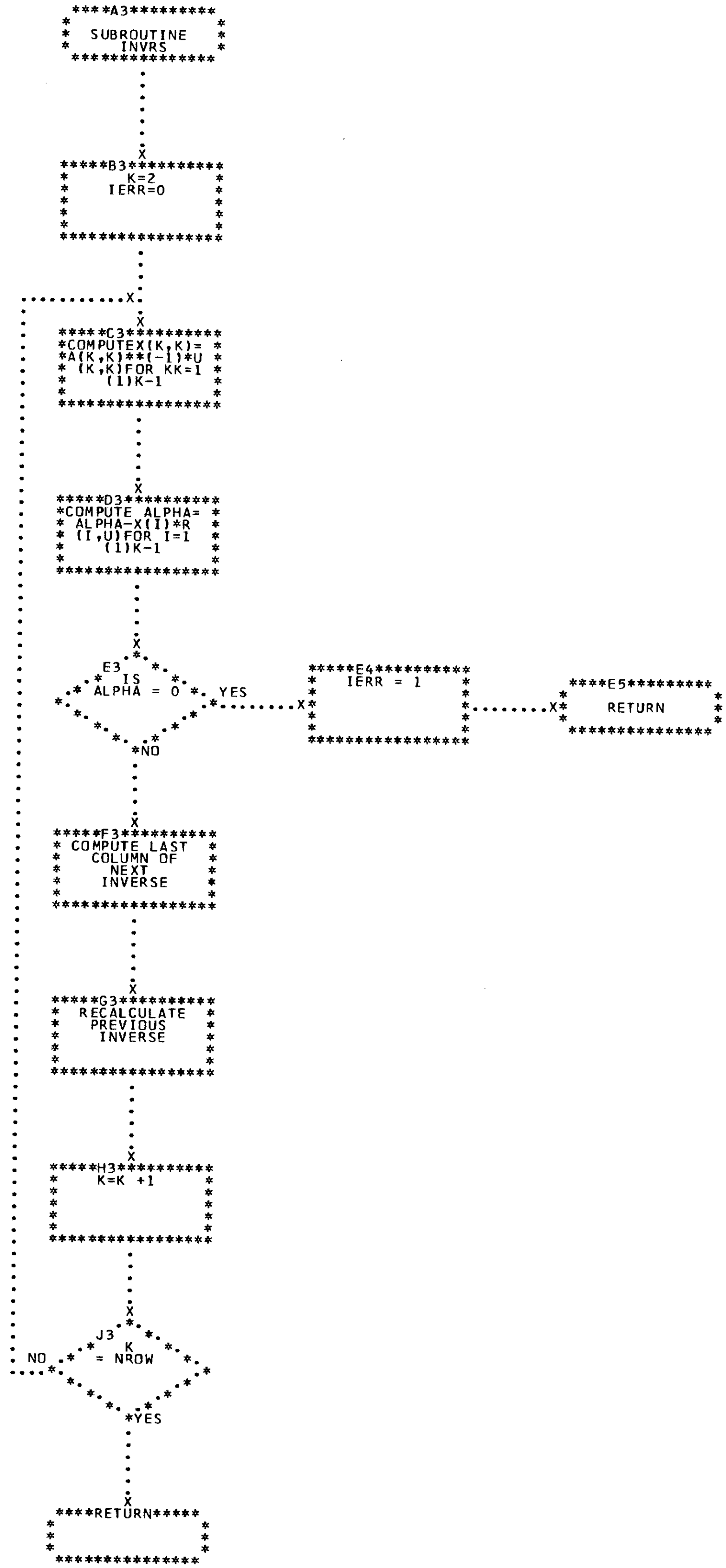
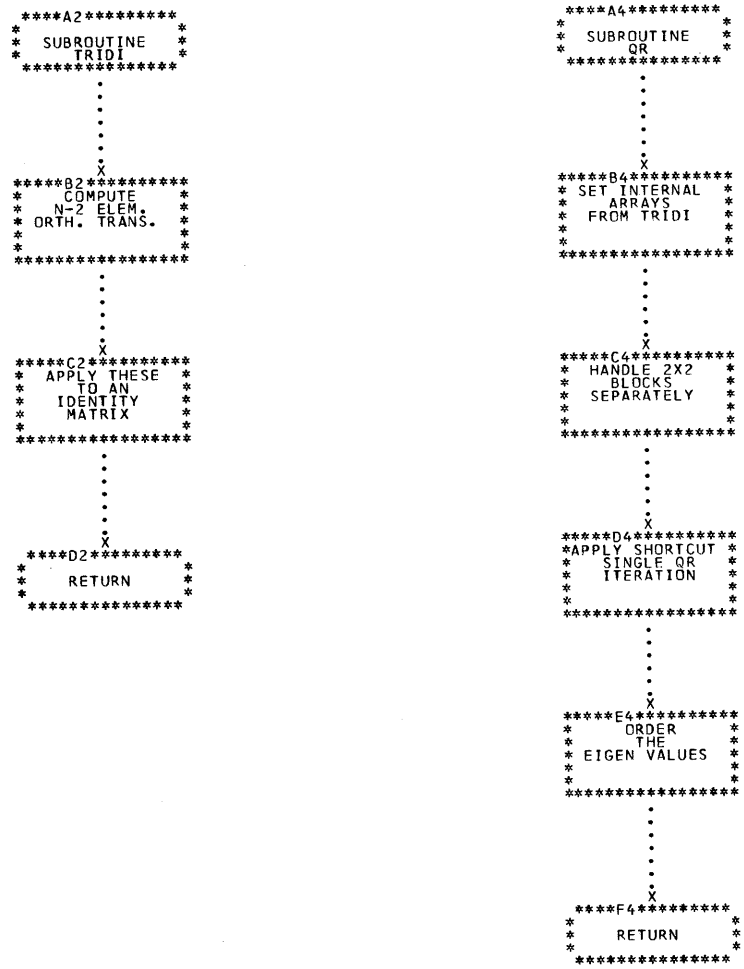
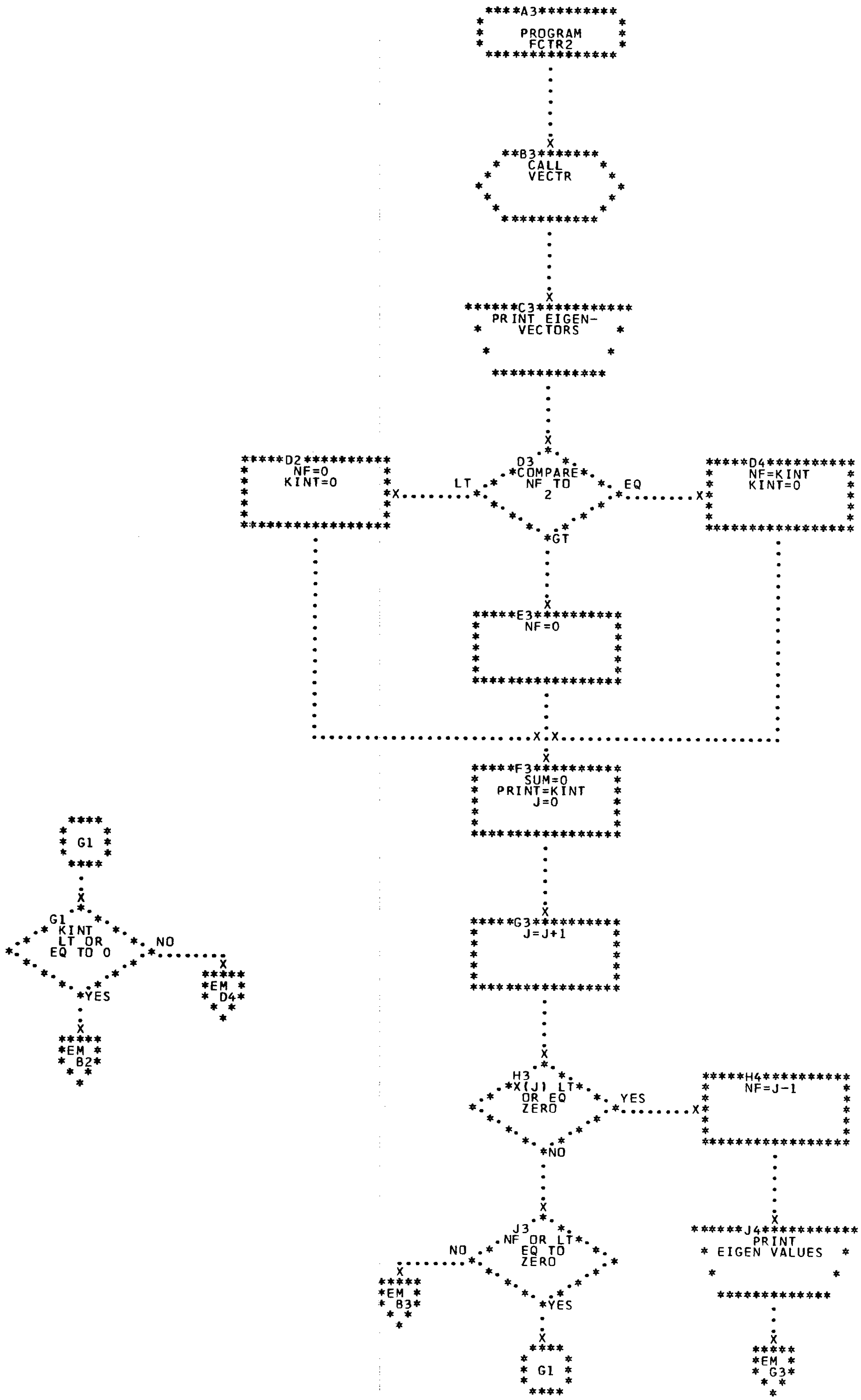

A3
SUBROUTINE INVRS
B3
K=2
IERR=0
C3
COMPUTEX(K,K)= A(K,K)**(-1)*U (K,K)FOR KK=1 (1)K-1
D3
COMPUTE ALPHA= ALPHA-X(I)*R (I,U)FOR I=1 (1)K-1
E3
IS ALPHA = 0
YES
NO
E4
IERR = 1
E5
RETURN
F3
COMPUTE LAST COLUMN OF NEXT INVERSE
G3
RECALCULATE PREVIOUS INVERSE
H3
K=K +1
J3
K = NROW
NO
YES
RETURN

CHART EJ


A2
SUBROUTINE TRIDI
B2
COMPUTE N-2 ELEM. ORTH. TRANS.
C2
APPLY THESE TO AN IDENTITY MATRIX
D2
RETURN
A4
SUBROUTINE QR
B4
SET INTERNAL ARRAYS FROM TRIDI
C4
HANDLE 2X2 BLOCKS SEPARATELY
D4
APPLY SHORTCUT SINGLE QR ITERATION
E4
ORDER THE EIGEN VALUES
F4
RETURN


A3
PROGRAM FCTR2
B3
CALL VECTR
C3
PRINT EIGEN-VECTORS
D2
NF=0
KINT=0
D3
COMPARE NF TO 2
LT
EQ
GT
D4
NF=KINT
KINT=0
E3
NF=0
F3
SUM=0
PRINT=KINT
J=0
G3
J=J+1
H3
X(J) LT OR EQ ZERO
YES
NO
H4
NF=J-1
J3
NF OR LT EQ TO ZERO
NO
YES
EM B3
G1
J4
PRINT EIGEN VALUES
EM G3
G1
G1
KINT LT OR EQ TO 0
NO
YES
EM D4
EM B2

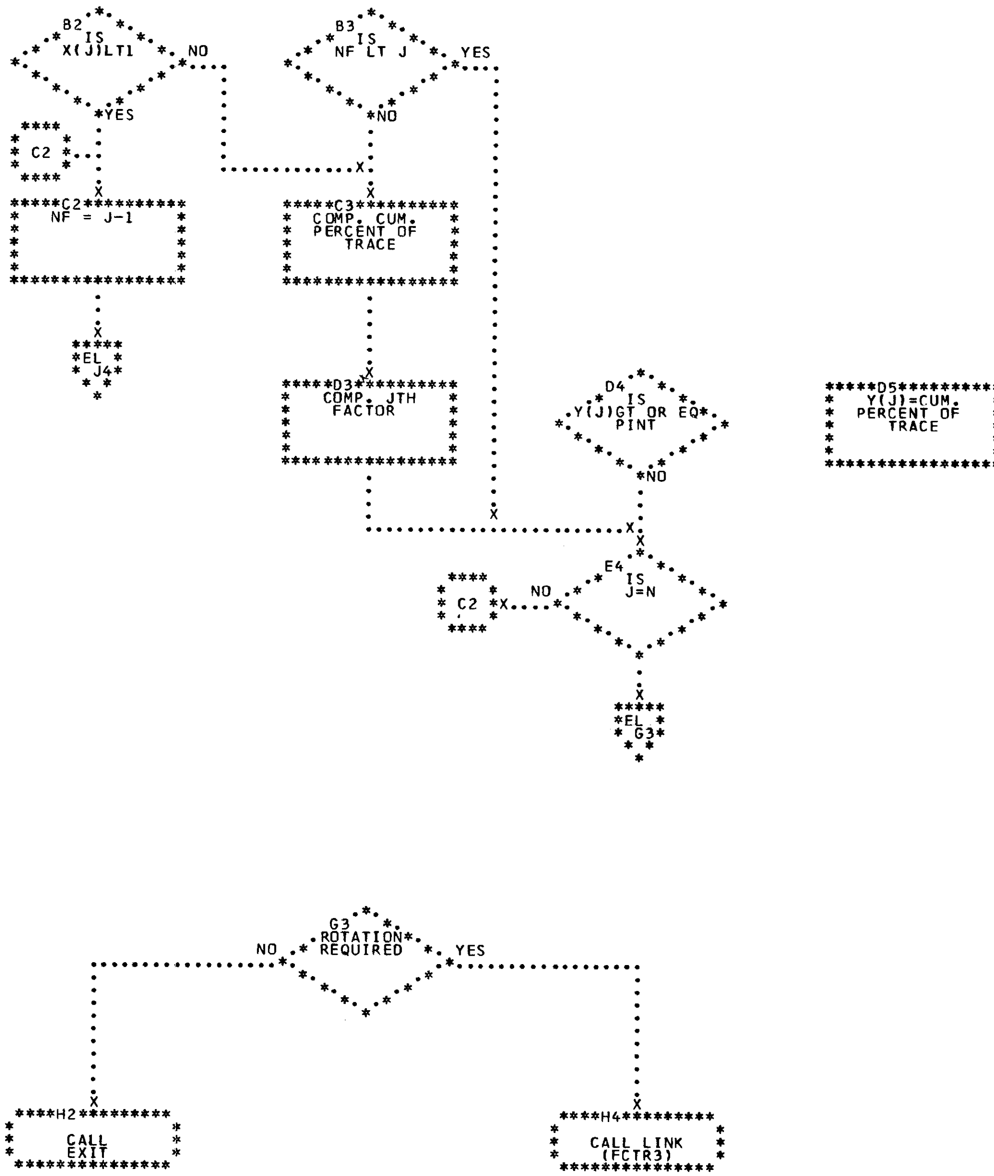
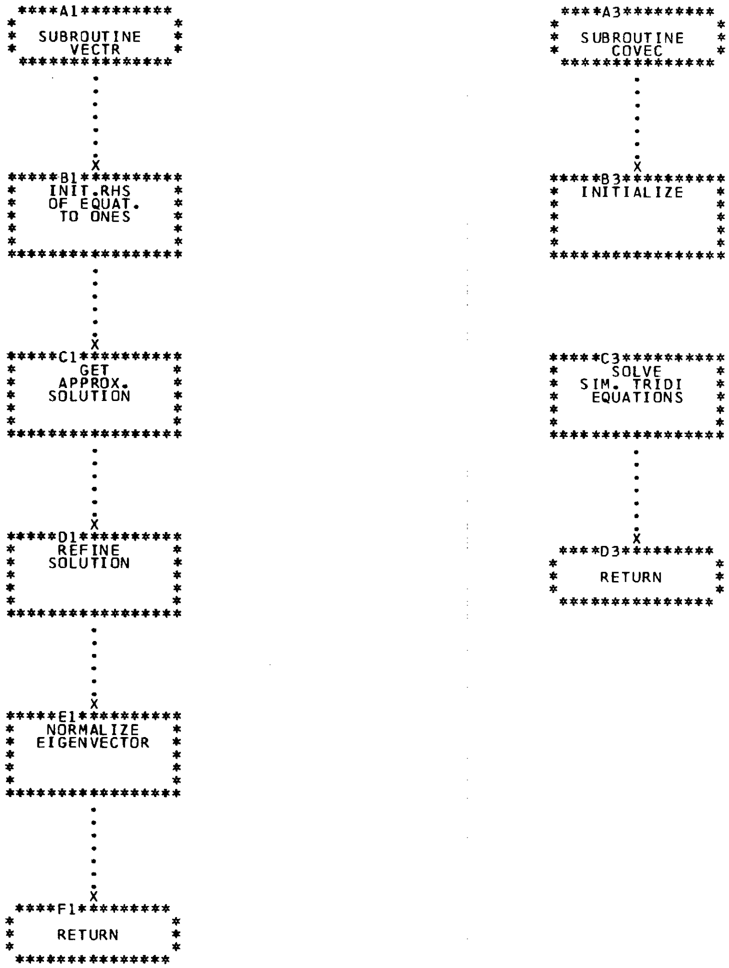
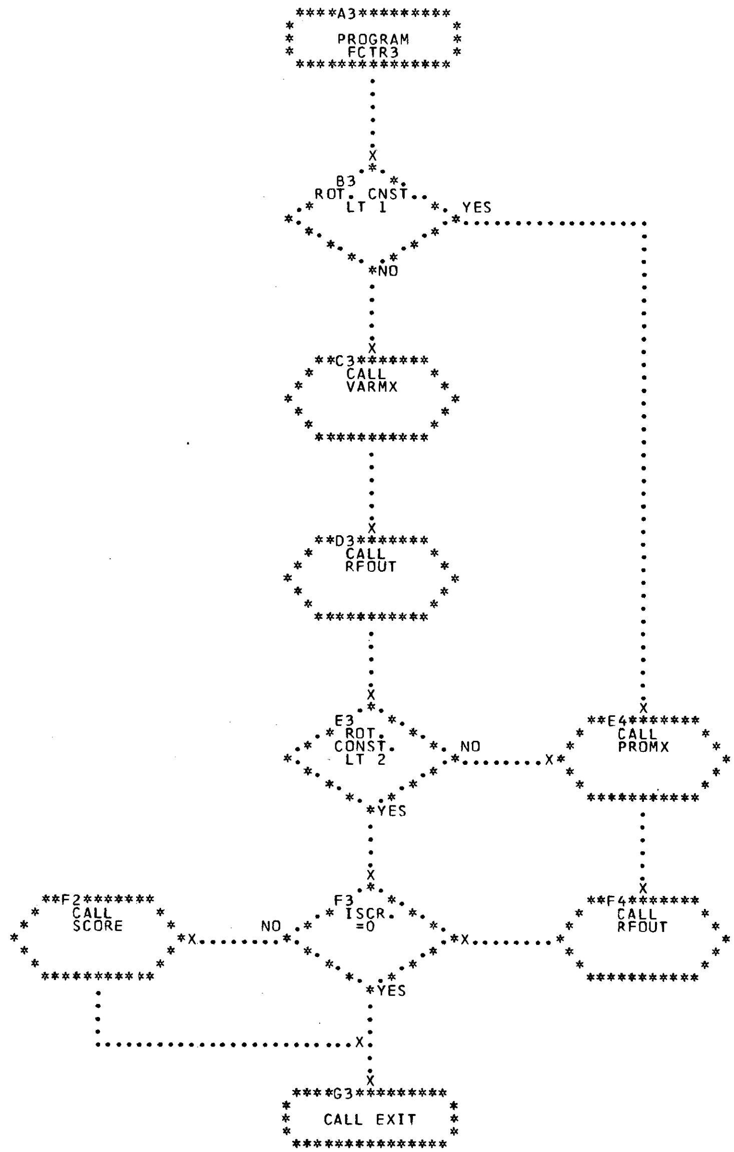
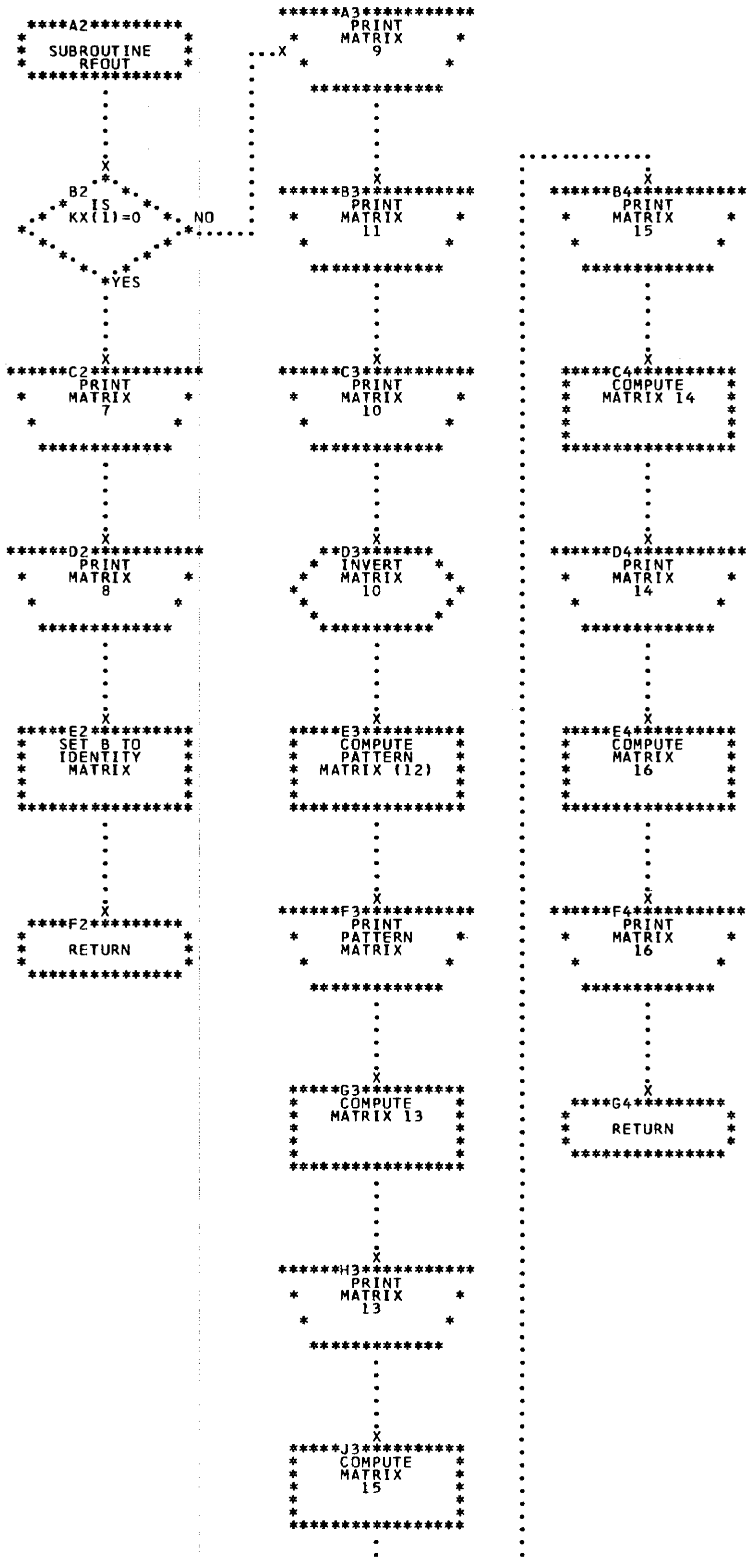
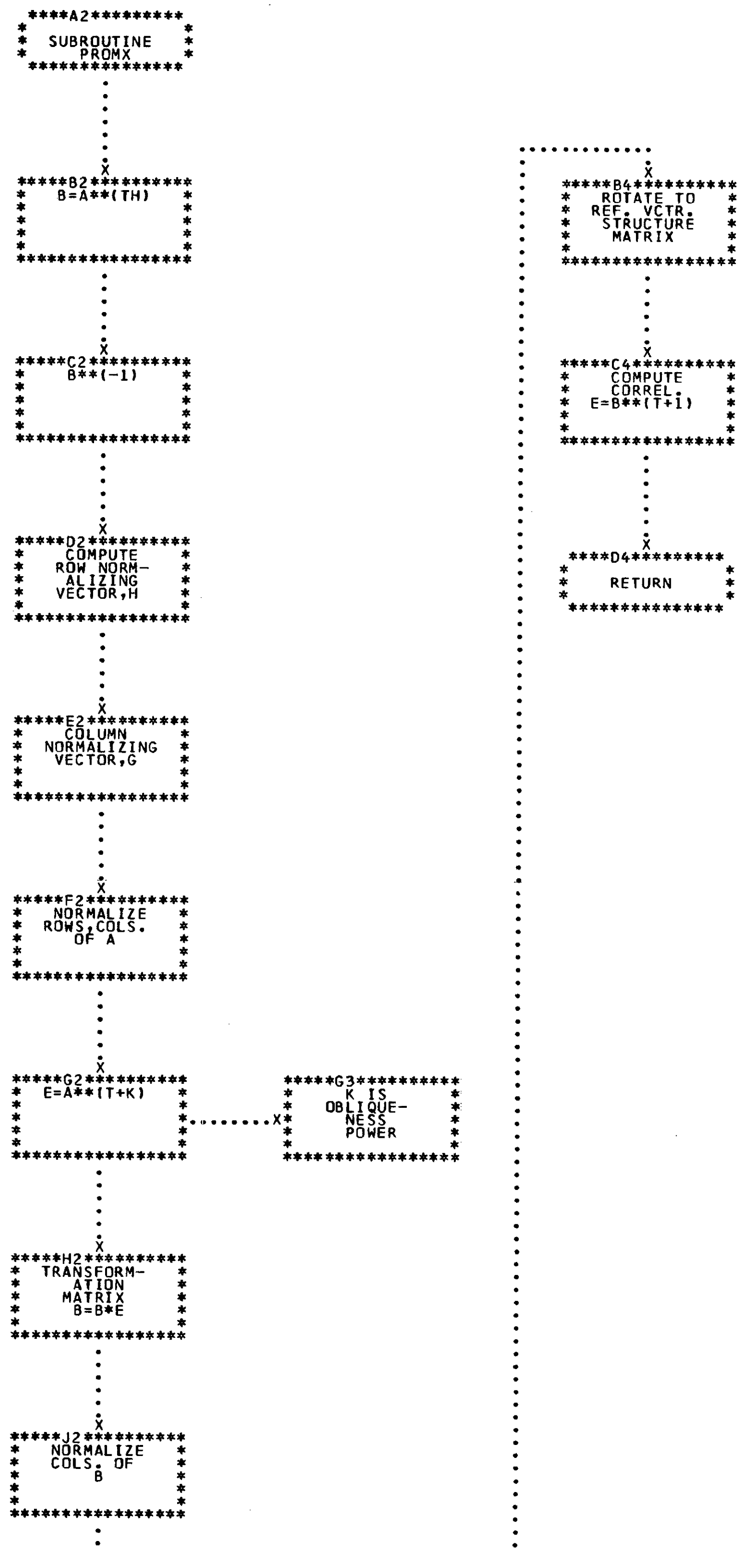

B2
IS
X(J)LT1
NO
YES
C2
C2
NF = J-1
EL
J4
B3
IS
NF LT J
YES
NO
C3
COMP. CUM.
PERCENT OF
TRACE
D3
COMP. JTH
FACTOR
D4
IS
Y(J)GT OR EQ
PINT
NO
D5
Y(J)=CUM.
PERCENT OF
TRACE
E4
IS
J=N
NO
C2
EL
G3
G3
ROTATION
REQUIRED
NO
YES
H2
CALL
EXIT
H4
CALL LINK
(FCTR3)

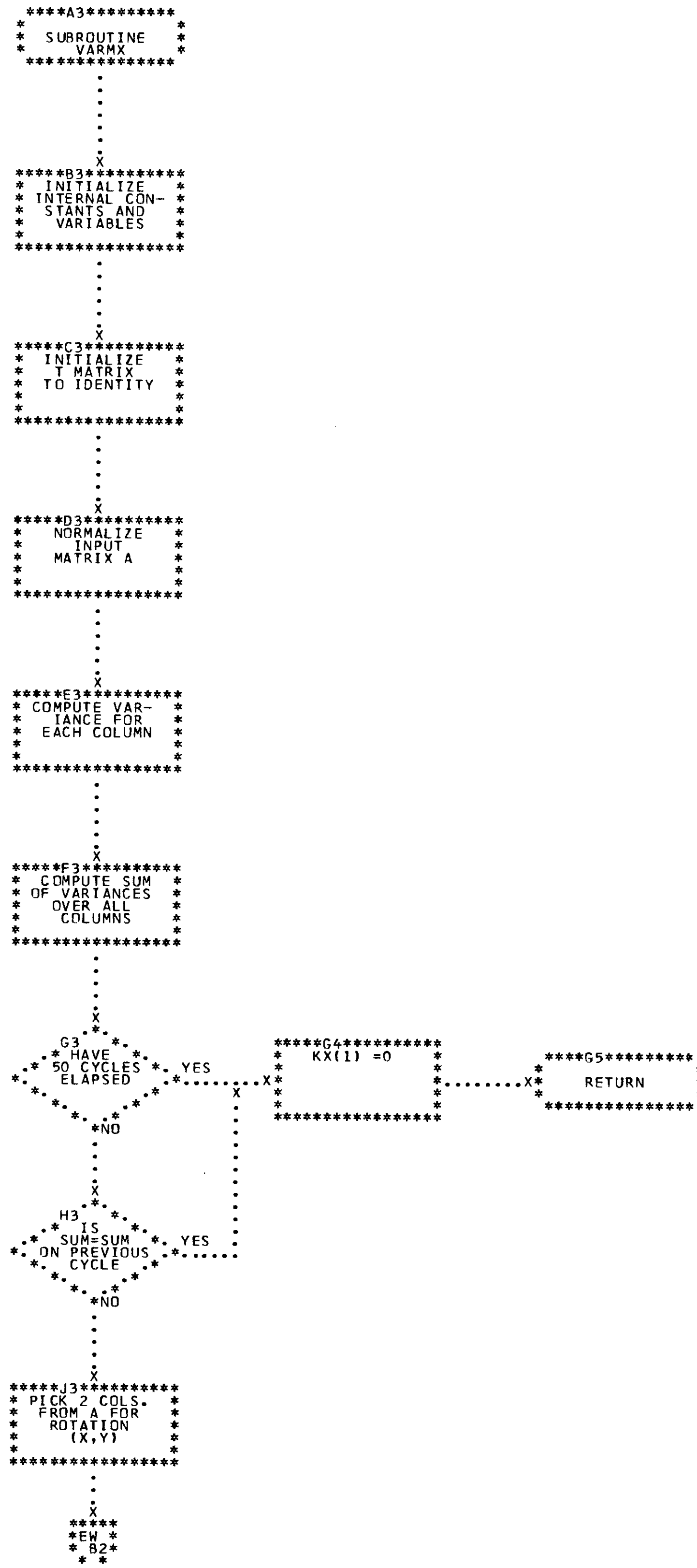
CHART EN

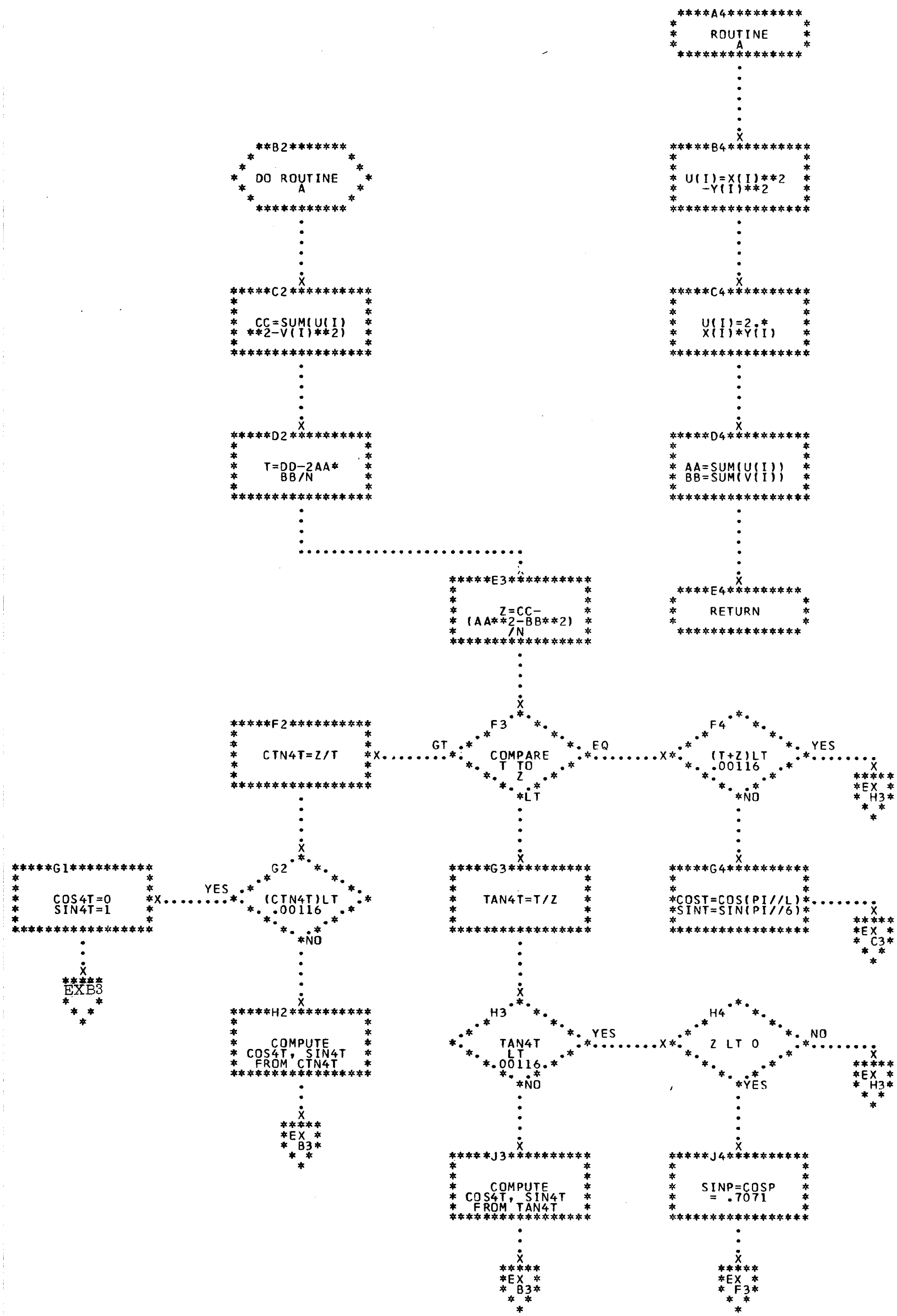

A1
SUBROUTINE VECTR
B1
INIT.RHS OF EQUAT. TO ONES
C1
GET APPROX. SOLUTION
D1
REFINE SOLUTION
E1
NORMALIZE EIGENVECTOR
F1
RETURN
A3
SUBROUTINE COVEC
B3
INITIALIZE
C3
SOLVE SIM. TRIDI EQUATIONS
D3
RETURN

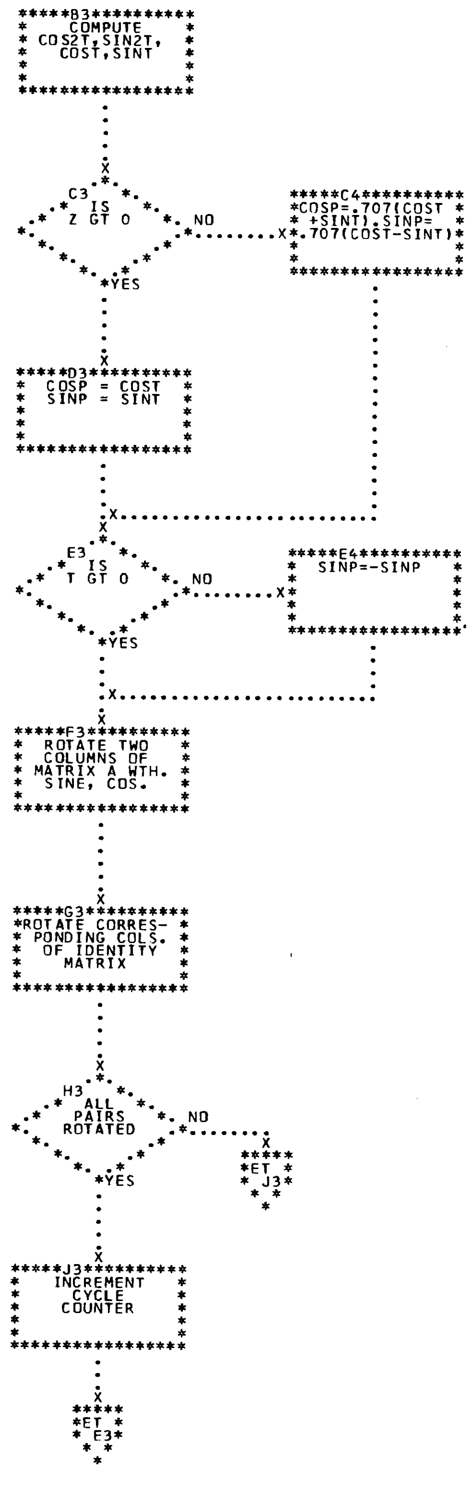

A3
PROGRAM FCTR3
B3
ROT. CNST. LT 1
YES
NO
C3
CALL VARMX
D3
CALL RFOUT
E3
ROT. CONST. LT 2
NO
YES
E4
CALL PROMX
F2
CALL SCORE
F3
ISCR. =0
NO
YES
F4
CALL RFOUT
G3
CALL EXIT

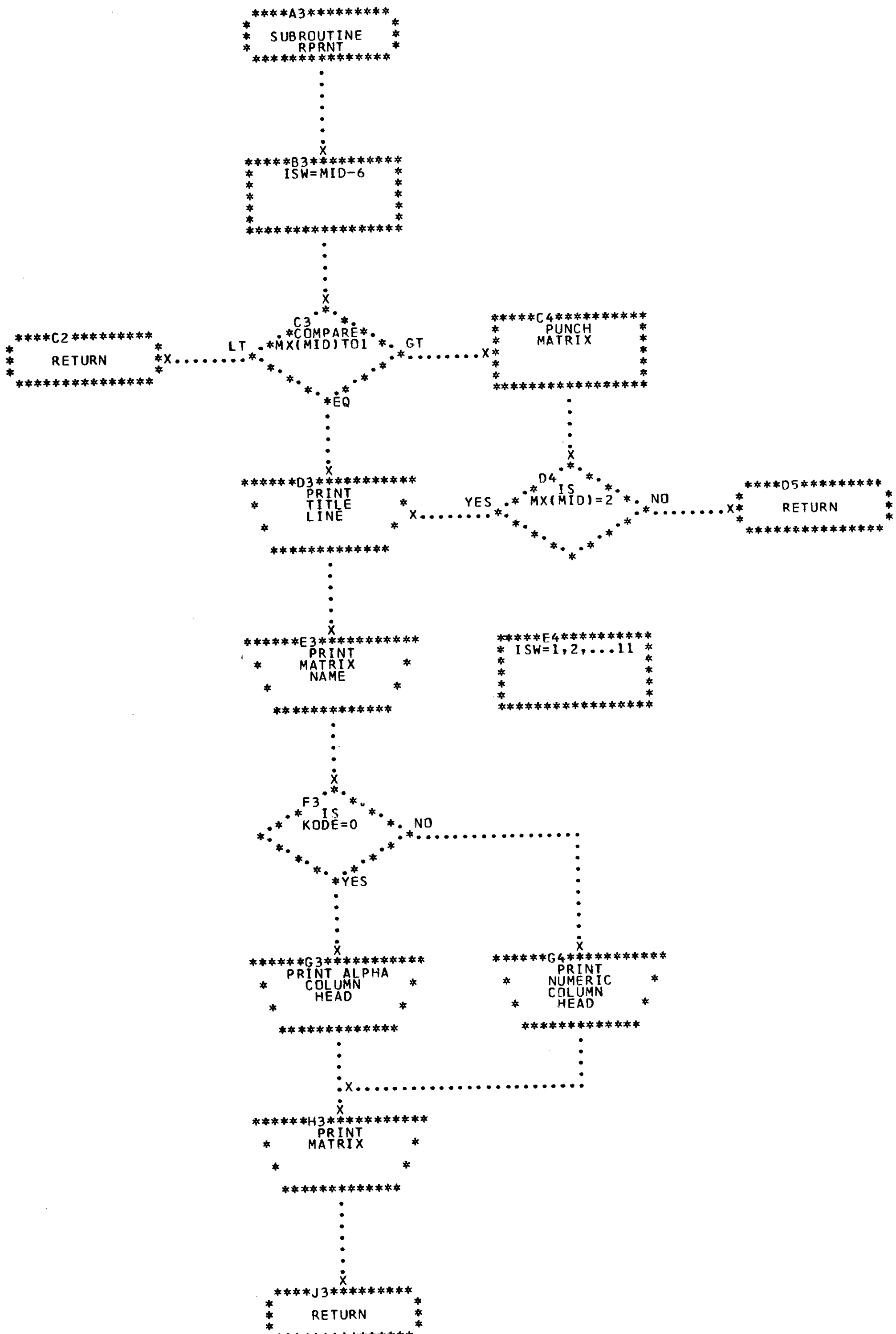

A2
SUBROUTINE RFOUT
B2
IS KX(1)=0
NO
YES
C2
PRINT MATRIX 7
D2
PRINT MATRIX 8
E2
SET B TO IDENTITY MATRIX
F2
RETURN
A3
PRINT MATRIX 9
B3
PRINT MATRIX 11
C3
PRINT MATRIX 10
D3
INVERT MATRIX 10
E3
COMPUTE PATTERN MATRIX (12)
F3
PRINT PATTERN MATRIX
G3
COMPUTE MATRIX 13
H3
PRINT MATRIX 13
J3
COMPUTE MATRIX 15
B4
PRINT MATRIX 15
C4
COMPUTE MATRIX 14
D4
PRINT MATRIX 14
E4
COMPUTE MATRIX 16
F4
PRINT MATRIX 16
G4
RETURN

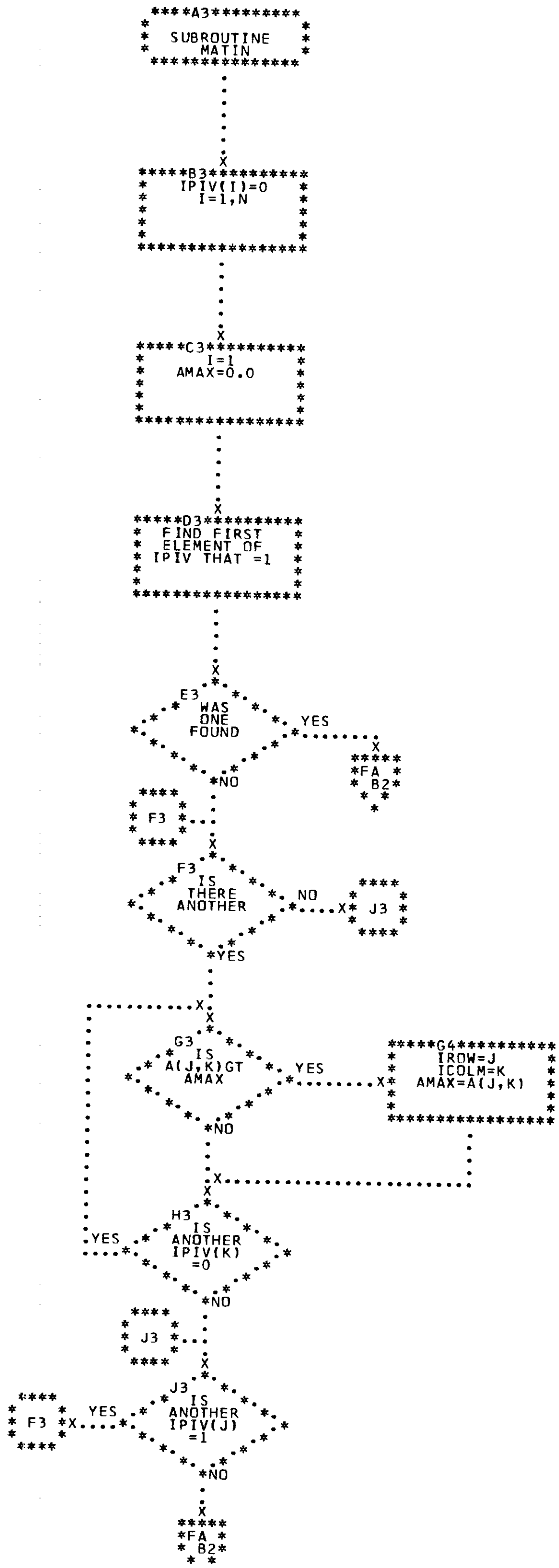

A2
SUBROUTINE PROMX
B2
B=A**(TH)
C2
B**(-1)
D2
COMPUTE ROW NORM- ALIZING VECTOR,H
E2
COLUMN NORMALIZING VECTOR,G
F2
NORMALIZE ROWS,COLS. OF A
G2
E=A**(T+K)
G3
K IS OBLIQUE- NESS POWER
H2
TRANSFORM- ATION MATRIX B=B*E
J2
NORMALIZE COLS. OF B
B4
ROTATE TO REF. VCTR. STRUCTURE MATRIX
C4
COMPUTE CORREL. E=B**(T+1)
D4
RETURN

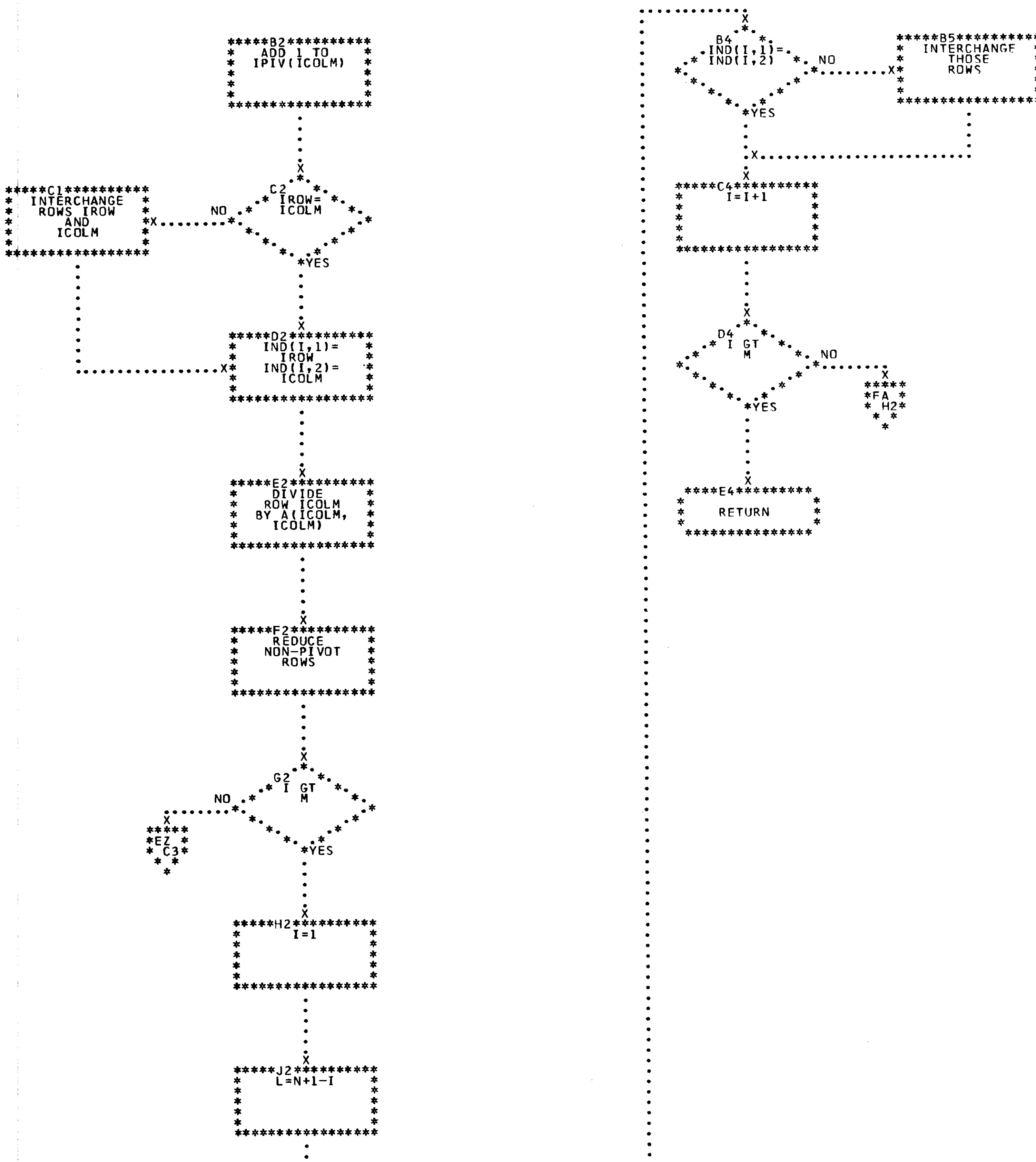

A3
SUBROUTINE VARMX
B3
INITIALIZE INTERNAL CON- STANTS AND VARIABLES
C3
INITIALIZE T MATRIX TO IDENTITY
D3
NORMALIZE INPUT MATRIX A
E3
COMPUTE VAR- IANCE FOR EACH COLUMN
F3
COMPUTE SUM OF VARIANCES OVER ALL COLUMNS
G3
HAVE 50 CYCLES ELAPSED
YES
NO
G4
KX(1) =0
G5
RETURN
H3
IS SUM=SUM ON PREVIOUS CYCLE
YES
NO
J3
PICK 2 COLS. FROM A FOR ROTATION (X,Y)
EW
B2

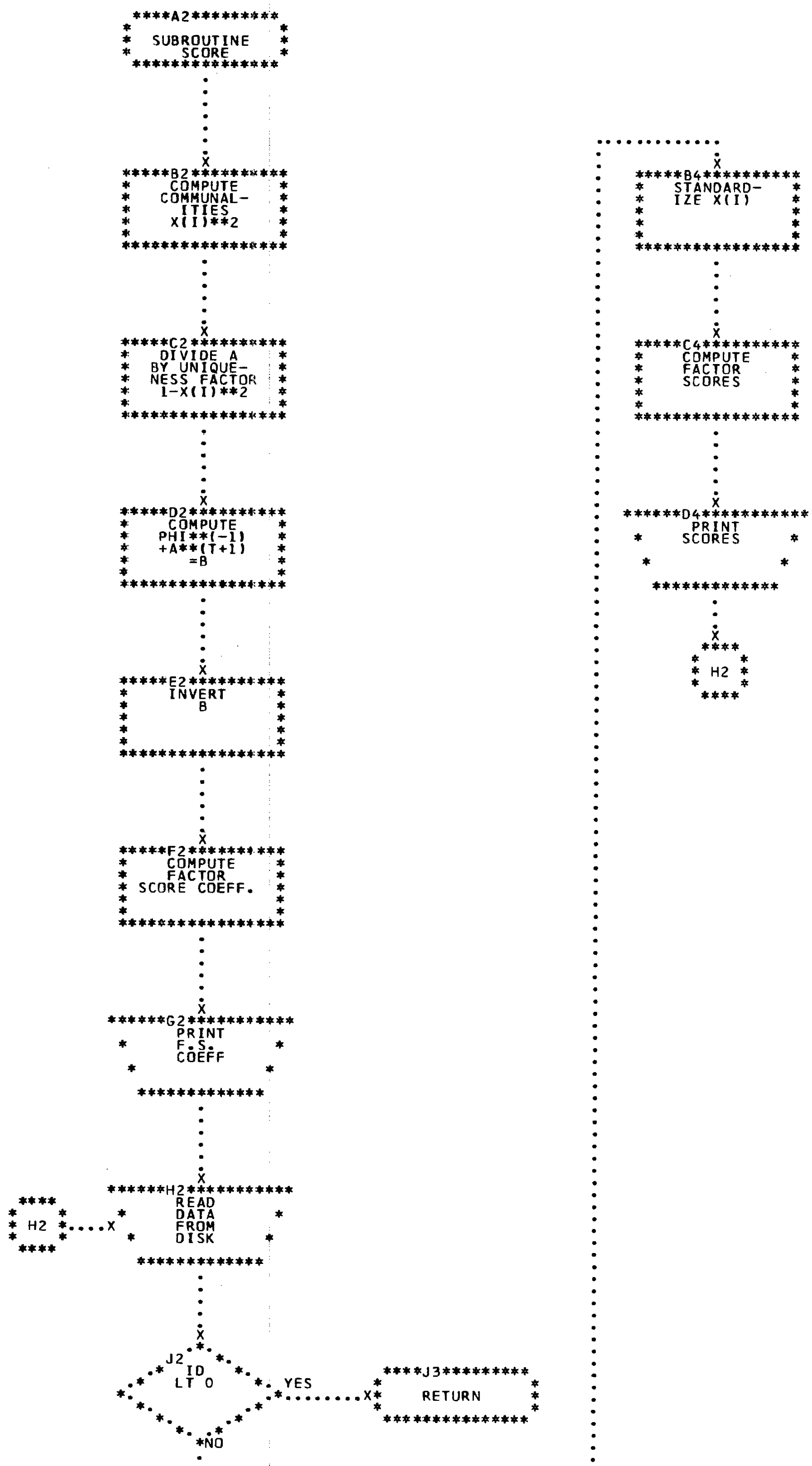

A4
ROUTINE A
B4
U(I)=X(I)**2 -Y(I)**2
C4
U(I)=2.* X(I)*Y(I)
D4
AA=SUM(U(I)) BB=SUM(V(I))
E4
RETURN
B2
DO ROUTINE A
C2
CC=SUM(U(I) **2-V(I)**2)
D2
T=DD-2AA* BB/N
E3
Z=CC- (AA**2-BB**2) /N
F2
CTN4T=Z/T
F3
COMPARE T TO Z
GT
EQ
LT
F4
(T+Z)LT .00116
YES
NO
EX H3
G1
COS4T=0 SIN4T=1
EX B3
G2
(CTN4T)LT .00116
YES
NO
G3
TAN4T=T/Z
G4
*COST=COS(PI//L)
*SINT=SIN(PI//6)
EX C3
H2
COMPUTE COS4T, SIN4T FROM CTN4T
EX B3
H3
TAN4T LT .00116
YES
NO
H4
Z LT 0
NO
YES
EX H3
J3
COMPUTE COS4T, SIN4T FROM TAN4T
EX B3
J4
SINP=COSP = .7071
EX F3

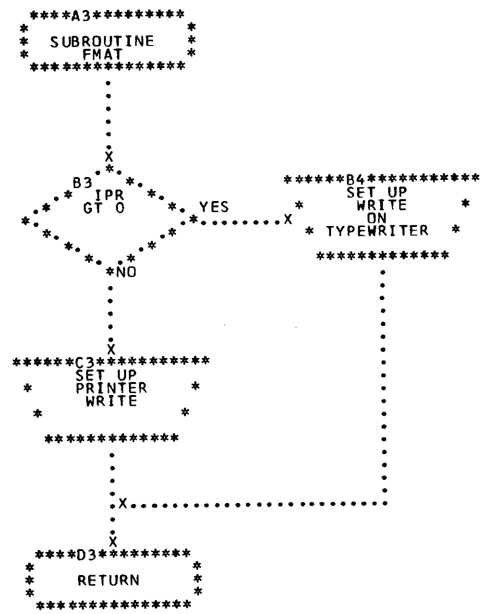

B3
COMPUTE COS2T,SIN2T, COST,SINT
C3
IS Z GT 0
NO
YES
C4
COSP=.707(COST +SINT).SINP= .707(COST-SINT)
D3
COSP = COST
SINP = SINT
E3
IS T GT 0
NO
YES
E4
SINP=-SINP
F3
ROTATE TWO COLUMNS OF MATRIX A WTH. SINE, COS.
G3
ROTATE CORRES- PONDING COLS. OF IDENTITY MATRIX
H3
ALL PAIRS ROTATED
NO
YES
ET J3
J3
INCREMENT CYCLE COUNTER
ET E3


A3
SUBROUTINE RPRNT
B3
ISW=MID-6
C3
COMPARE MX(MID)TO1
LT
GT
EQ
C2
RETURN
C4
PUNCH MATRIX
D3
PRINT TITLE LINE
D4
IS MX(MID)=2
YES
NO
D5
RETURN
E3
PRINT MATRIX NAME
E4
ISW=1,2,...11
F3
IS KODE=0
NO
YES
G3
PRINT ALPHA COLUMN HEAD
G4
PRINT NUMERIC COLUMN HEAD
H3
PRINT MATRIX
J3
RETURN


A3
SUBROUTINE MATIN
B3
IPIV(I)=0 I=1,N
C3
I=1 AMAX=0.0
D3
FIND FIRST ELEMENT OF IPIV THAT =1
E3
WAS ONE FOUND
YES
FA B2
NO
F3
F3
IS THERE ANOTHER
NO
J3
YES
G3
IS A(J,K)GT AMAX
YES
G4
IROW=J ICOLM=K AMAX=A(J,K)
NO
H3
IS ANOTHER IPIV(K) =0
YES
NO
J3
J3
IS ANOTHER IPIV(J) =1
YES
F3
NO
FA B2


B2
ADD 1 TO IPIV(ICOLM)
C1
INTERCHANGE ROWS IROW AND ICOLM
C2
IROW= ICOLM
NO
YES
D2
IND(I,1)= IROW
IND(I,2)= ICOLM
E2
DIVIDE ROW ICOLM BY A(ICOLM, ICOLM)
F2
REDUCE NON-PIVOT ROWS
G2
I GT M
NO
EZ C3
YES
H2
I=1
J2
L=N+1-I
B4
IND(I,1)= IND(I,2)
NO
YES
B5
INTERCHANGE THOSE ROWS
C4
I=I+1
D4
I GT M
NO
FA H2
YES
E4
RETURN


A2
SUBROUTINE SCORE
B2
COMPUTE COMMUNAL- ITIES X(I)**2
C2
DIVIDE A BY UNIQUE- NESS FACTOR 1-X(I)**2
D2
COMPUTE PHI**(-1) +A**(T+1) =B
E2
INVERT B
F2
COMPUTE FACTOR SCORE COEFF.
G2
PRINT F.S. COEFF
H2
H2
READ DATA FROM DISK
J2
ID LT 0
YES
NO
J3
RETURN
B4
STANDARD- IZE X(I)
C4
COMPUTE FACTOR SCORES
D4
PRINT SCORES
H2


A3
SUBROUTINE FMAT
B3
IPR GT 0
YES
NO
B4
SET UP WRITE ON TYPEWRITER
C3
SET UP PRINTER WRITE
D3
RETURN

Y20-0093-1

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Printed in U.S.A. Y20-0093-1