IBM APPLICATION PROGRAM

IBM 1130 SCIENTIFIC SUBROUTINE PACKAGE

SYSTEM MANUAL

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

(1130-CM-02X)

CONTENTS

INTRODUCTION

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

SYMBOLS USED

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

BLOCK NUMBERING

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

EQUATIONS

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

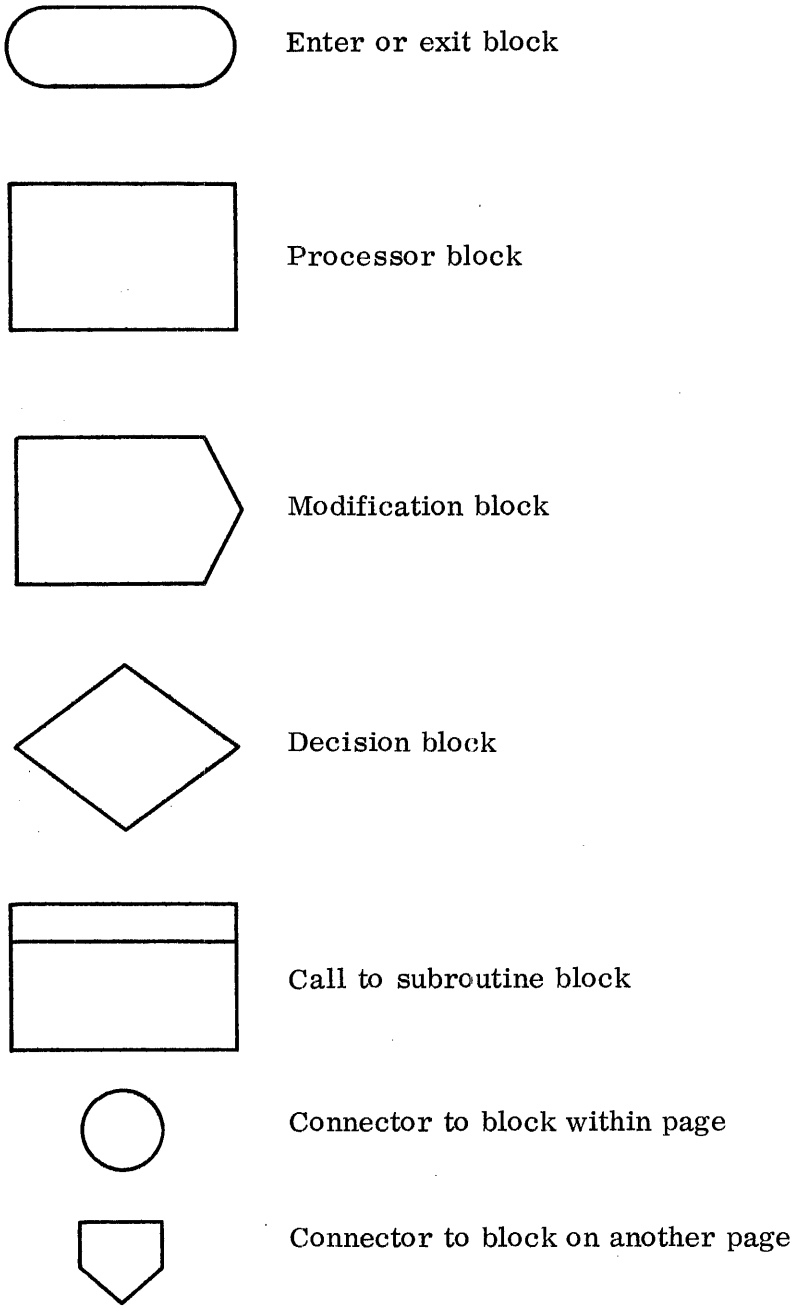


Figure 1. Flowchart blocks

GUIDE TO FLOWCHARTS

FLOWCHARTS

Statistics

Data Screening

Elementary Statistics

Correlation

Multiple Linear Regression

Polynomial Regression

Canonical Correlation

Analysis of Variance

Discriminant Analysis

Factor Analysis

Time Series

Nonparametric Statistics

Random Number Generators

Special Matrix Operations

Matrices

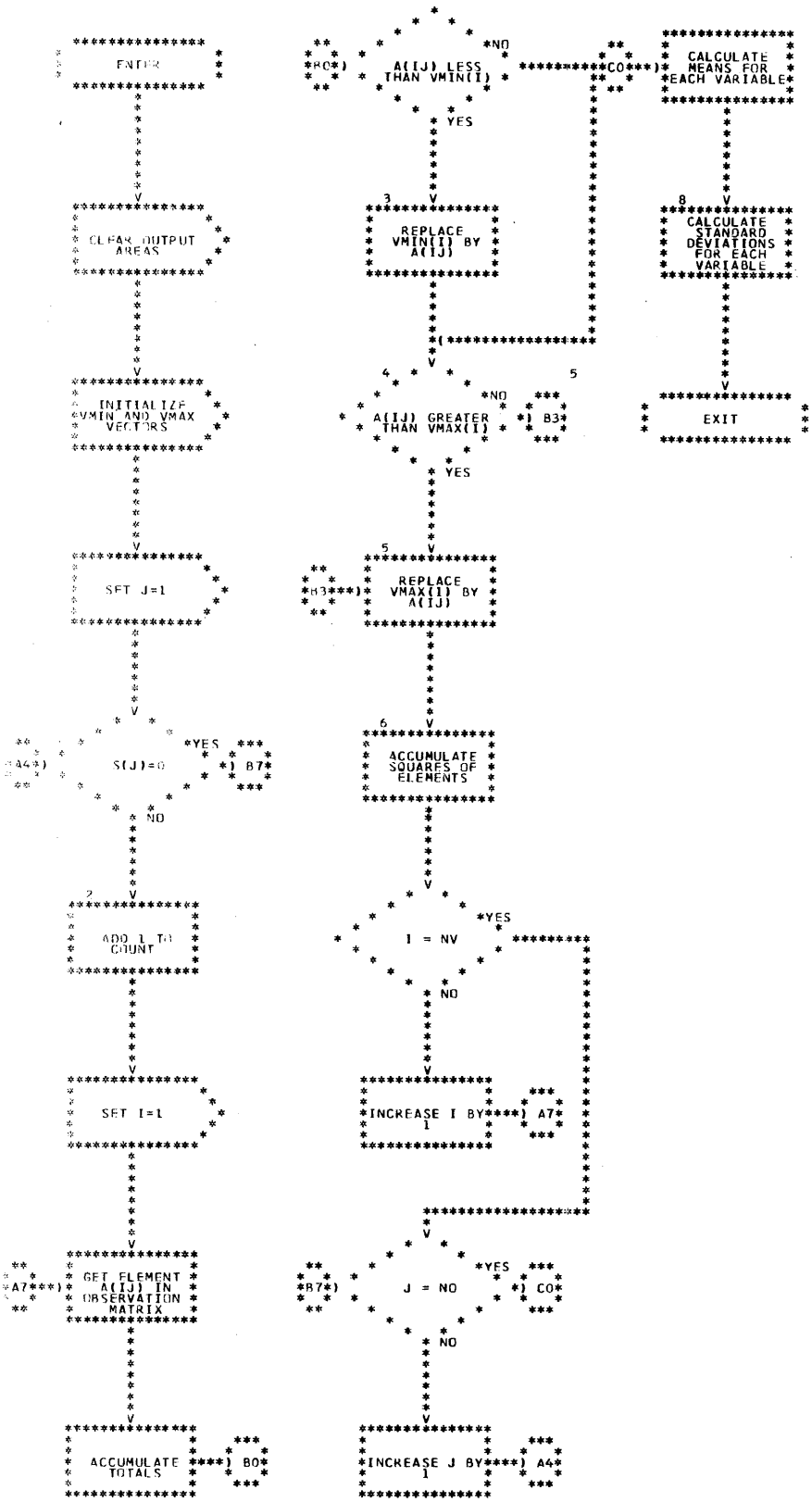
Other Mathematics

Integration

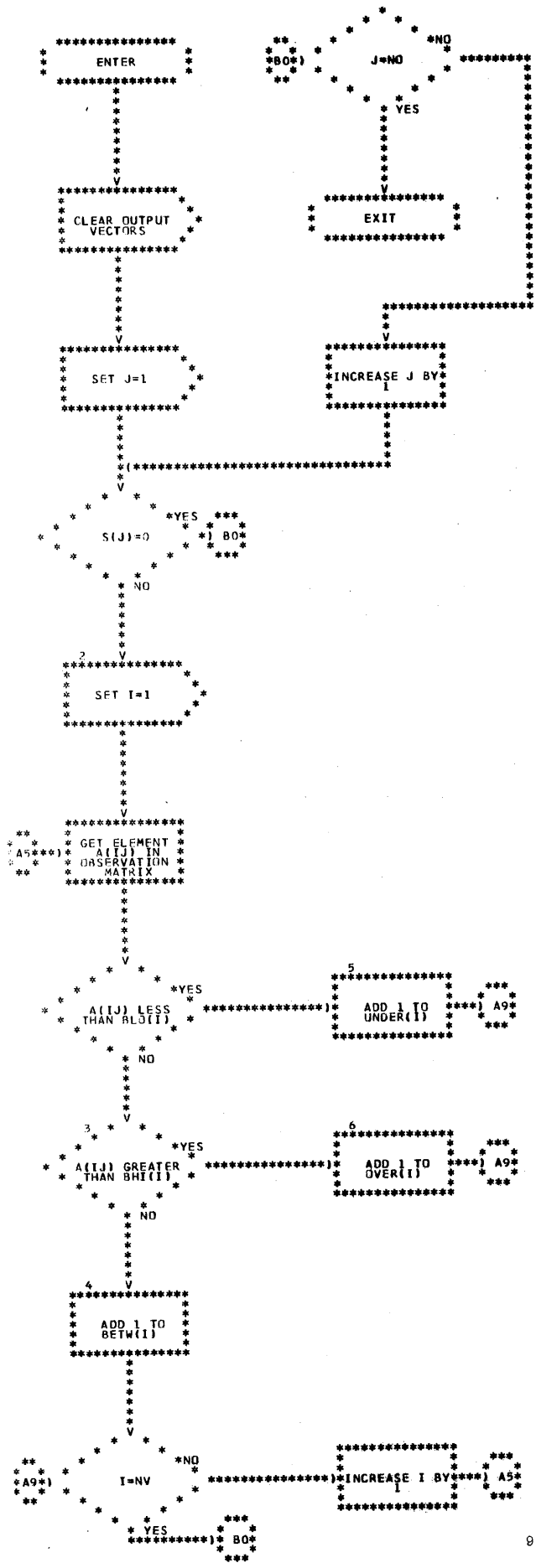
Fourier Analysis

Special Operations and Mathematical Functions

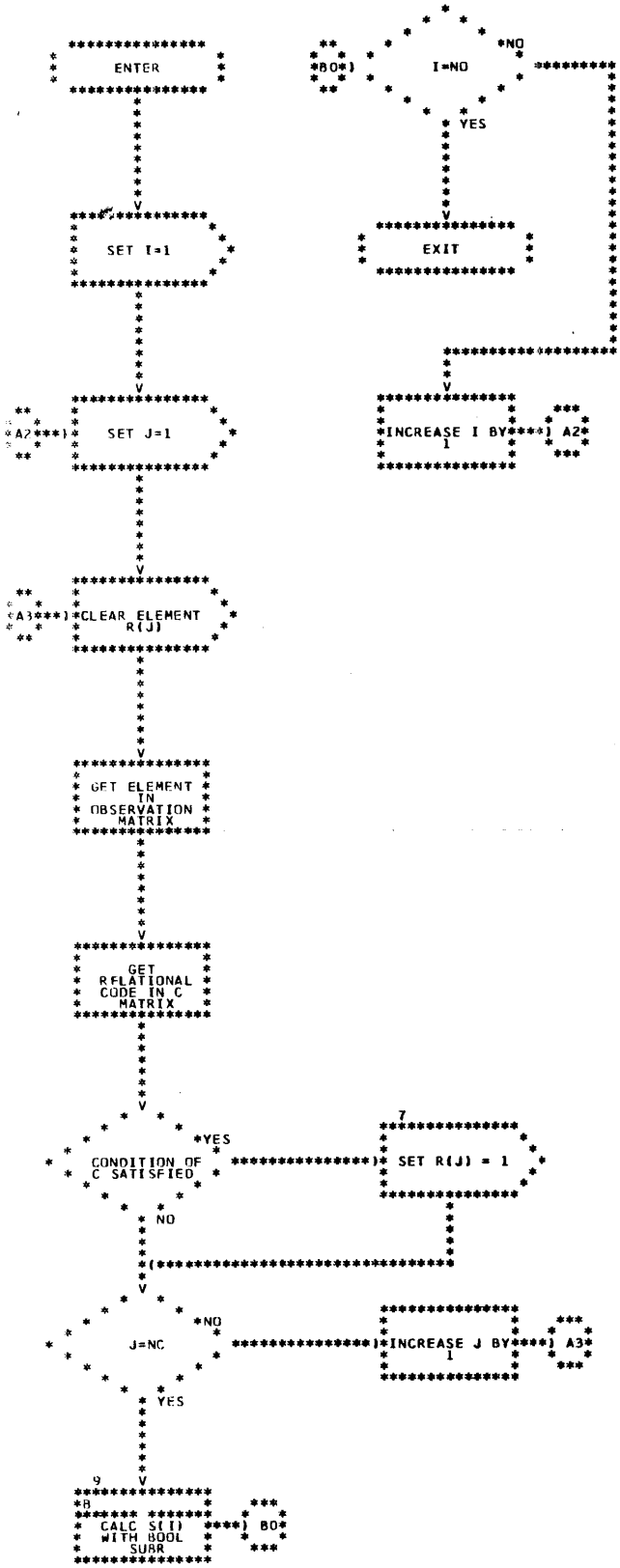
SUBROUTINE TALLY

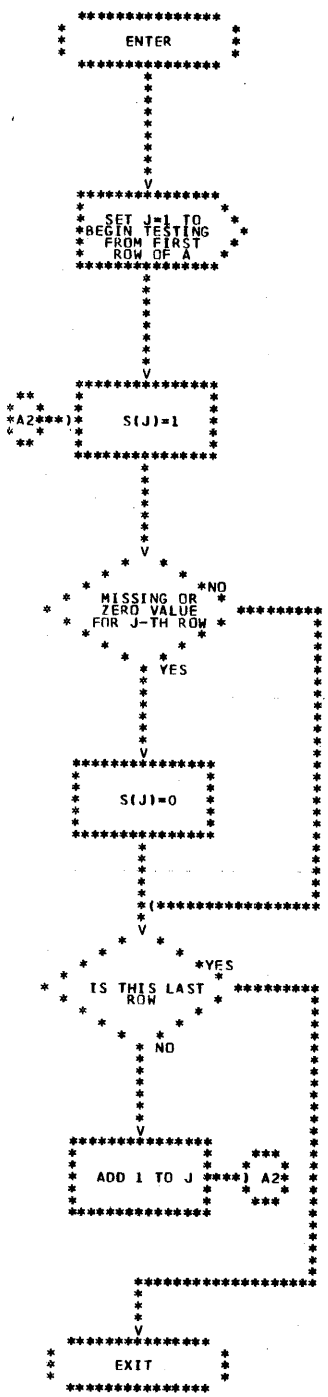


SUBROUTINE BOUND

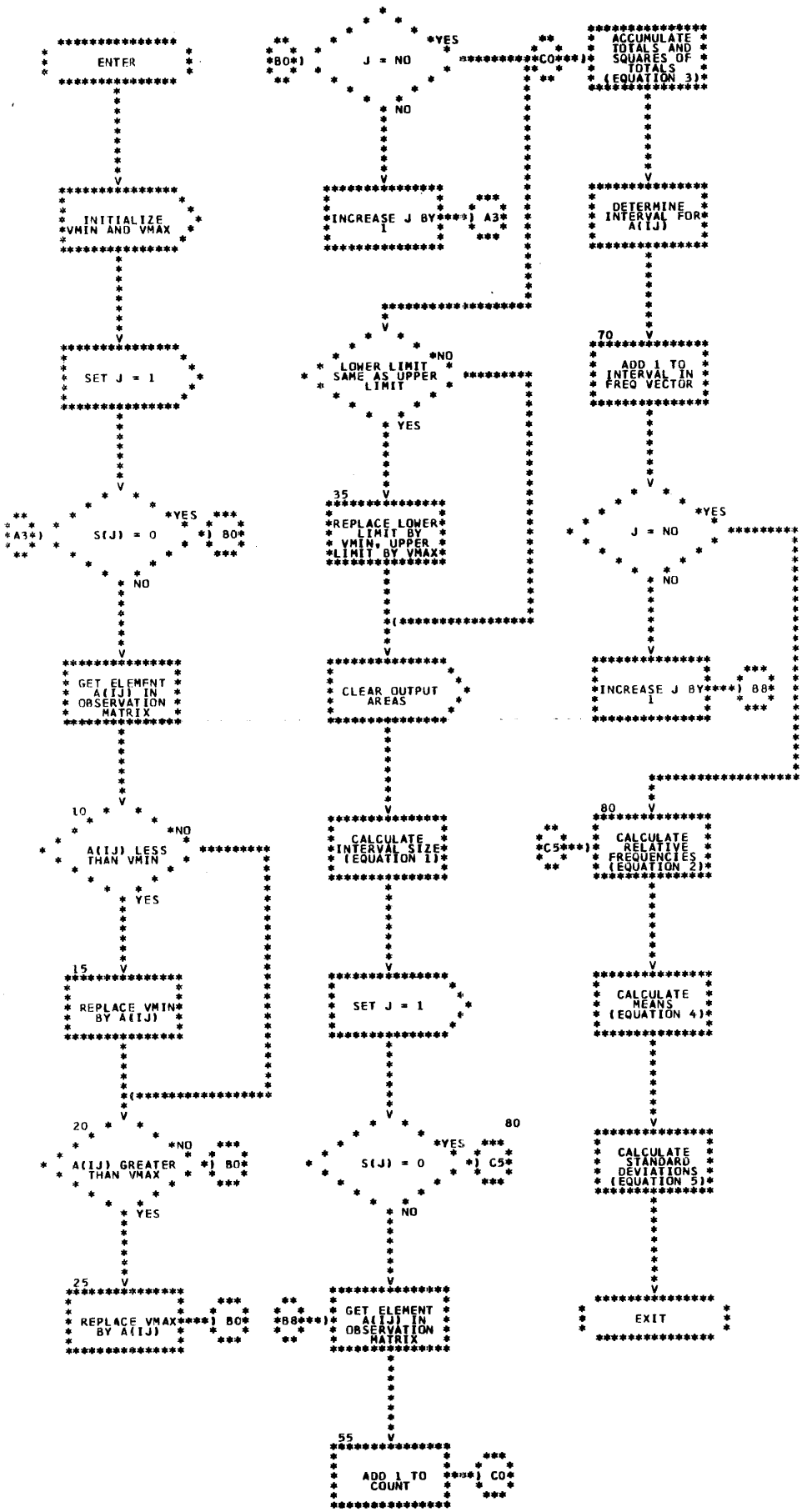


SUBROUTINE SUBST

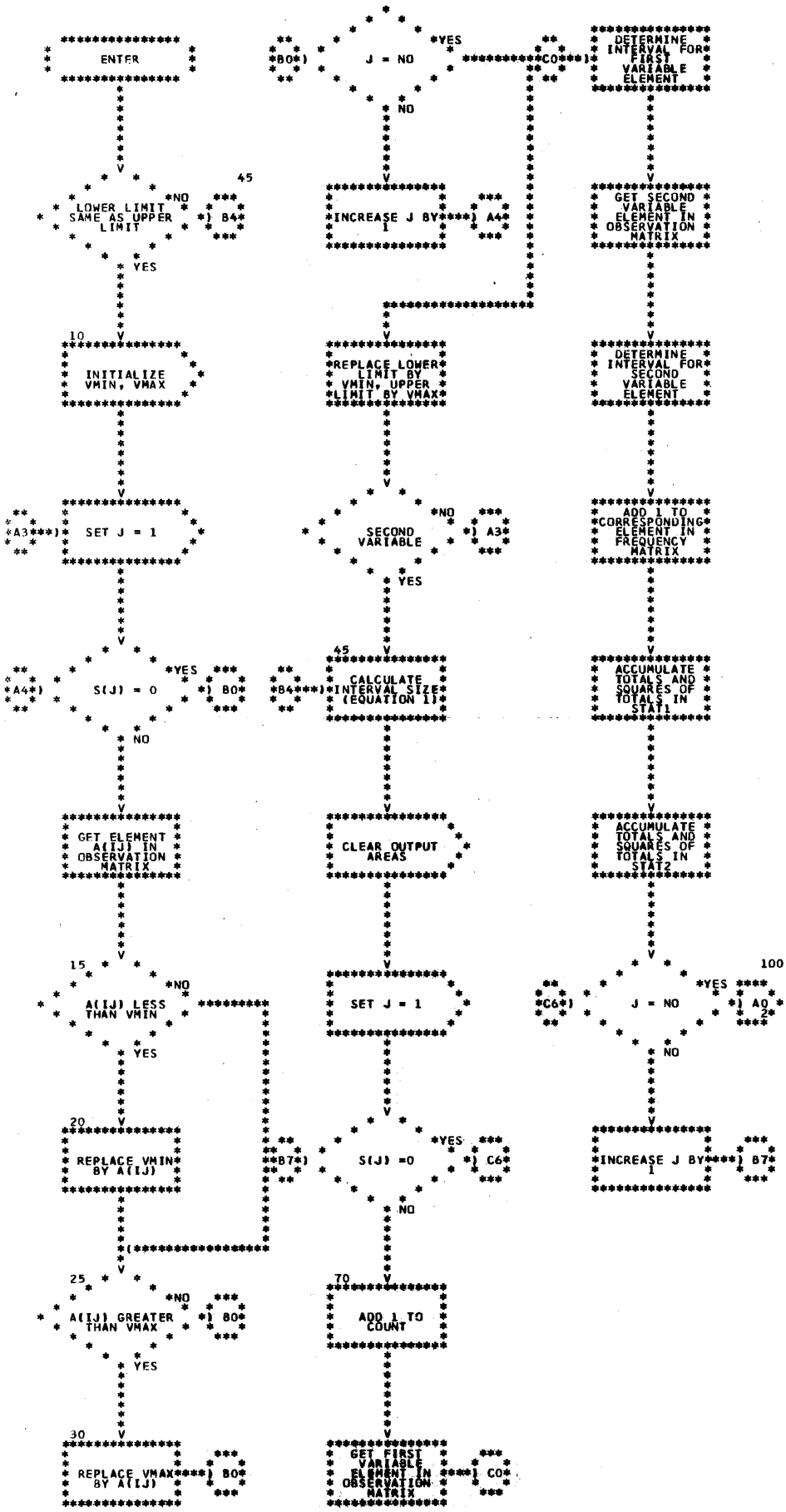



ENTER
SET J=1 TO BEGIN TESTING FROM FIRST ROW OF A
S(J)=1
A2
NO
MISSING OR ZERO VALUE FOR J-TH ROW
YES
S(J)=0
YES
IS THIS LAST ROW
NO
ADD 1 TO J
A2
EXIT

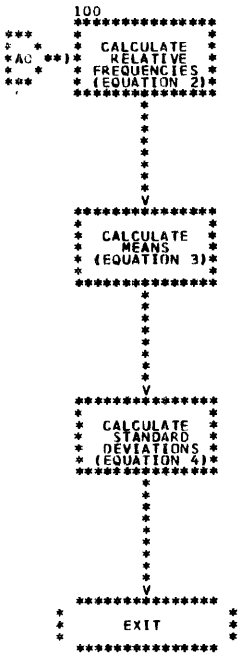
SUBROUTINE TAB1



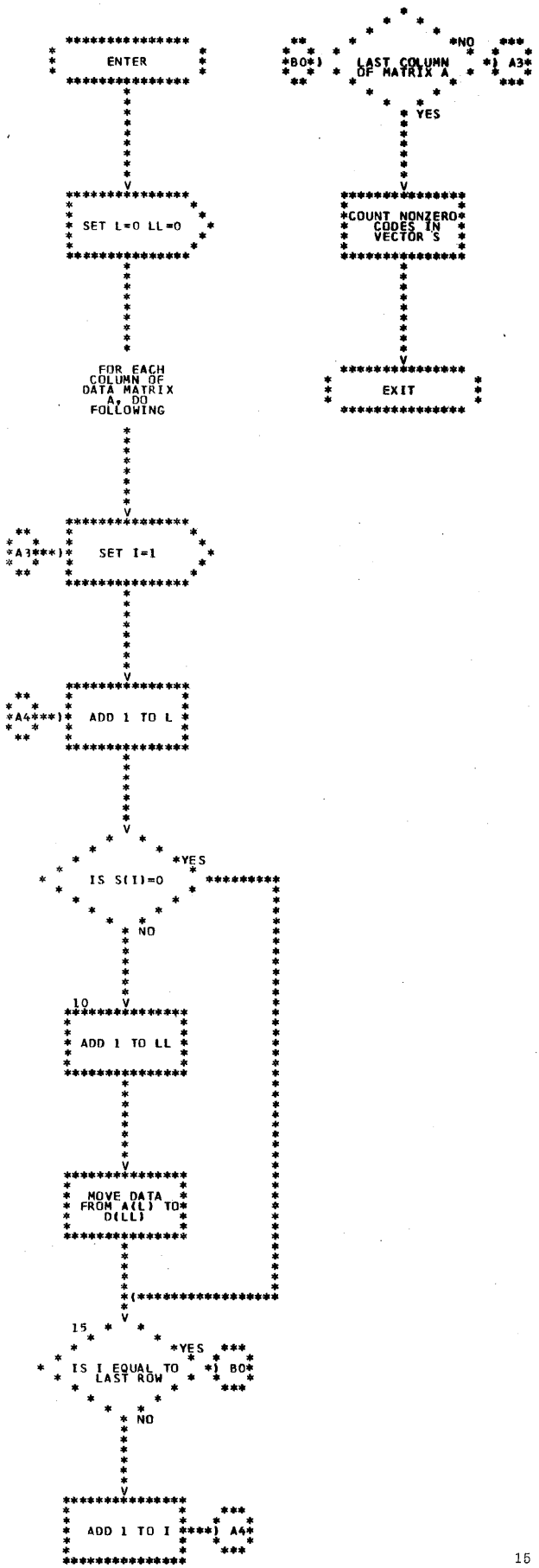
SUBROUTINE TAB2



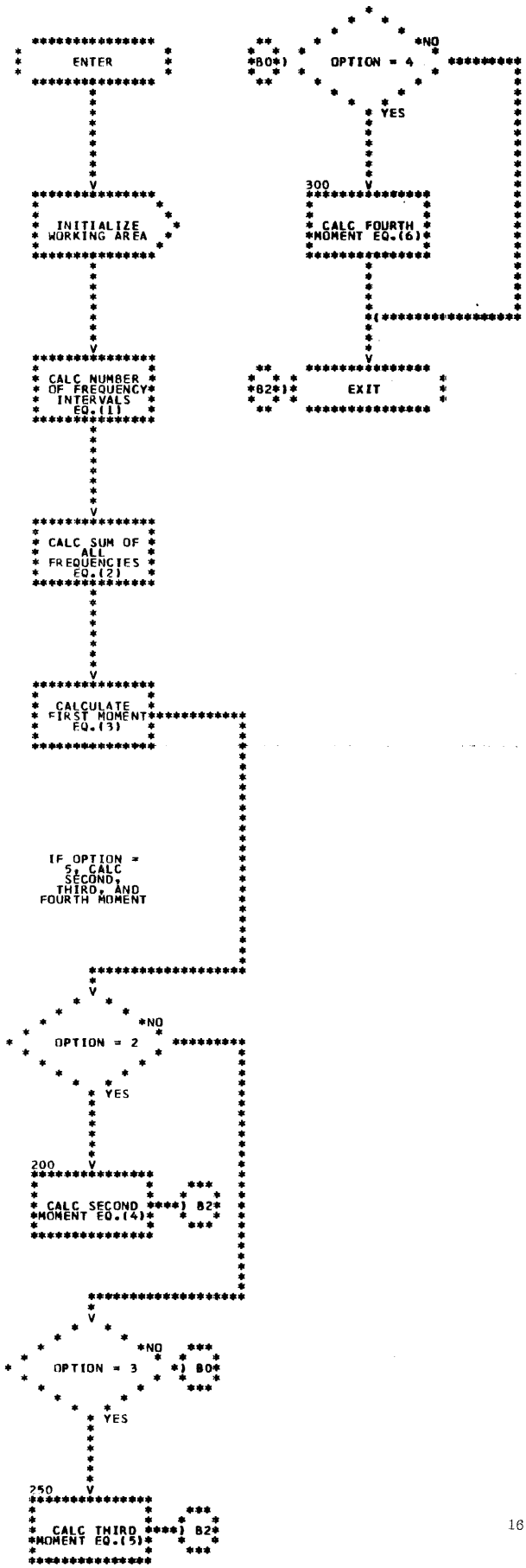
SUBROUTINE TAB2

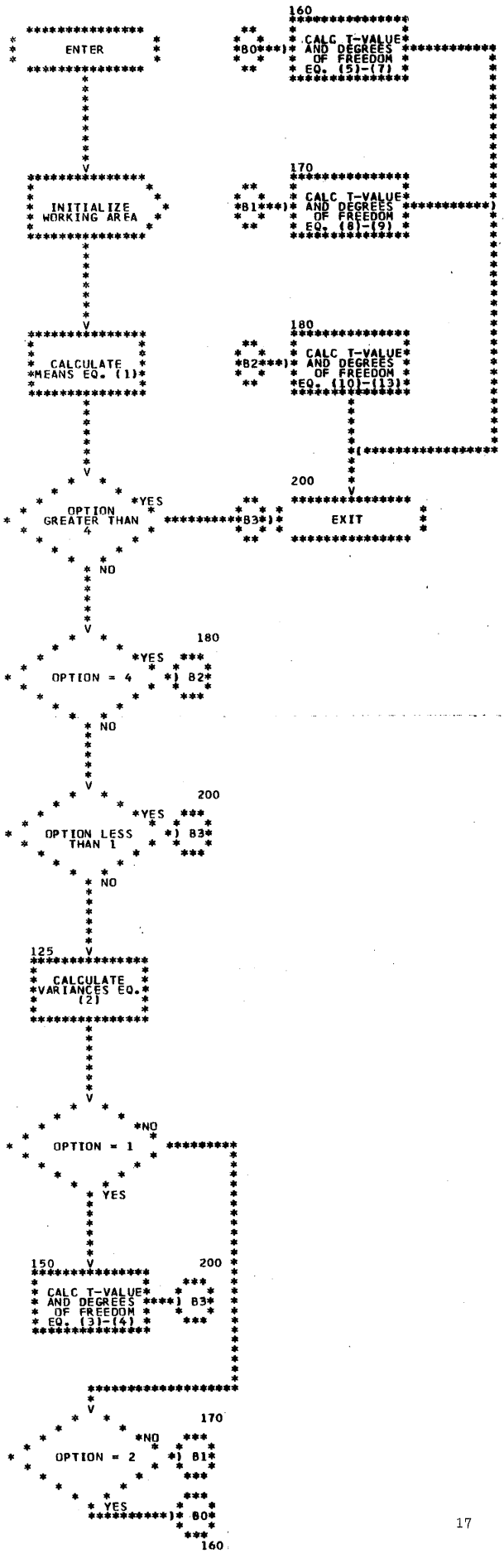


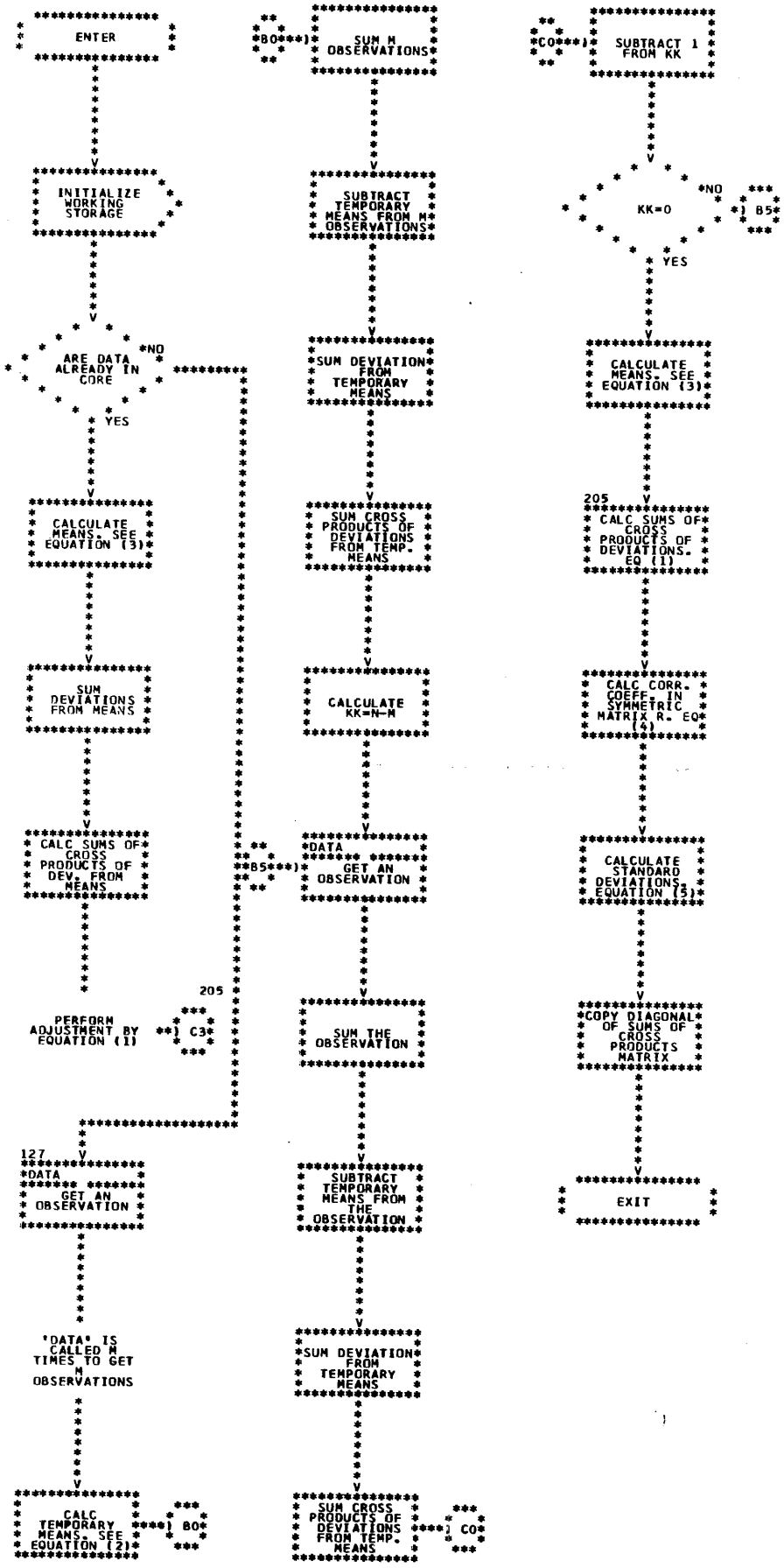
SUBROUTINE SUBMX

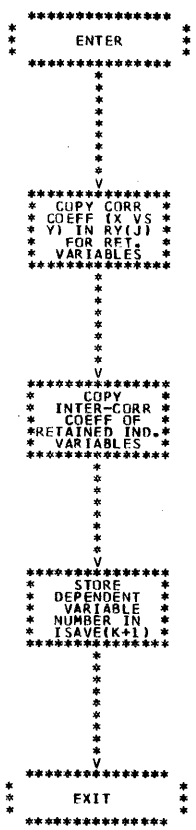


SUBROUTINE MOMEN

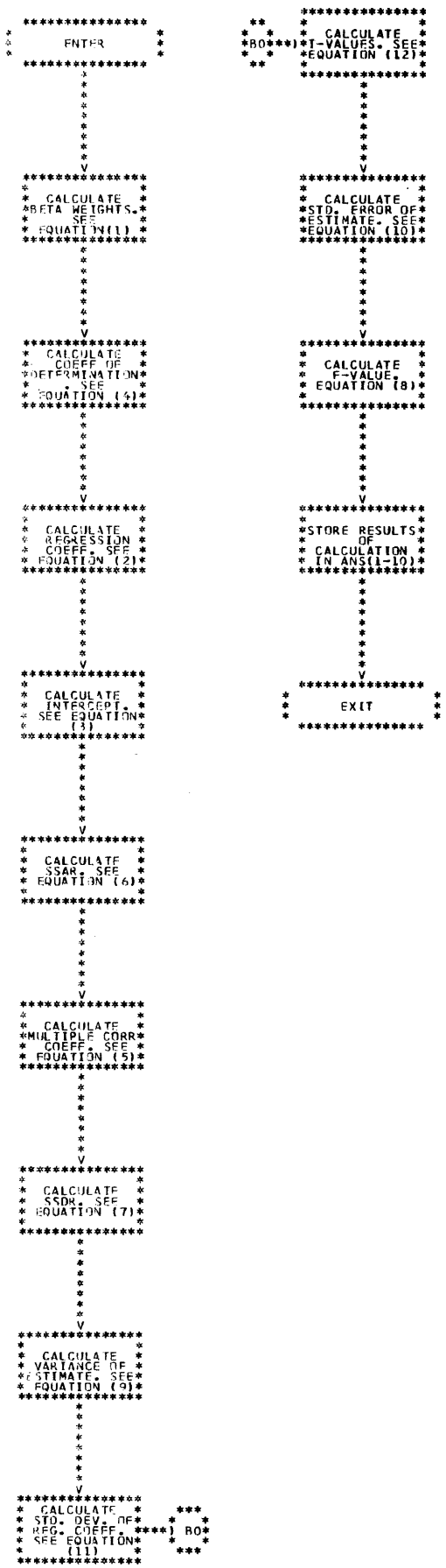



ENTER
INITIALIZE WORKING AREA
CALCULATE MEANS EQ. (1)
OPTION GREATER THAN 4
YES
NO
OPTION = 4
180
B2
OPTION LESS THAN 1
200
B3
CALCULATE VARIANCES EQ. (2)
125
OPTION = 1
YES
NO
CALC T-VALUE AND DEGREES OF FREEDOM EQ. (3)-(4)
150
OPTION = 2
B1
B0
160
170
CALC T-VALUE AND DEGREES OF FREEDOM EQ. (5)-(7)
CALC T-VALUE AND DEGREES OF FREEDOM EQ. (8)-(9)
CALC T-VALUE AND DEGREES OF FREEDOM EQ. (10)-(13)
EXIT

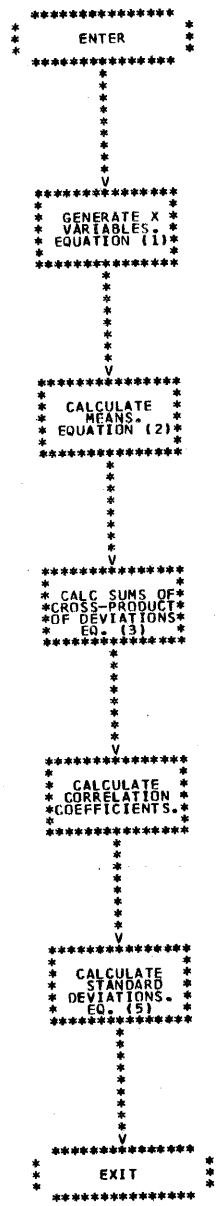

ENTER
INITIALIZE WORKING STORAGE
ARE DATA ALREADY IN CORE
*NO
YES
CALCULATE MEANS. SEE EQUATION (3)
SUM DEVIATIONS FROM MEANS
CALC SUMS OF CROSS PRODUCTS OF DEV. FROM MEANS
PERFORM ADJUSTMENT BY EQUATION (1)
205
C3
B5
127
*DATA GET AN OBSERVATION
'DATA' IS CALLED M TIMES TO GET M OBSERVATIONS
CALC TEMPORARY MEANS. SEE EQUATION (2)
B0
SUM M OBSERVATIONS
SUBTRACT TEMPORARY MEANS FROM M OBSERVATIONS
SUM DEVIATION FROM TEMPORARY MEANS
SUM CROSS PRODUCTS OF DEVIATIONS FROM TEMP. MEANS
CALCULATE KK=N-M
*DATA GET AN OBSERVATION
SUM THE OBSERVATION
SUBTRACT TEMPORARY MEANS FROM THE OBSERVATION
SUM DEVIATION FROM TEMPORARY MEANS
SUM CROSS PRODUCTS OF DEVIATIONS FROM TEMP. MEANS
C0
SUBTRACT 1 FROM KK
KK=0
*NO
YES
B5
CALCULATE MEANS. SEE EQUATION (3)
205
CALC SUMS OF CROSS PRODUCTS OF DEVIATIONS. EQ (1)
CALC CORR. COEFF. IN SYMMETRIC MATRIX R. EQ (4)
CALCULATE STANDARD DEVIATIONS. EQUATION (5)
COPY DIAGONAL OF SUMS OF CROSS PRODUCTS MATRIX
EXIT


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

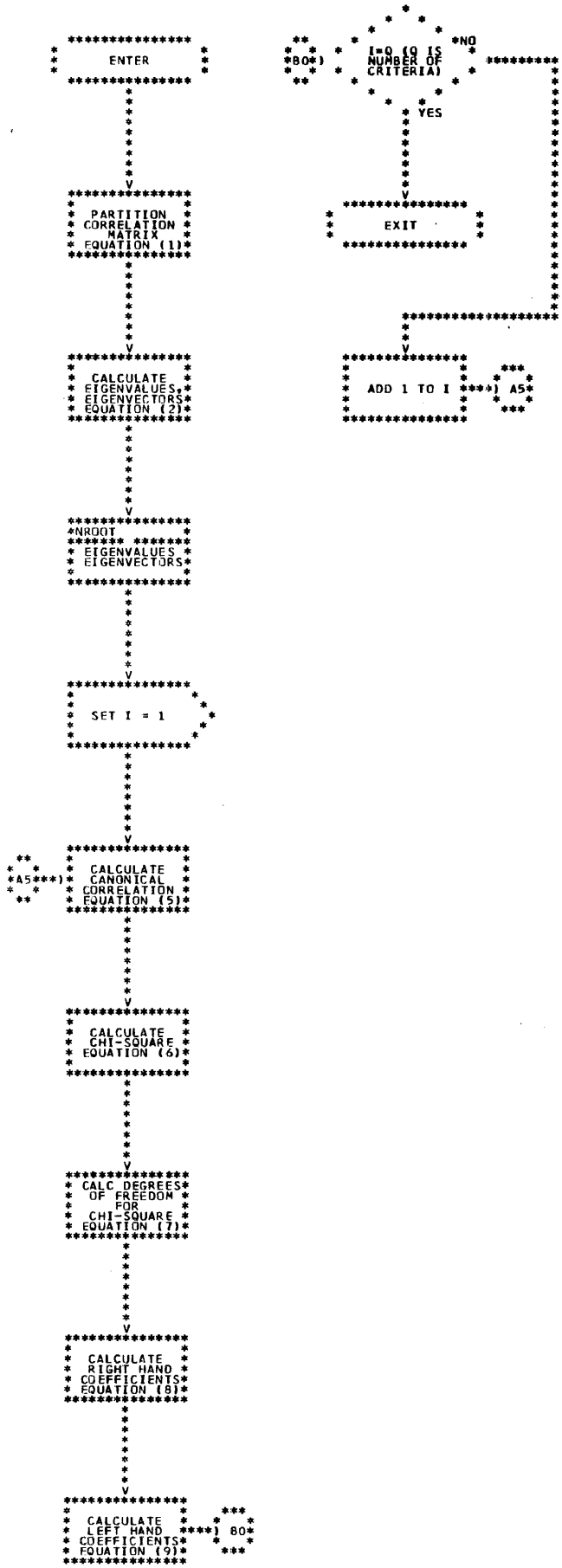
SUBROUTINE MULTR

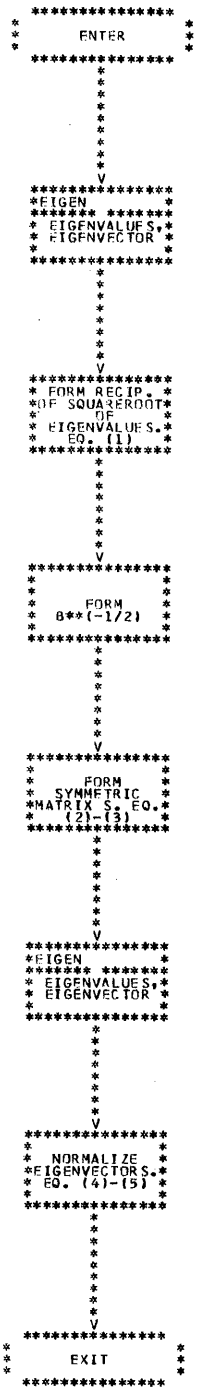


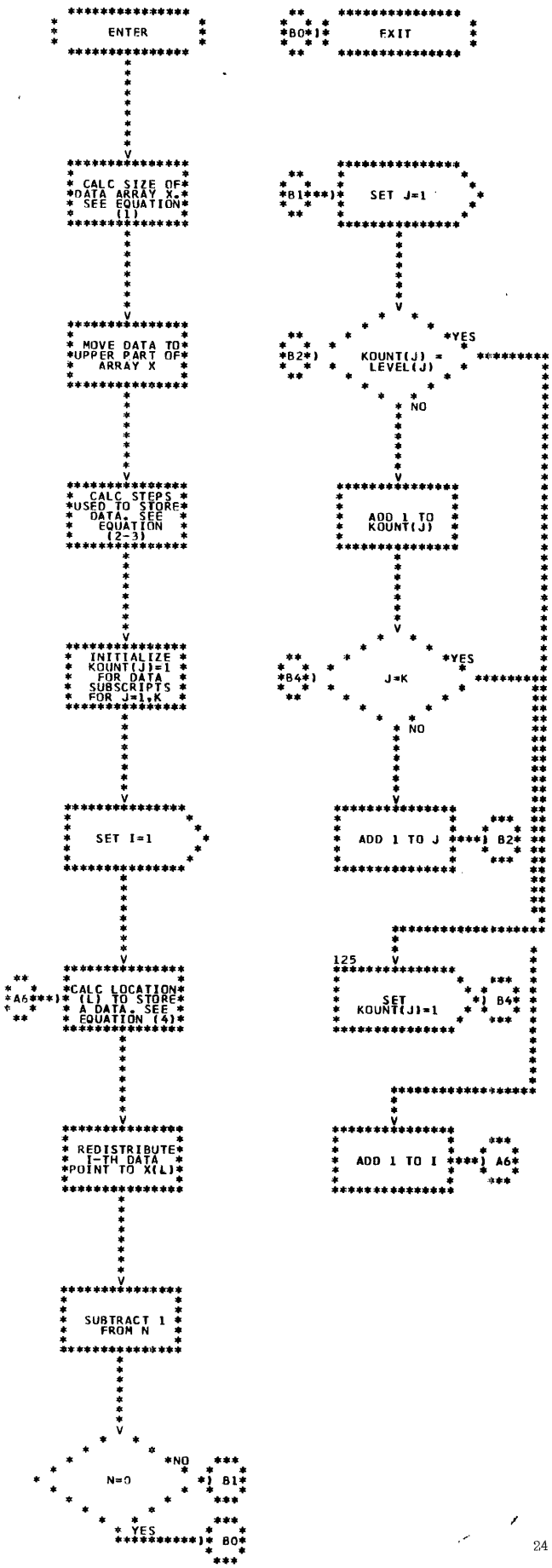
SUBROUTINE GDATA



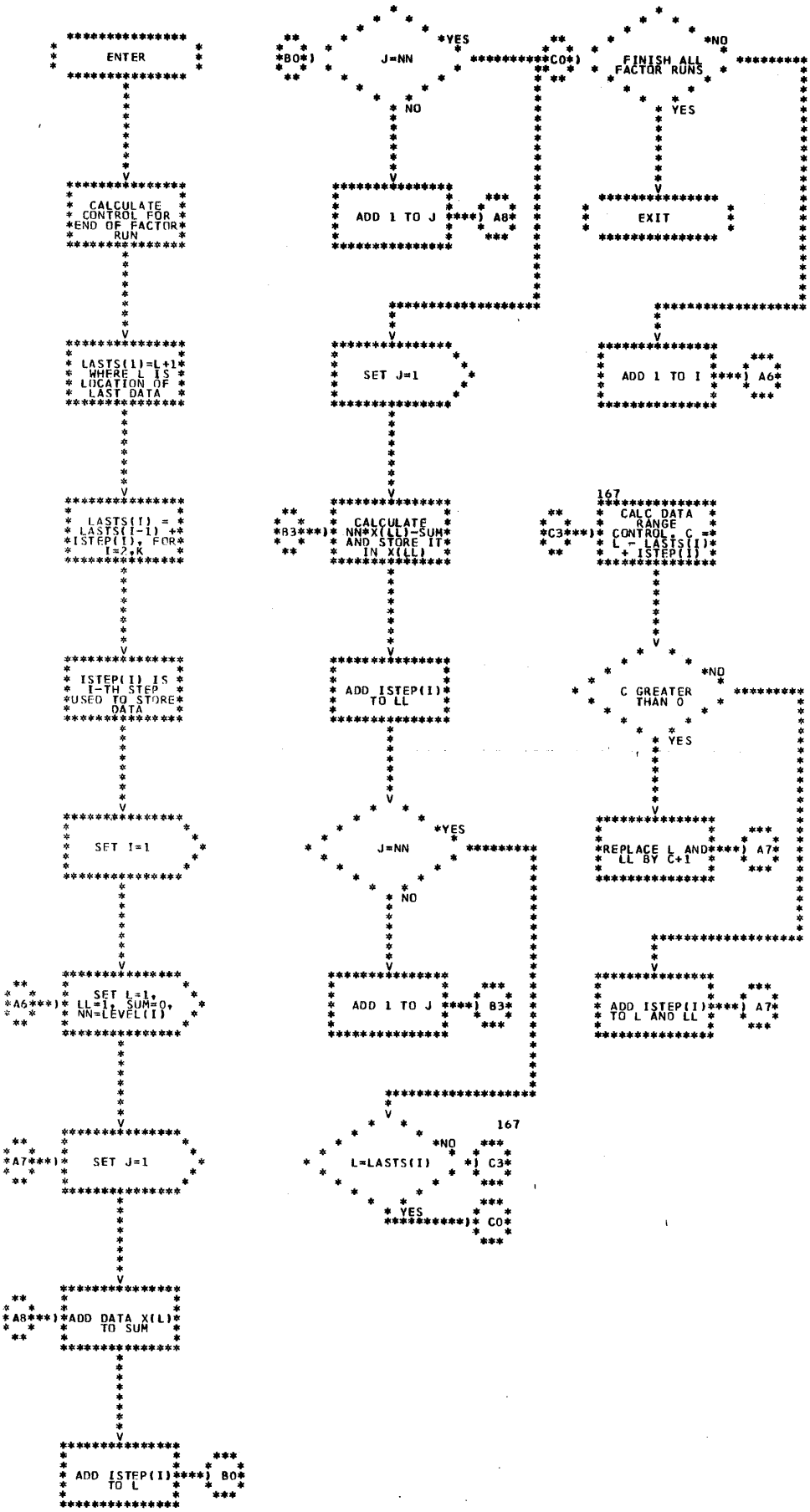
SUBROUTINE CANOR



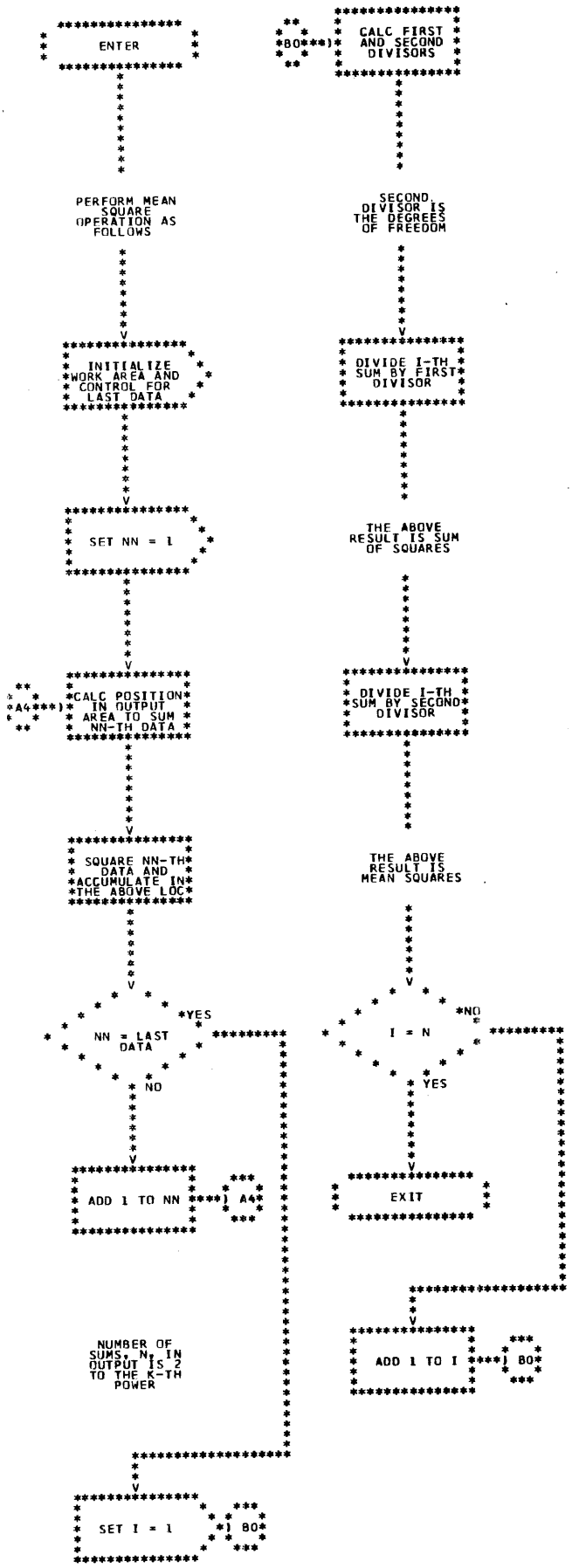

ENTER
EIGEN
EIGENVALUES,
EIGENVECTOR
FORM RECIP. OF SQUAREROOT OF EIGENVALUES. EQ. (1)
FORM B**(-1/2)
FORM SYMMETRIC MATRIX S. EQ. (2)-(3)
EIGEN
EIGENVALUES,
EIGENVECTOR
NORMALIZE EIGENVECTORS. EQ. (4)-(5)
EXIT

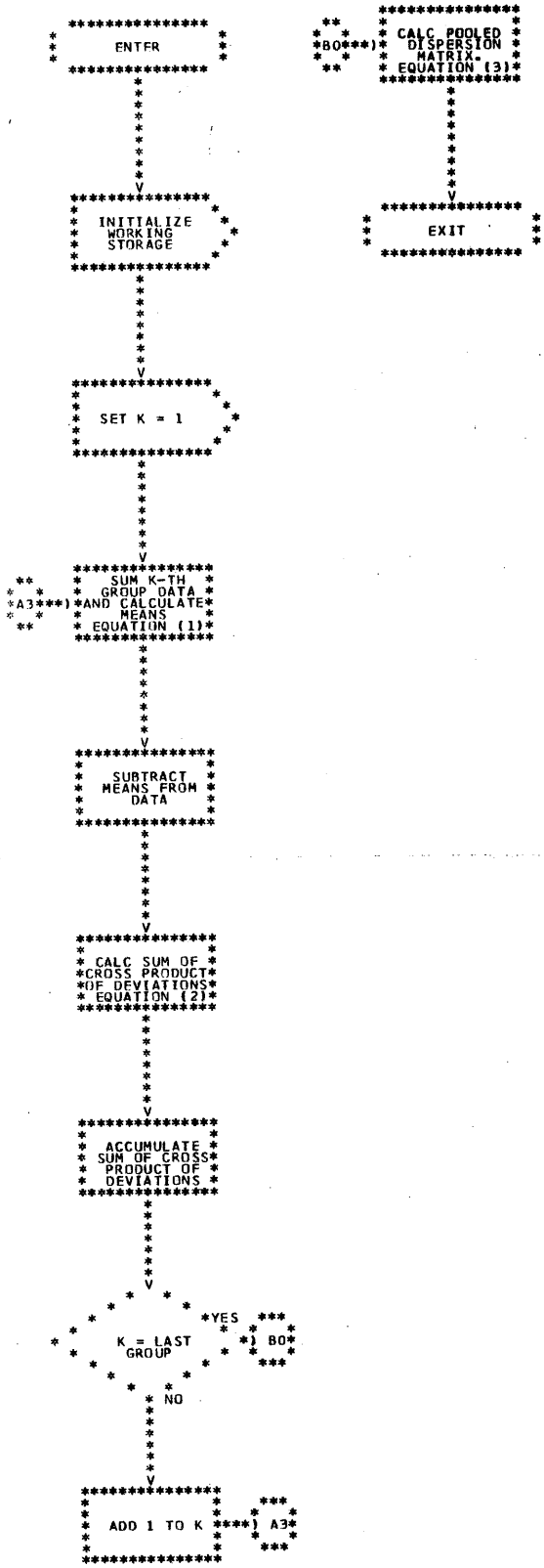

ENTER
B0
EXIT
CALC SIZE OF DATA ARRAY X. SEE EQUATION (1)
B1
SET J=1
MOVE DATA TO UPPER PART OF ARRAY X
B2
KOUNT(J) = LEVEL(J)
YES
NO
CALC STEPS USED TO STORE DATA. SEE EQUATION (2-3)
ADD 1 TO KOUNT(J)
INITIALIZE KOUNT(J)=1 FOR DATA SUBSCRIPTS FOR J=1,K
B4
J=K
YES
NO
SET I=1
ADD 1 TO J
B2
125
SET KOUNT(J)=1
B4
A6
CALC LOCATION (L) TO STORE A DATA. SEE EQUATION (4)
ADD 1 TO I
A6
REDISTRIBUTE I-TH DATA POINT TO X(L)
SUBTRACT 1 FROM N
N=0
NO
B1
YES
B0

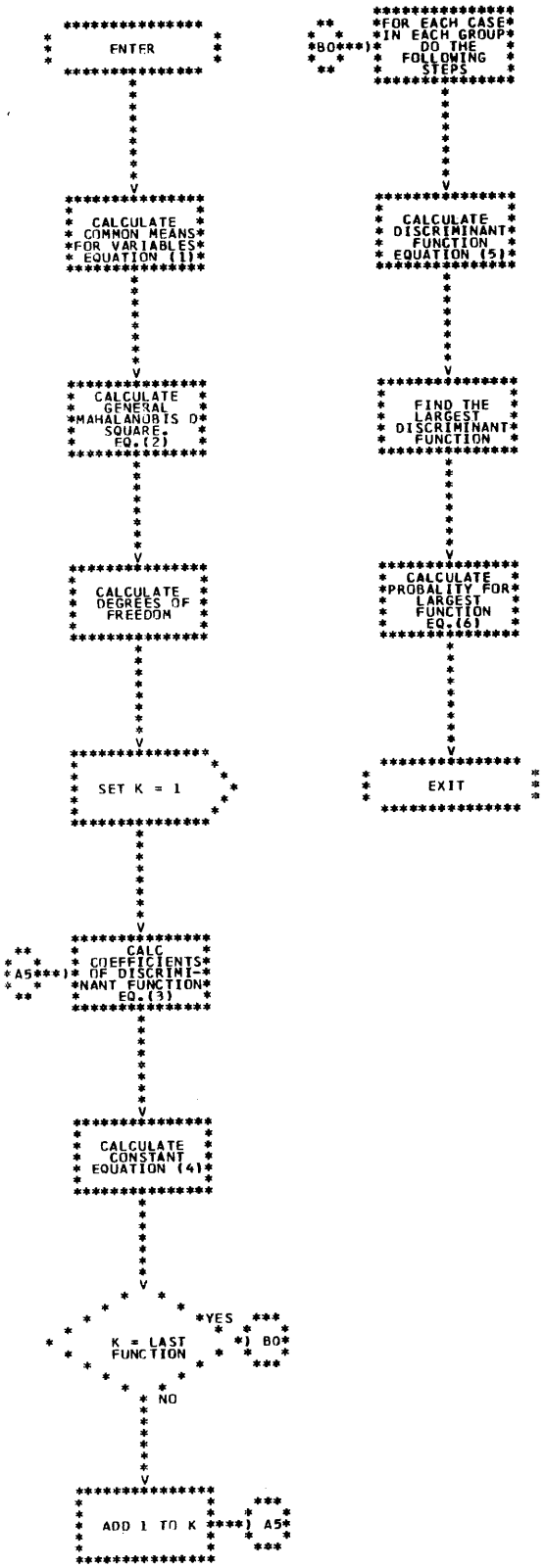
SUBROUTINE AVCAL

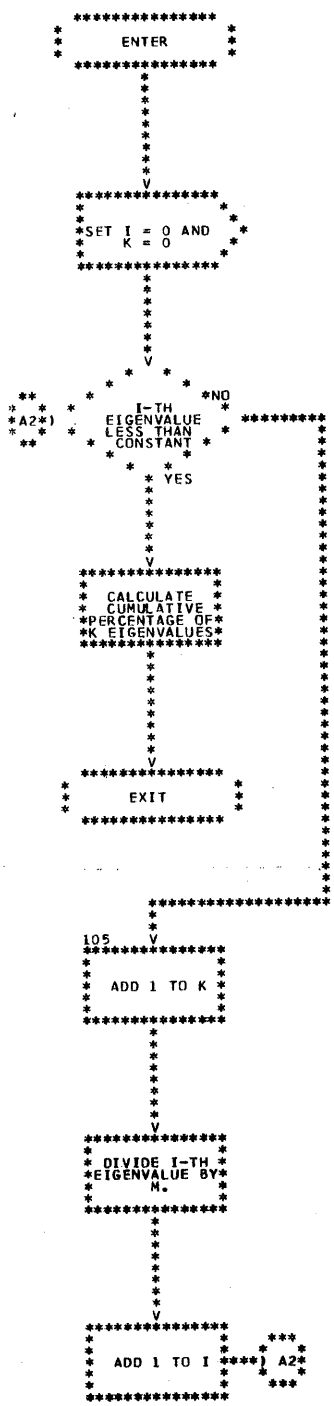


SUBROUTINE MEANQ

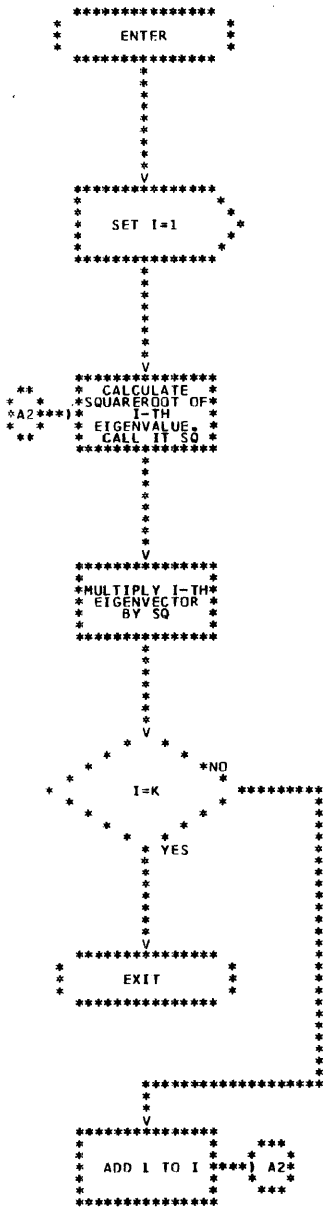



ENTER
INITIALIZE WORKING STORAGE
SET K = 1
SUM K-TH GROUP DATA AND CALCULATE MEANS EQUATION (1)
SUBTRACT MEANS FROM DATA
CALC SUM OF CROSS PRODUCT OF DEVIATIONS EQUATION (2)
ACCUMULATE SUM OF CROSS PRODUCT OF DEVIATIONS
K = LAST GROUP
YES
NO
ADD 1 TO K
A3
B0
CALC POOLED DISPERSION MATRIX. EQUATION (3)
EXIT

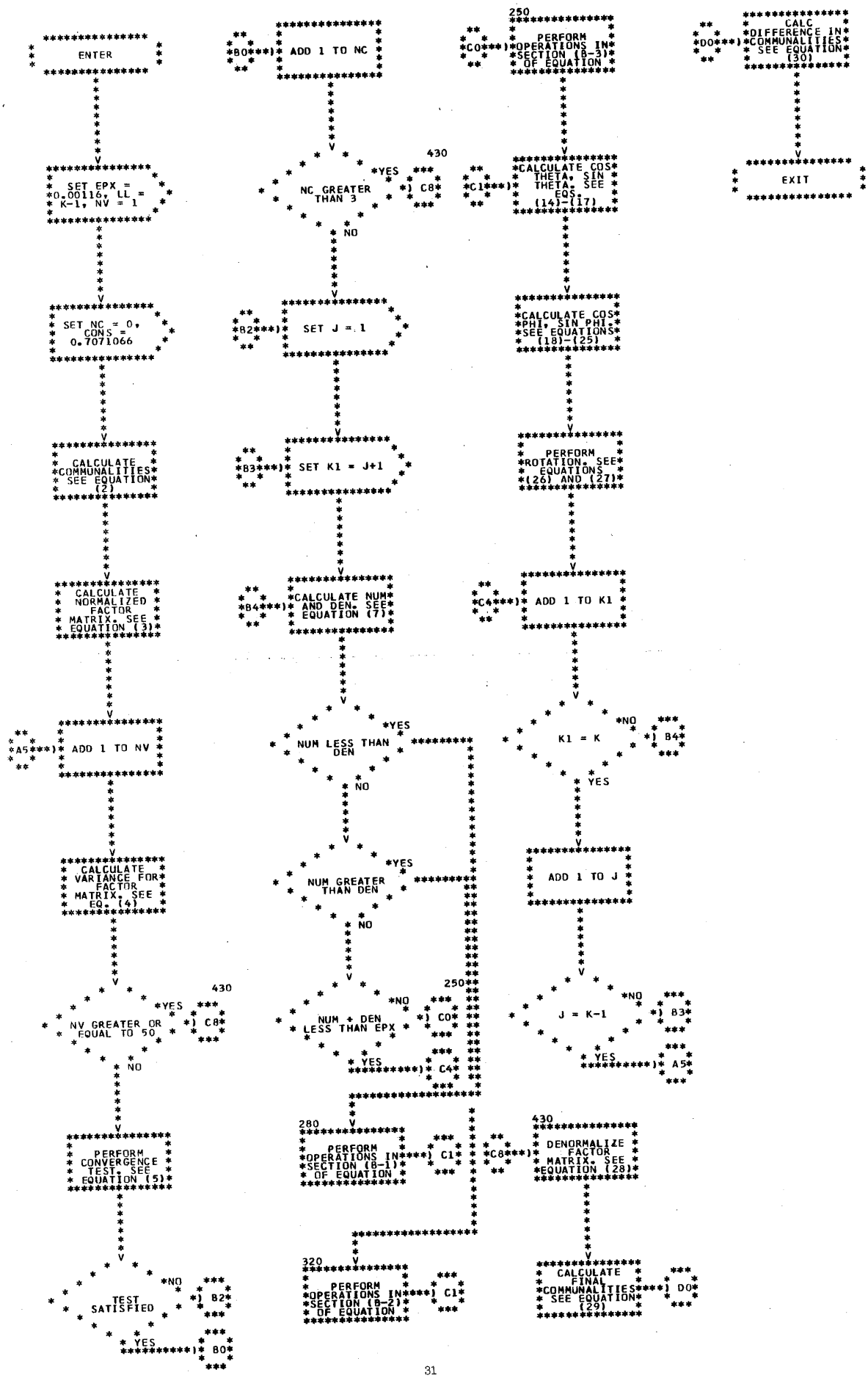

ENTER
CALCULATE COMMON MEANS FOR VARIABLES EQUATION (1)
CALCULATE GENERAL MAHALANOBIS D SQUARE. EQ.(2)
CALCULATE DEGREES OF FREEDOM
SET K = 1
CALC COEFFICIENTS OF DISCRINI-NANT FUNCTION EQ.(3)
CALCULATE CONSTANT EQUATION (4)
K = LAST FUNCTION
YES
NO
ADD 1 TO K
B0
A5
FOR EACH CASE IN EACH GROUP DO THE FOLLOWING STEPS
CALCULATE DISCRIMINANT FUNCTION EQUATION (5)
FIND THE LARGEST DISCRIMINANT FUNCTION
CALCULATE PROBALITY FOR LARGEST FUNCTION EQ.(6)
EXIT


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

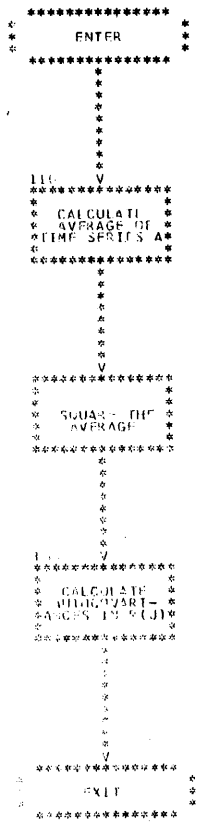
SUBROUTINE LOAD

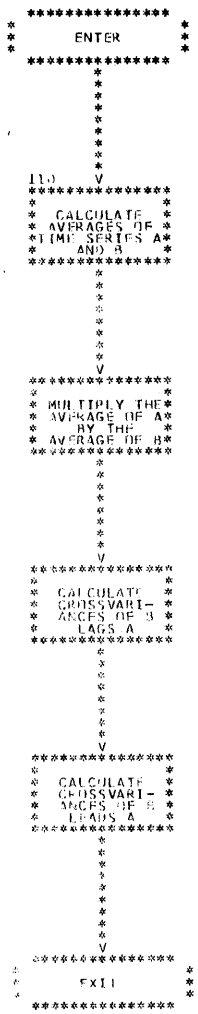


SUBROUTINE VARMX

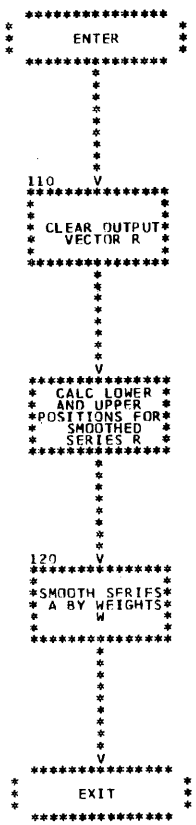


SUBROUTINE AUTO

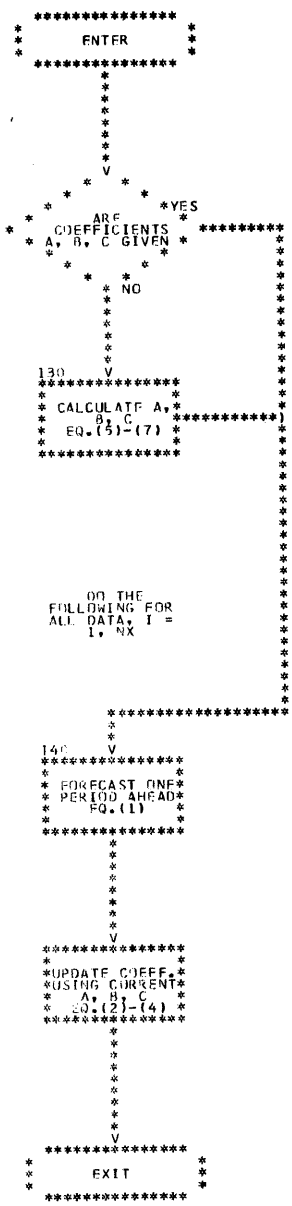


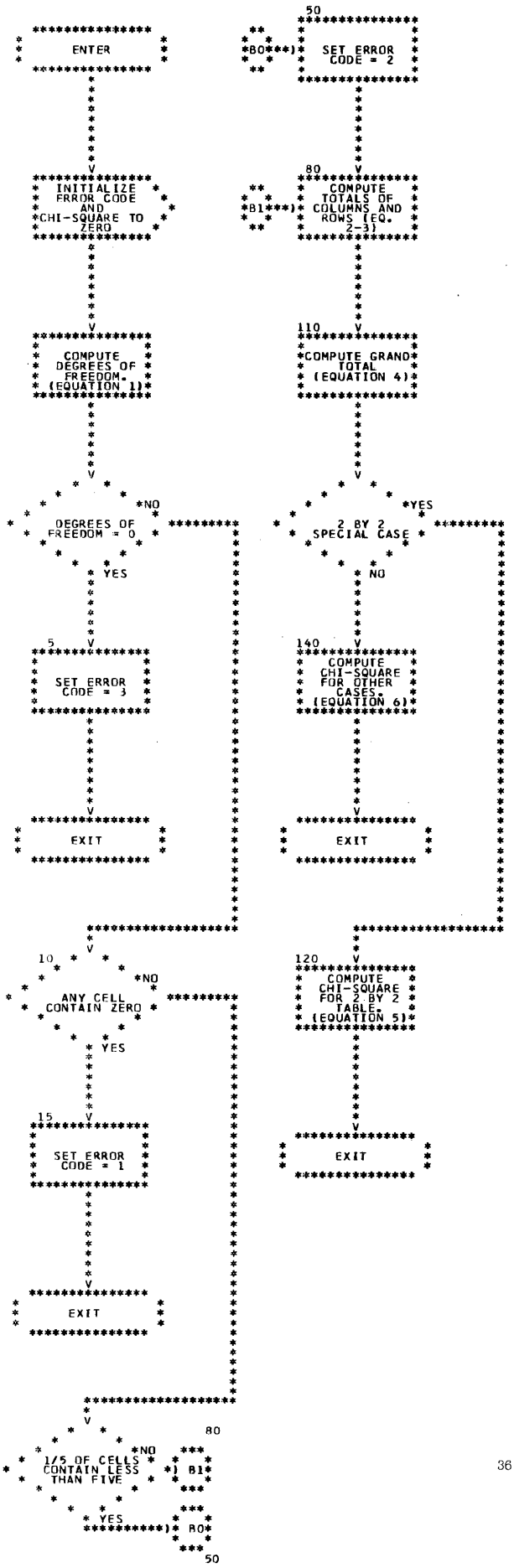

ENTER
CALCULATE AVERAGES OF TIME SERIES A AND B
MULTIPLY THE AVERAGE OF A BY THE AVERAGE OF B
CALCULATE CROSSVARIANCES OF B LAGS A
CALCULATE CROSSVARIANCES OF B LEADS A
EXIT

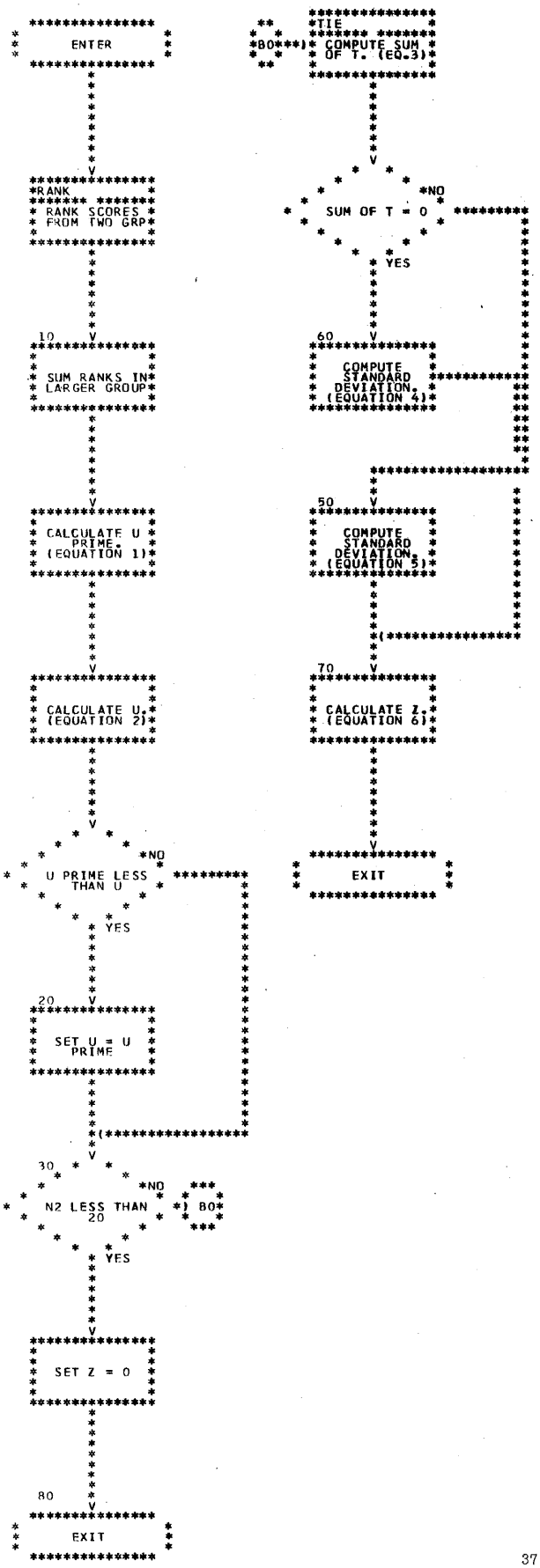
SUBROUTINE SMO



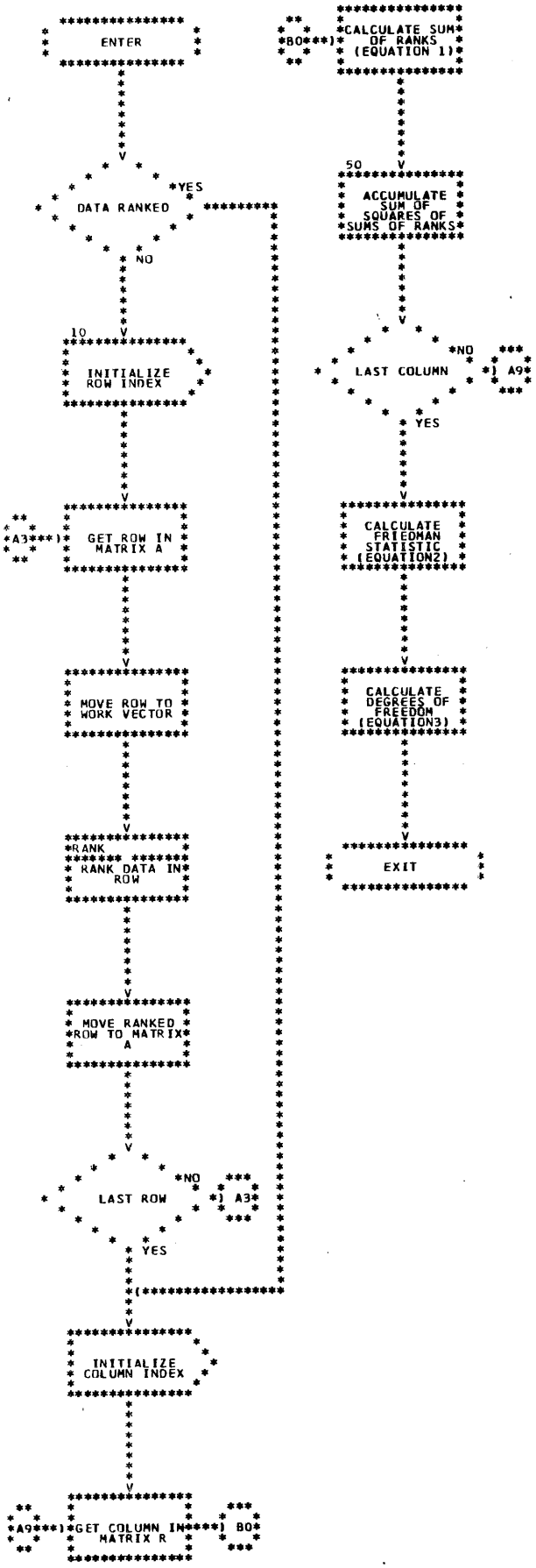
SUBROUTINE EXSMO



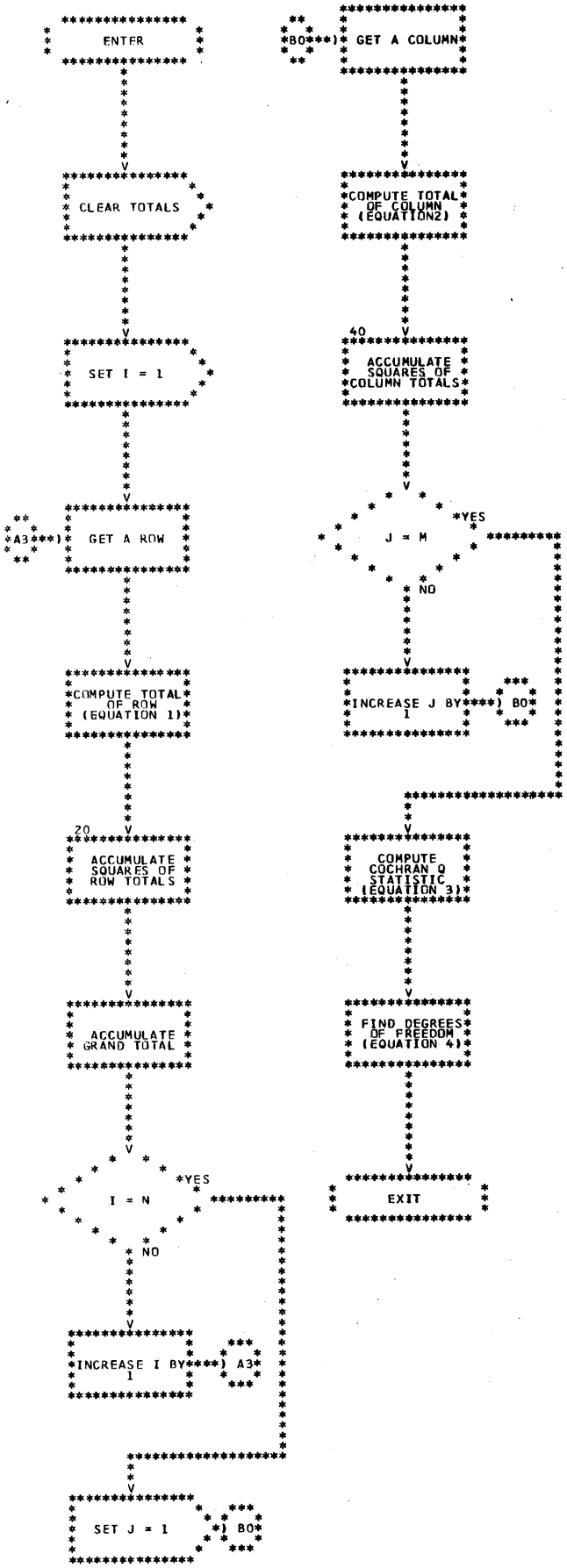

ENTER
INITIALIZE ERROR CODE AND CHI-SQUARE TO ZERO
COMPUTE DEGREES OF FREEDOM. (EQUATION 1)
DEGREES OF FREEDOM = 0
*NO
YES
5
SET ERROR CODE = 3
EXIT
ANY CELL CONTAIN ZERO
*NO
YES
15
SET ERROR CODE = 1
EXIT
1/5 OF CELLS CONTAIN LESS THAN FIVE
*NO
YES
80
*B1
*B0
50
SET ERROR CODE = 2
80
COMPUTE TOTALS OF COLUMNS AND ROWS (EQ. 2-3)
110
COMPUTE GRAND TOTAL (EQUATION 4)
2 BY 2 SPECIAL CASE
*YES
NO
140
COMPUTE CHI-SQUARE FOR OTHER CASES (EQUATION 6)
EXIT
120
COMPUTE CHI-SQUARE FOR 2 BY 2 TABLE. (EQUATION 5)
EXIT


ENTER
RANK
RANK SCORES FROM TWO GRP
10
SUM RANKS IN LARGER GROUP
CALCULATE U PRIME. (EQUATION 1)
CALCULATE U. (EQUATION 2)
U PRIME LESS THAN U
NO
YES
20
SET U = U PRIME
30
N2 LESS THAN 20
NO
80
EXIT
TIE
BO
COMPUTE SUM OF T. (EQ.3)
SUM OF T = 0
NO
YES
60
COMPUTE STANDARD DEVIATION. (EQUATION 4)
50
COMPUTE STANDARD DEVIATION. (EQUATION 5)
70
CALCULATE Z. (EQUATION 6)
EXIT

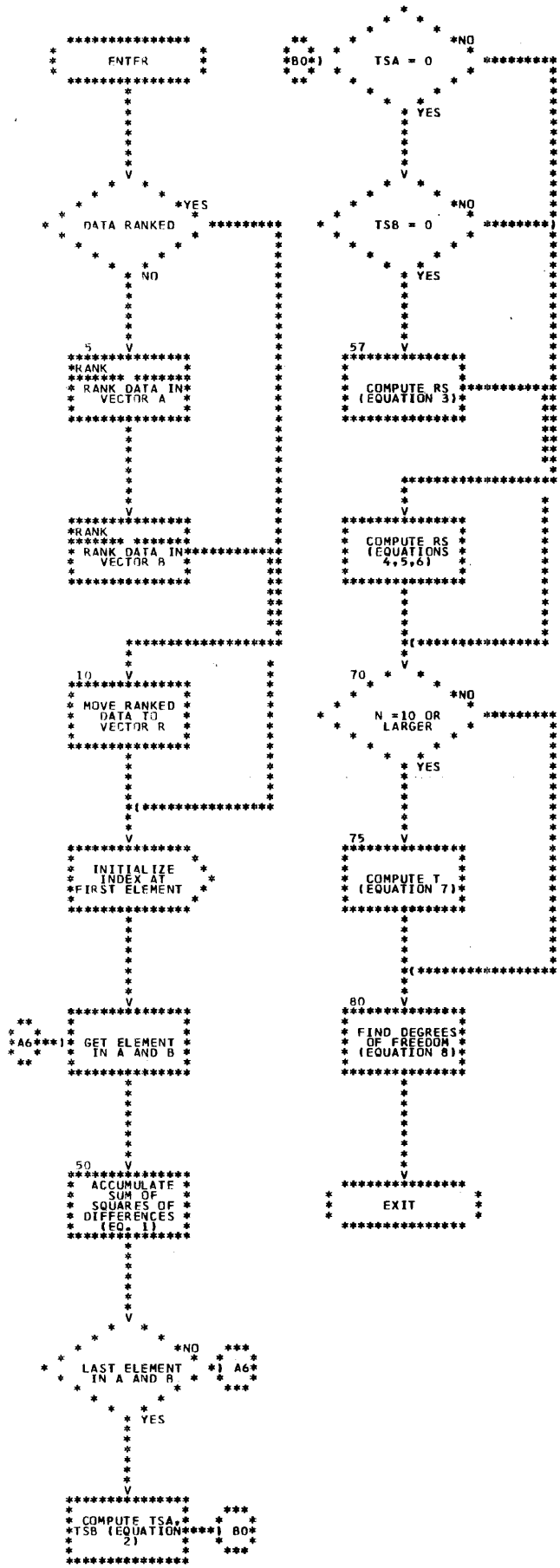
SUBROUTINE TWOAV



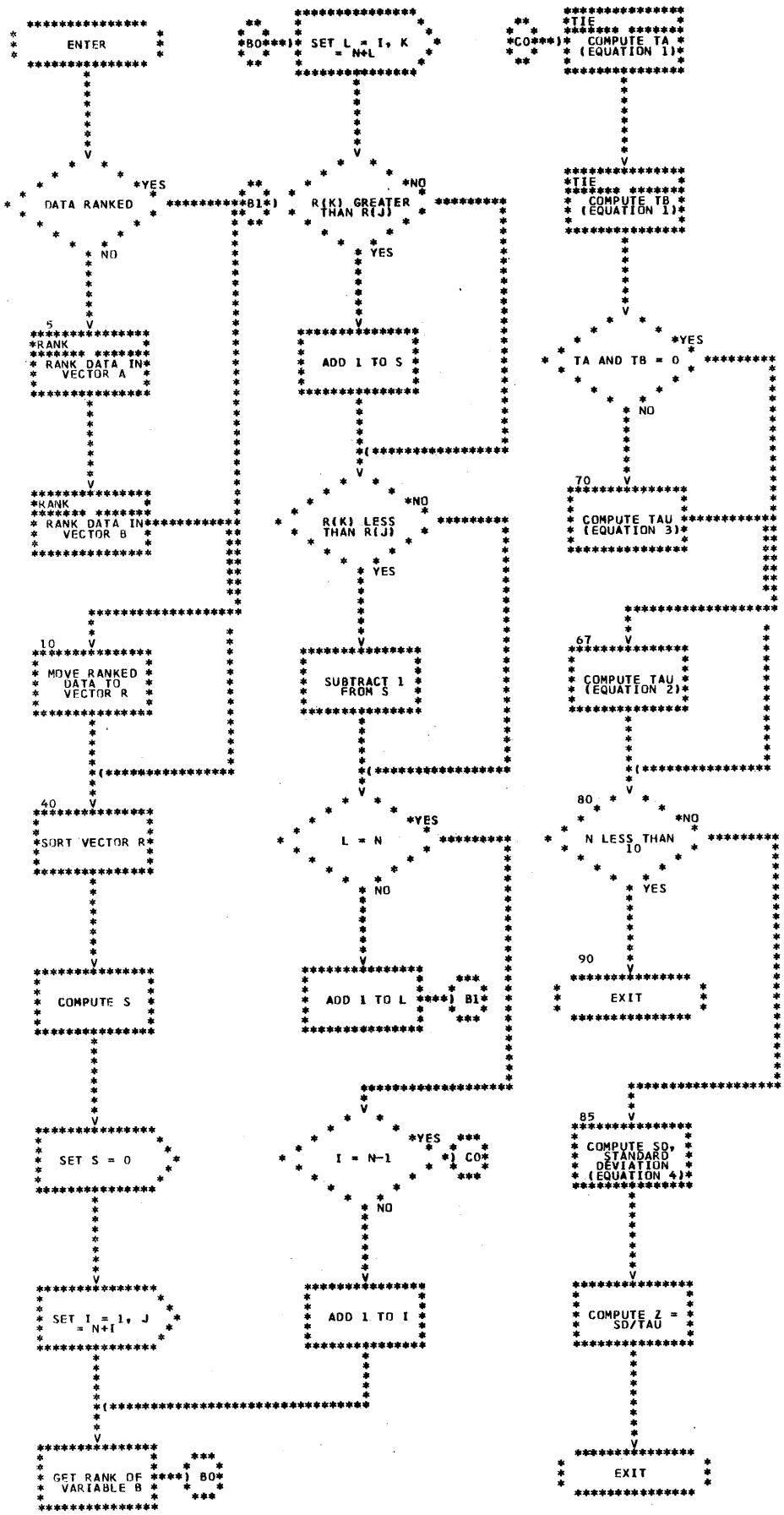
SUBROUTINE QTEST



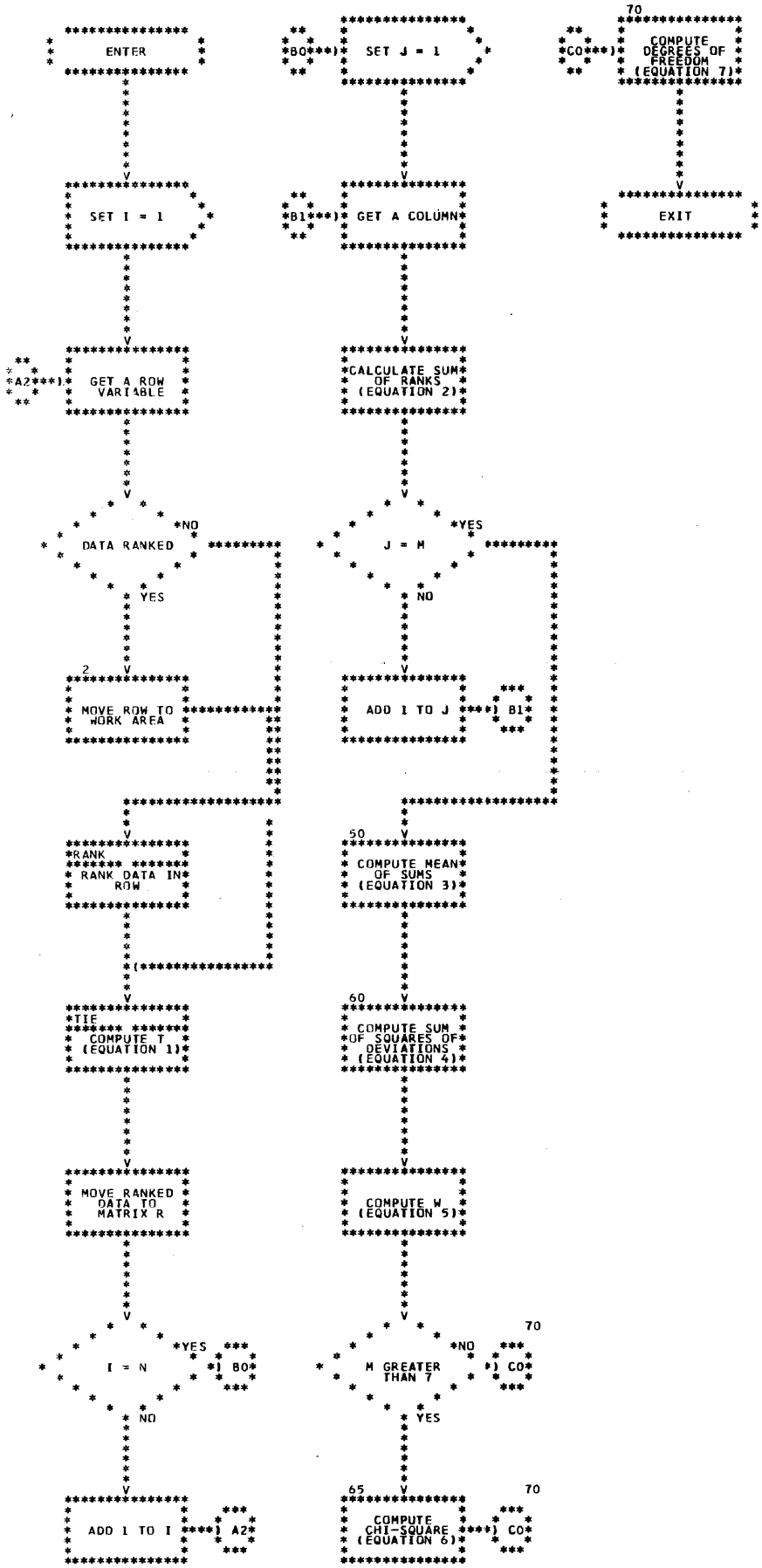
SUBROUTINE SRANK

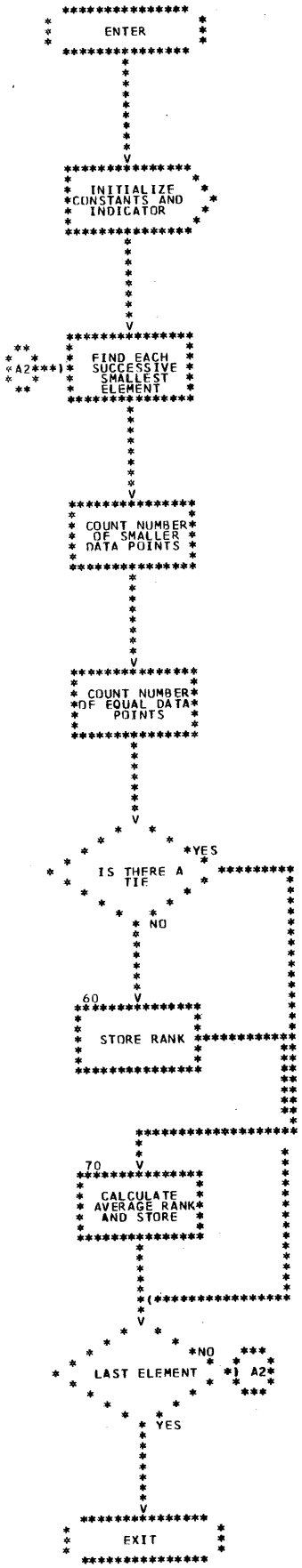


SUBROUTINE KRANK

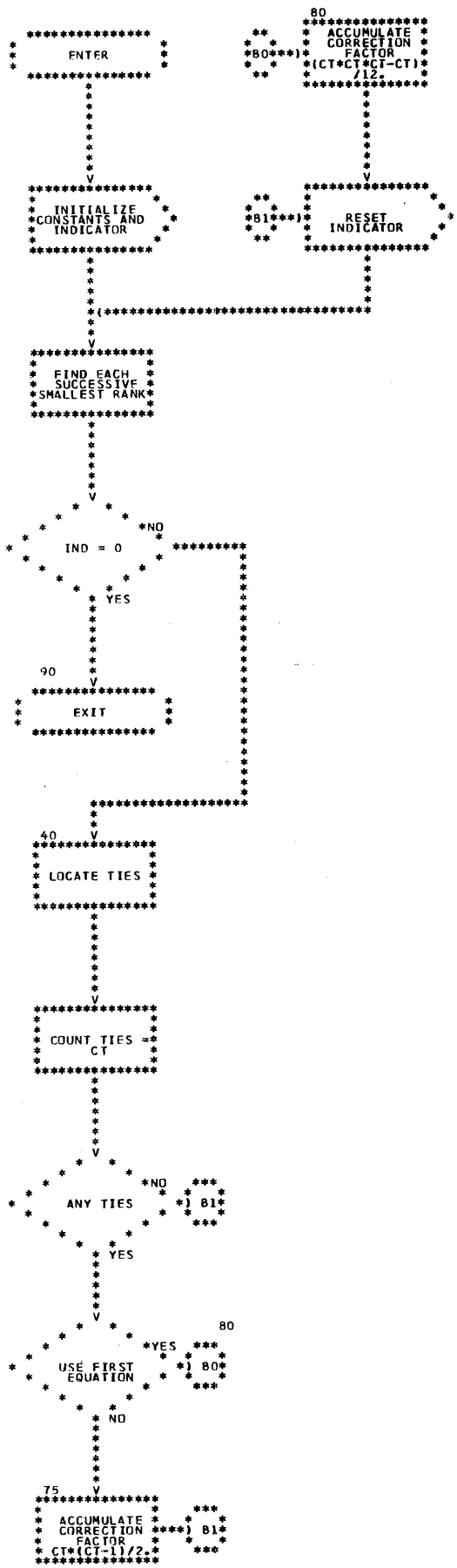


SUBROUTINE WTEST

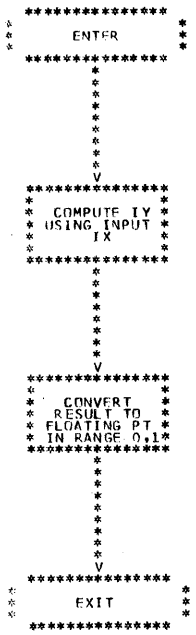



ENTER
INITIALIZE CONSTANTS AND INDICATOR
FIND EACH SUCCESSIVE SMALLEST ELEMENT
A2
COUNT NUMBER OF SMALLER DATA POINTS
COUNT NUMBER OF EQUAL DATA POINTS
IS THERE A TIE
YES
NO
60
STORE RANK
70
CALCULATE AVERAGE RANK AND STORE
LAST ELEMENT
NO
A2
YES
EXIT

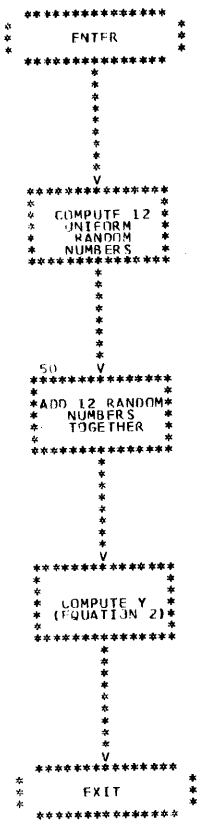
SUBROUTINE TIE



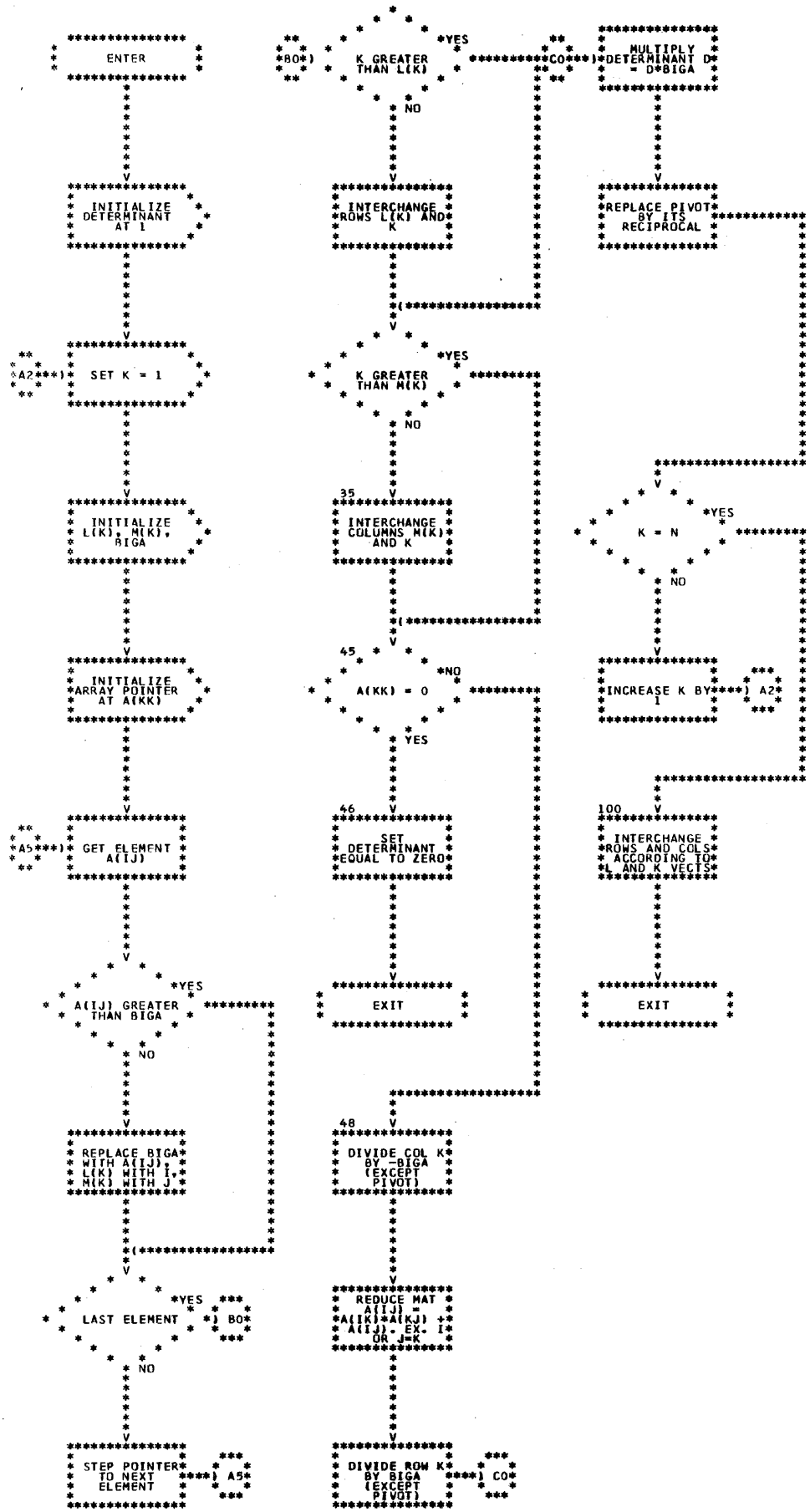
SUBROUTINE RANDU



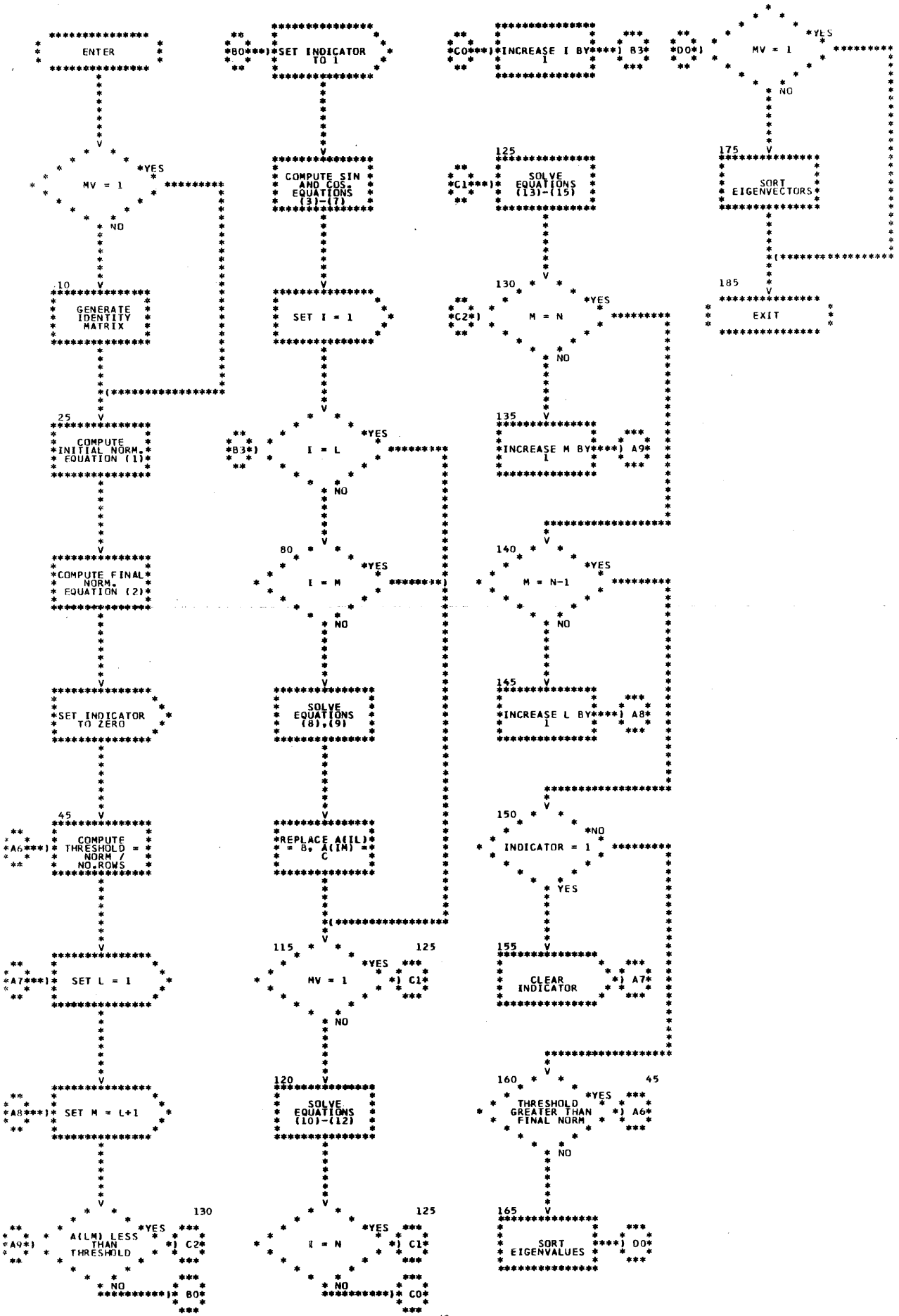
SUBROUTINE GAUSS

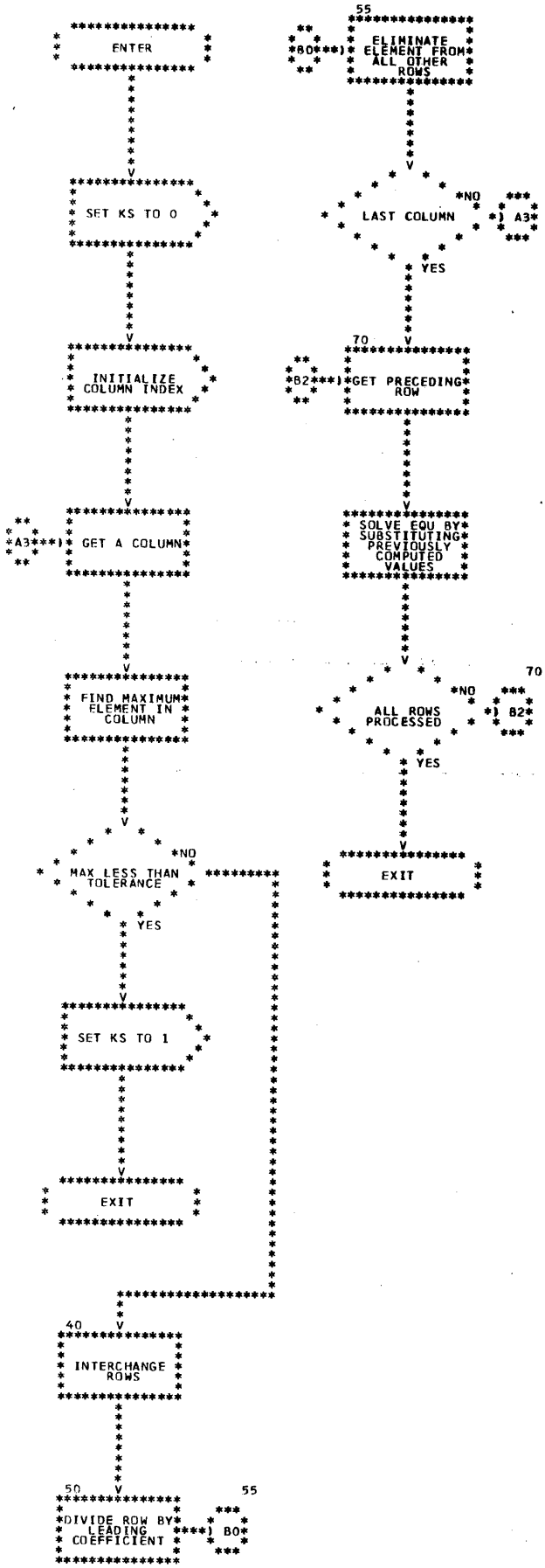


SUBROUTINE MINV

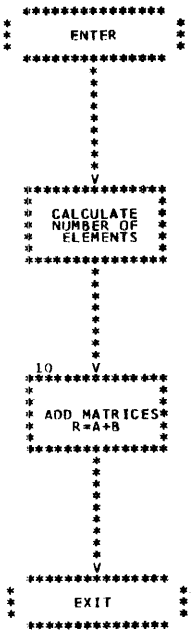


SUBROUTINE EIGEN

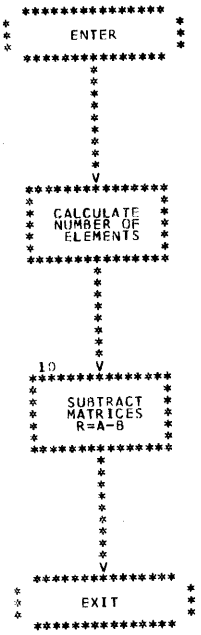


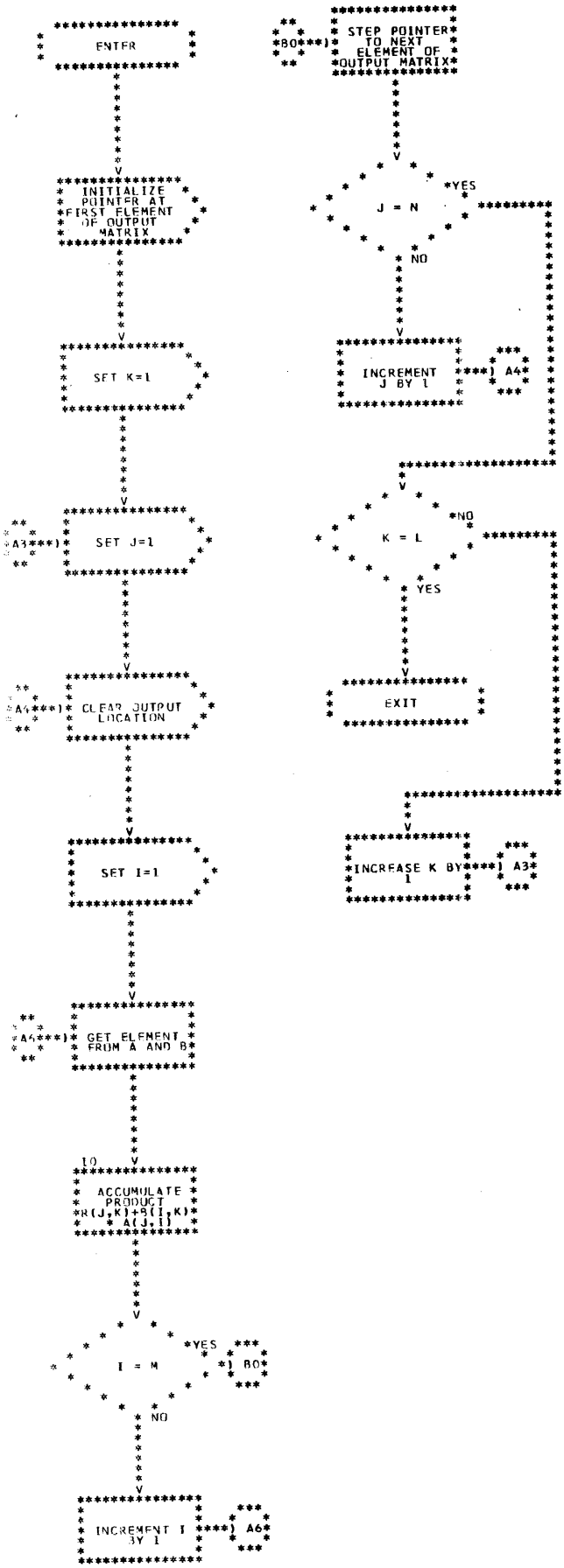

ENTER
SET KS TO 0
INITIALIZE COLUMN INDEX
GET A COLUMN
FIND MAXIMUM ELEMENT IN COLUMN
MAX LESS THAN TOLERANCE
NO
YES
SET KS TO 1
EXIT
40
INTERCHANGE ROWS
50
DIVIDE ROW BY LEADING COEFFICIENT
55
B0
ELIMINATE ELEMENT FROM ALL OTHER ROWS
LAST COLUMN
NO
A3
YES
70
B2
GET PRECEDING ROW
SOLVE EQU BY SUBSTITUTING PREVIOUSLY COMPUTED VALUES
ALL ROWS PROCESSED
NO
70
B2
YES
EXIT

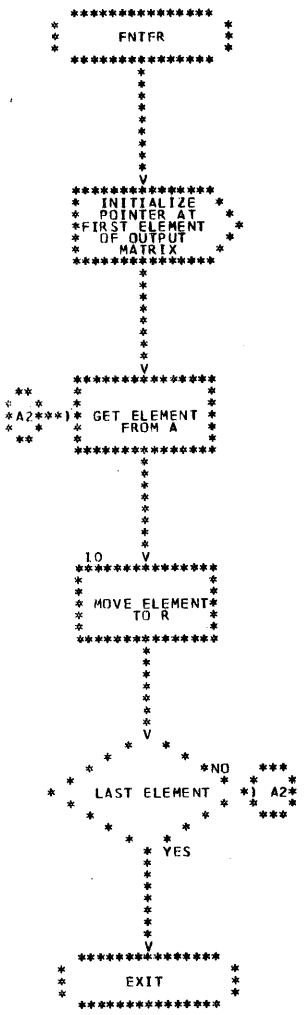
SUBROUTINE GMADD

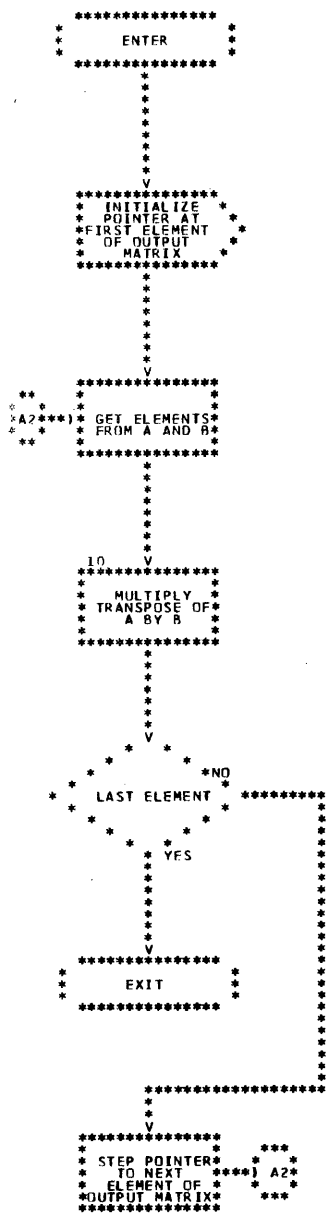


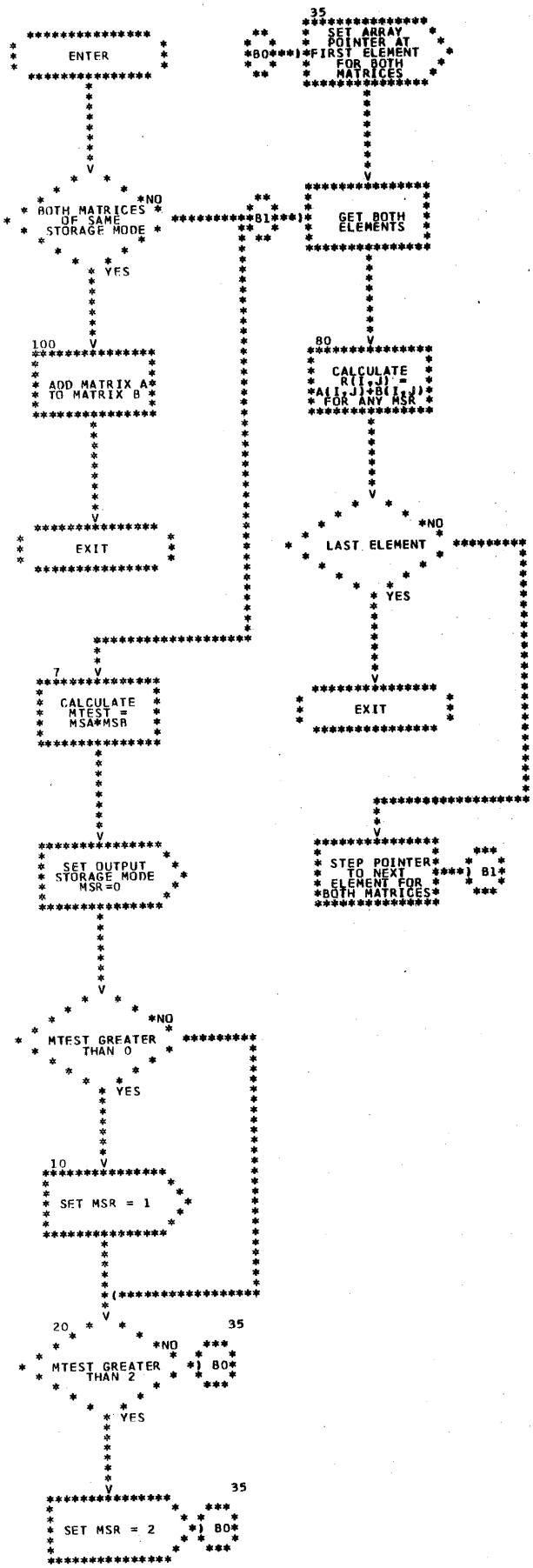
SUBROUTINE GMSUB

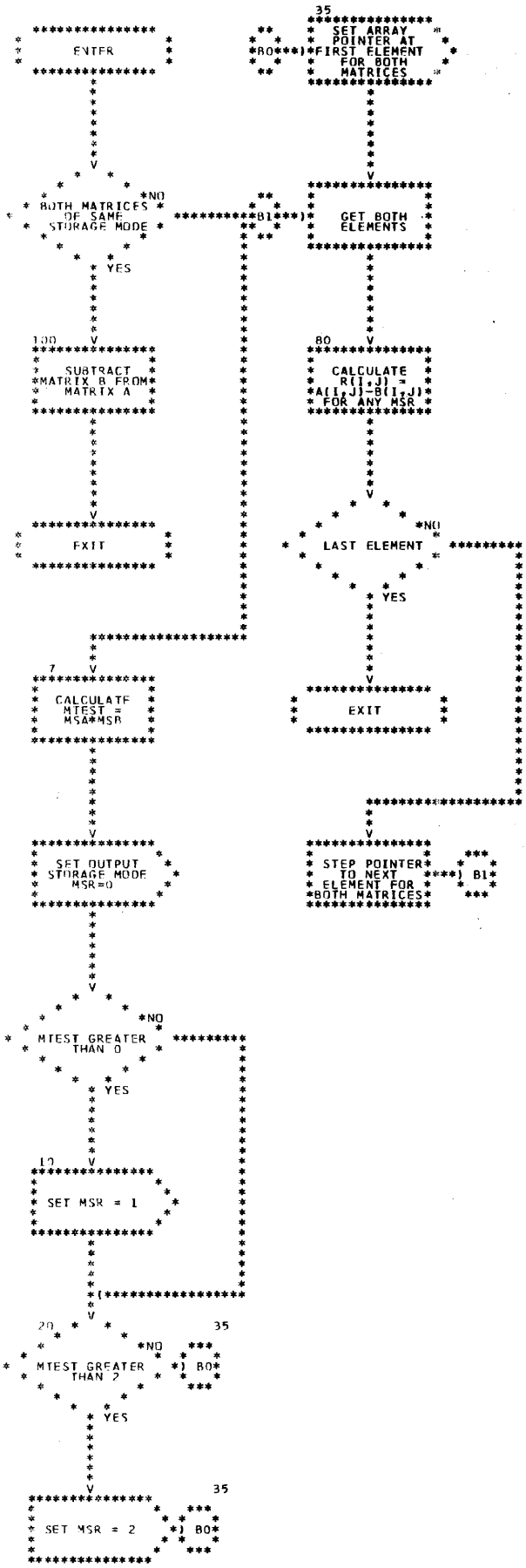


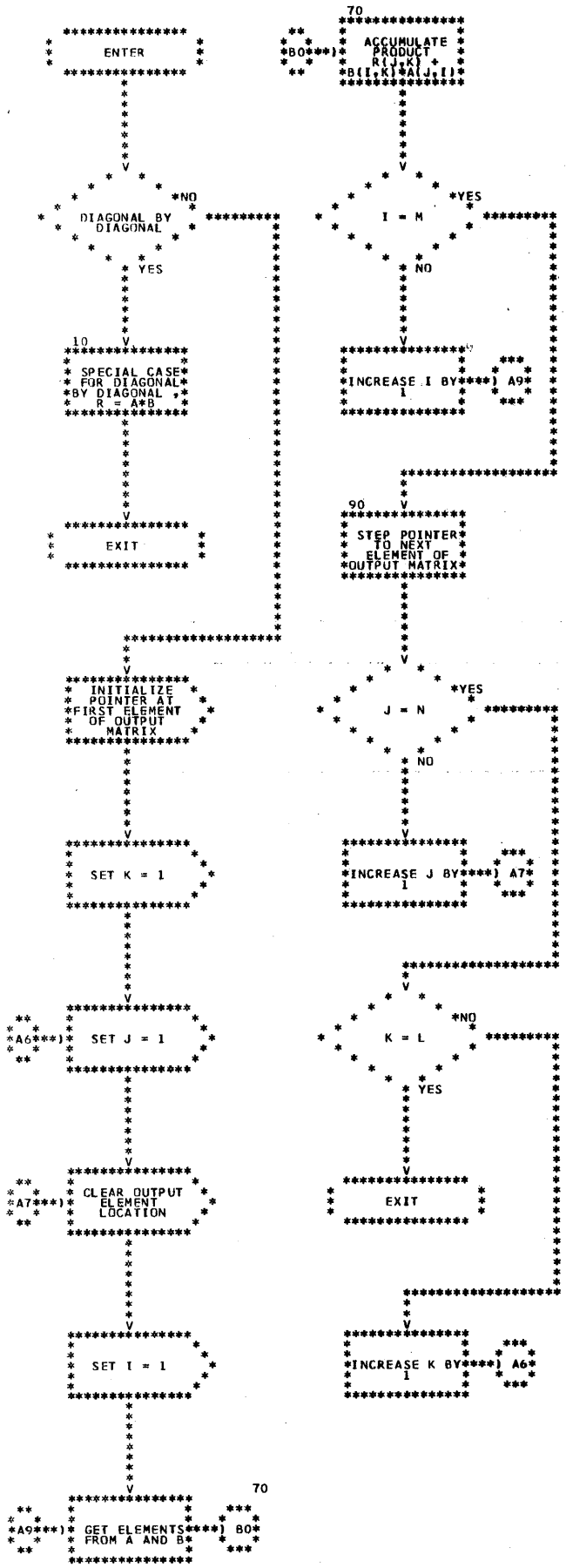

ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF OUTPUT MATRIX
SET K=1
SET J=1
CLEAR OUTPUT LOCATION
SET I=1
GET ELEMENT FROM A AND B
ACCUMULATE PRODUCT R(J,K)+B(I,K)*A(J,I)
I = M
YES
NO
INCREMENT I BY 1
A6
B0
STEP POINTER TO NEXT ELEMENT OF OUTPUT MATRIX
J = N
YES
NO
INCREMENT J BY 1
A4
K = L
NO
YES
EXIT
INCREASE K BY I
A3

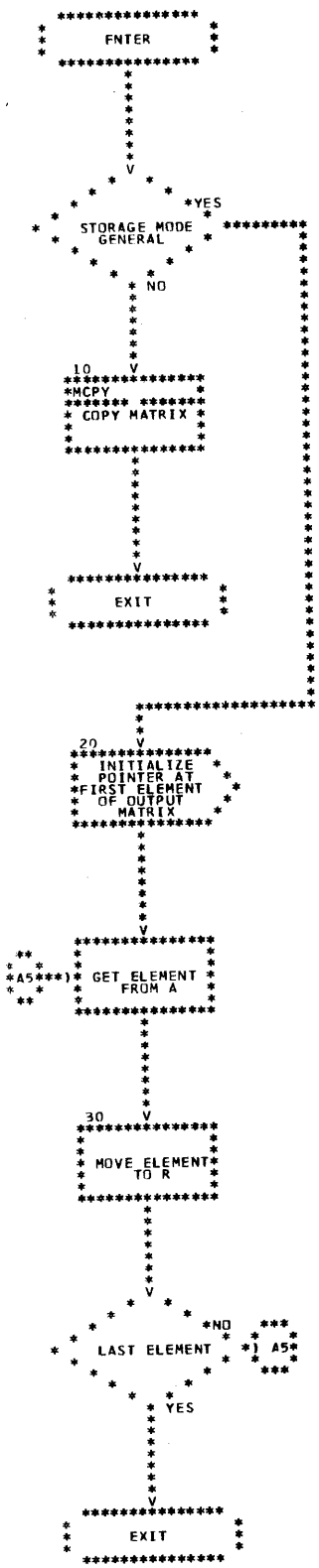

ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF OUTPUT MATRIX
GET ELEMENT FROM A
A2
IO
MOVE ELEMENT TO R
NO
LAST ELEMENT
A2
YES
EXIT


ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF OUTPUT MATRIX
A2
GET ELEMENTS FROM A AND B
10
MULTIPLY TRANSPOSE OF A BY B
NO
LAST ELEMENT
YES
EXIT
STEP POINTER TO NEXT ELEMENT OF OUTPUT MATRIX
A2

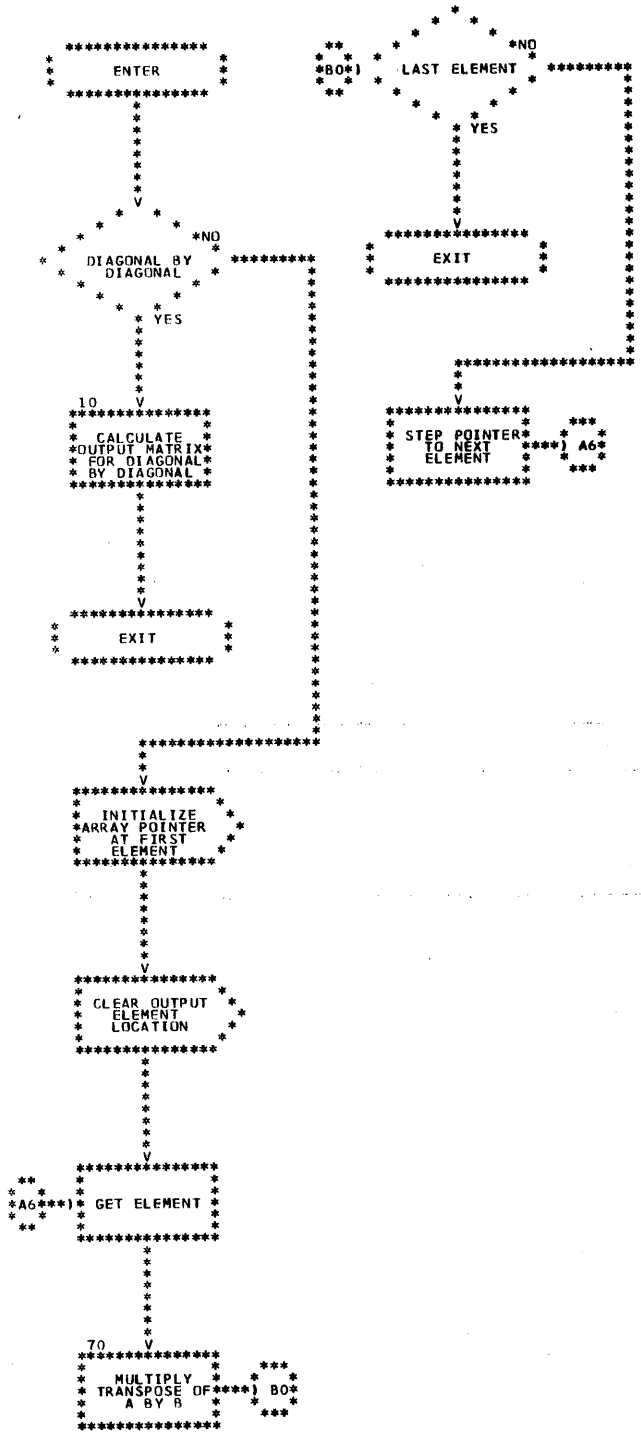

ENTER
BOTH MATRICES OF SAME STORAGE MODE
NO
YES
B1
35
SET ARRAY POINTER AT FIRST ELEMENT FOR BOTH MATRICES
B0
GET BOTH ELEMENTS
80
CALCULATE R[I,J] = A[I,J]+B[I,J] FOR ANY MSR
LAST ELEMENT
NO
YES
EXIT
STEP POINTER TO NEXT ELEMENT FOR BOTH MATRICES
B1
100
ADD MATRIX A TO MATRIX B
EXIT
7
CALCULATE MTEST = MSA*MSB
SET OUTPUT STORAGE MODE MSR=0
MTEST GREATER THAN 0
NO
YES
10
SET MSR = 1
20
MTEST GREATER THAN 2
NO
B0
YES
35
SET MSR = 2
B0

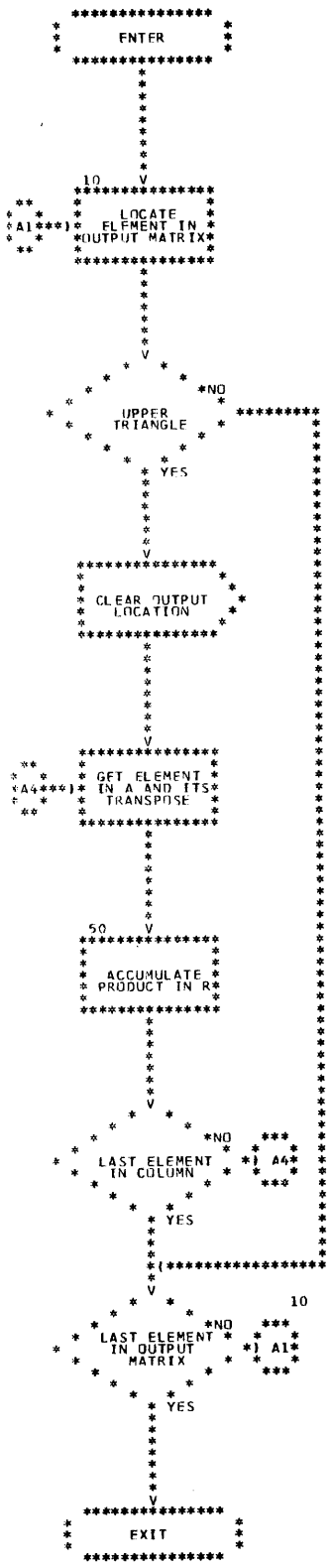

ENTER
35
SET ARRAY POINTER AT FIRST ELEMENT FOR BOTH MATRICES
BOTH MATRICES OF SAME STORAGE MODE
NO
B1
GET BOTH ELEMENTS
YES
100
SUBTRACT MATRIX B FROM MATRIX A
80
CALCULATE R[I,J] = A[I,J]-B[I,J] FOR ANY MSR
LAST ELEMENT
NO
YES
EXIT
7
CALCULATE MTEST = MSA*MSR
EXIT
STEP POINTER TO NEXT ELEMENT FOR BOTH MATRICES
B1
SET OUTPUT STORAGE MODE MSR=0
MTEST GREATER THAN 0
NO
YES
10
SET MSR = 1
20
MTEST GREATER THAN 2
NO
B0
YES
35
SET MSR = 2


ENTER
DIAGONAL BY DIAGONAL
NO
YES
10
SPECIAL CASE FOR DIAGONAL BY DIAGONAL , R = A*B
EXIT
INITIALIZE POINTER AT FIRST ELEMENT OF OUTPUT MATRIX
SET K = 1
SET J = 1
CLEAR OUTPUT ELEMENT LOCATION
SET I = 1
70
GET ELEMENTS FROM A AND B
B0
A6
A7
A9
70
ACCUMULATE PRODUCT R(J,K) + B(I,K)*A(J,I)
I = M
YES
NO
INCREASE I BY 1
A9
90
STEP POINTER TO NEXT ELEMENT OF OUTPUT MATRIX
J = N
YES
NO
INCREASE J BY 1
A7
K = L
NO
YES
EXIT
INCREASE K BY 1
A6

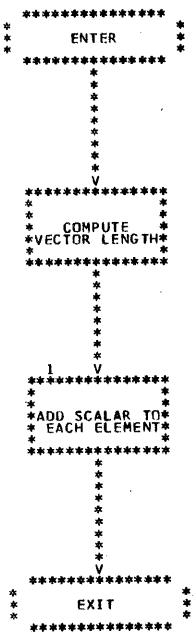

ENTER
STORAGE MODE GENERAL
YES
NO
10
MCPY
COPY MATRIX
EXIT
20
INITIALIZE POINTER AT FIRST ELEMENT OF OUTPUT MATRIX
GET ELEMENT FROM A
A5
30
MOVE ELEMENT TO R
LAST ELEMENT
NO
A5
YES
EXIT

SUBROUTINE TPRD

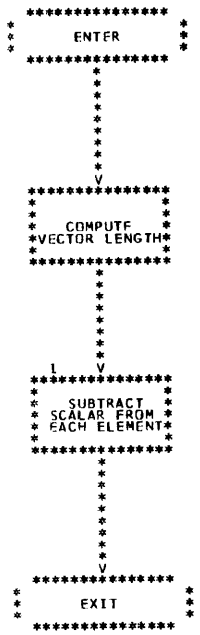



ENTER
10
LOCATE ELEMENT IN OUTPUT MATRIX
A1
UPPER TRIANGLE
NO
YES
CLEAR OUTPUT LOCATION
GET ELEMENT IN A AND ITS TRANSPOSE
A4
50
ACCUMULATE PRODUCT IN R
LAST ELEMENT IN COLUMN
NO
A4
YES
10
LAST ELEMENT IN OUTPUT MATRIX
NO
A1
YES
EXIT

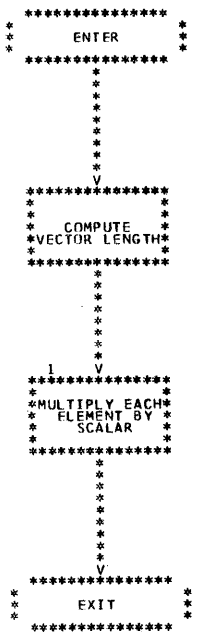
SUBROUTINE SADD



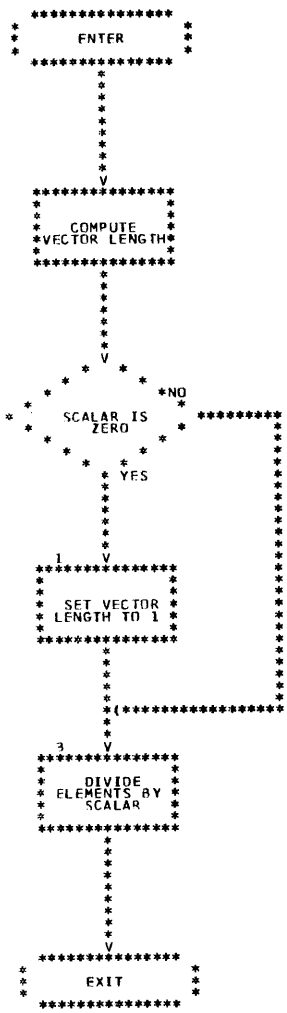
SUBROUTINE SSUB

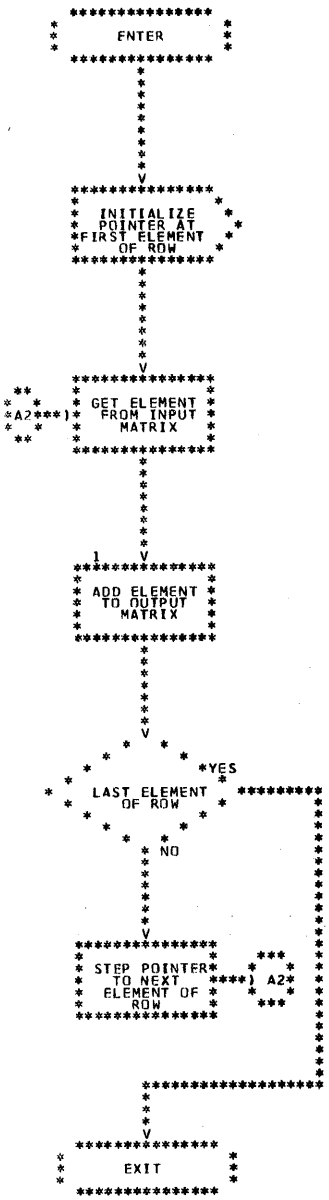


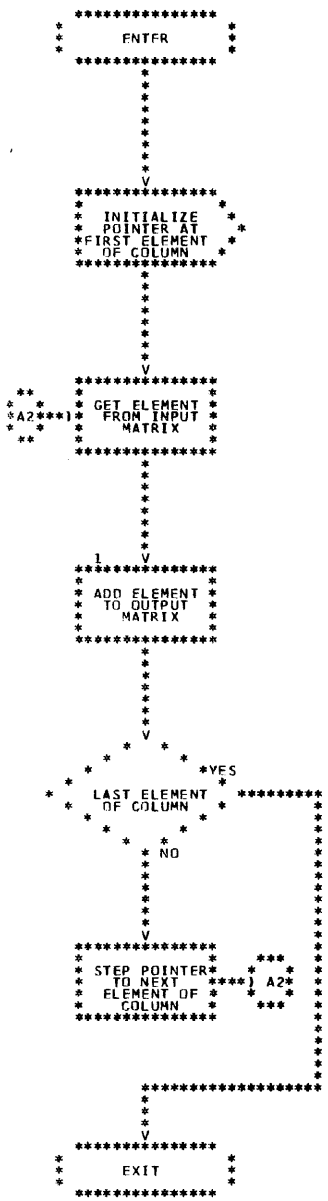
SUBROUTINE SMPY

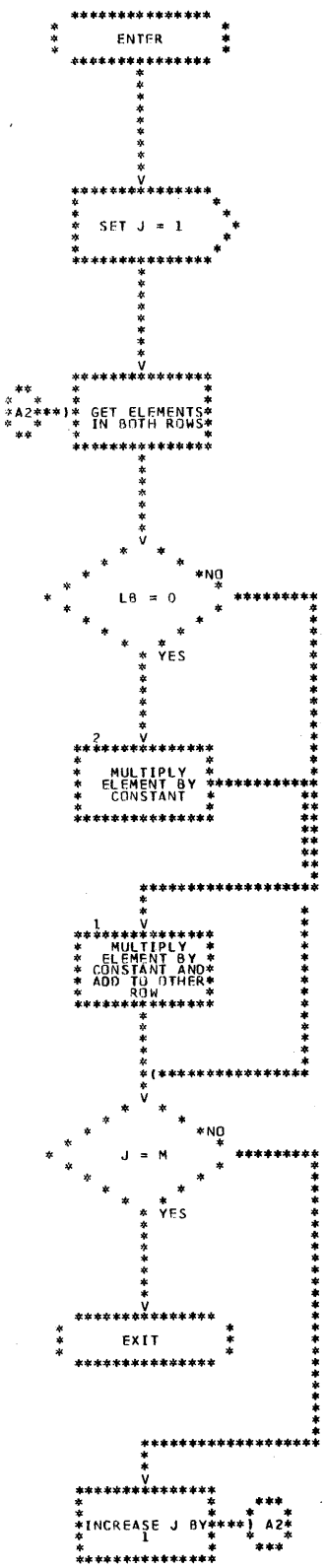


SUBROUTINE SDIV

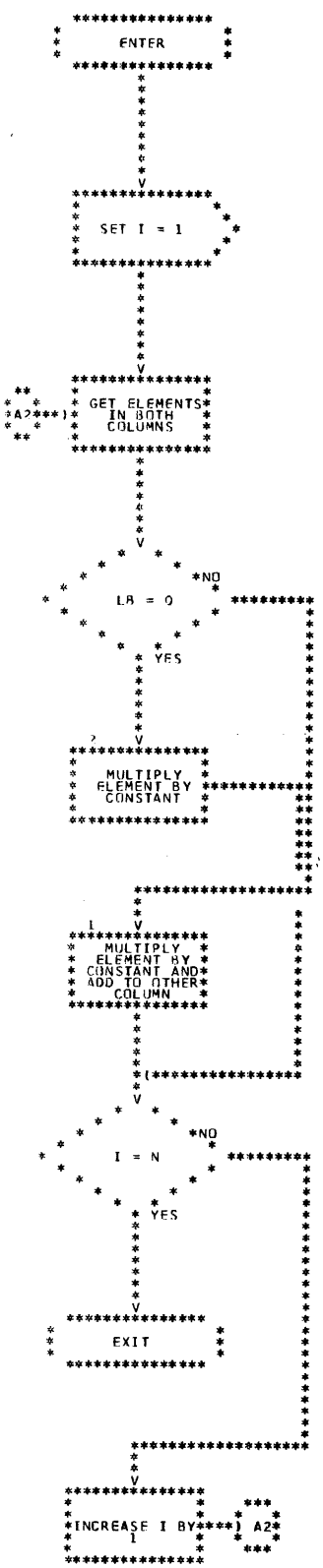



ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF ROW
GET ELEMENT FROM INPUT MATRIX
A2
ADD ELEMENT TO OUTPUT MATRIX
LAST ELEMENT OF ROW
YES
NO
STEP POINTER TO NEXT ELEMENT OF ROW
A2
EXIT

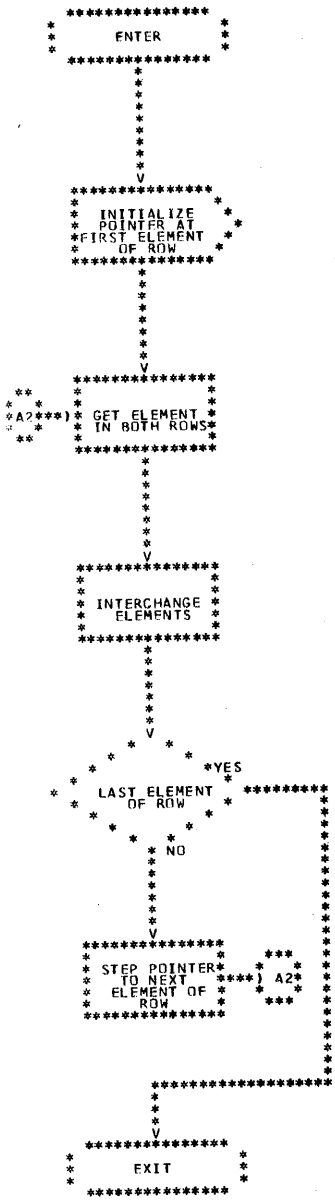

ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF COLUMN
GET ELEMENT FROM INPUT MATRIX
A2
1
ADD ELEMENT TO OUTPUT MATRIX
YES
LAST ELEMENT OF COLUMN
NO
STEP POINTER TO NEXT ELEMENT OF COLUMN
A2
EXIT


ENTER
SET J = 1
GET ELEMENTS IN BOTH ROWS
A2
LB = 0
NO
YES
2
MULTIPLY ELEMENT BY CONSTANT
1
MULTIPLY ELEMENT BY CONSTANT AND ADD TO OTHER ROW
J = M
NO
YES
EXIT
INCREASE J BY 1
A2

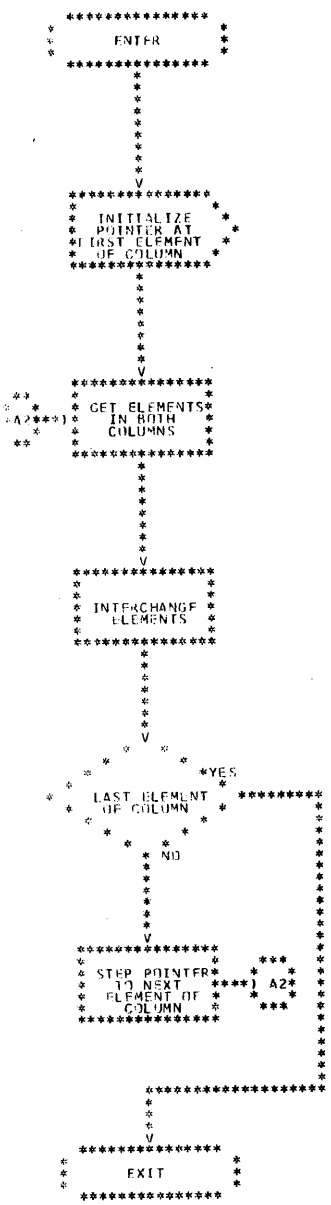
SUBROUTINE SCHA



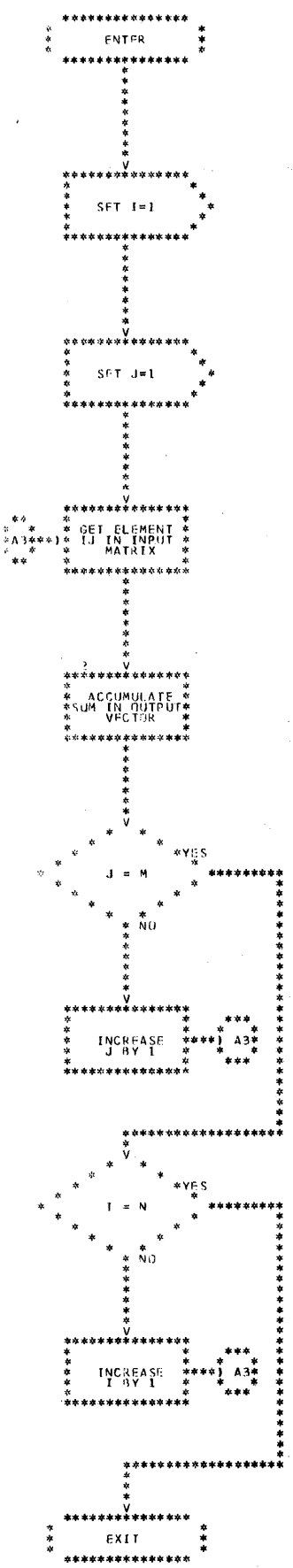
SUBROUTINE RINT

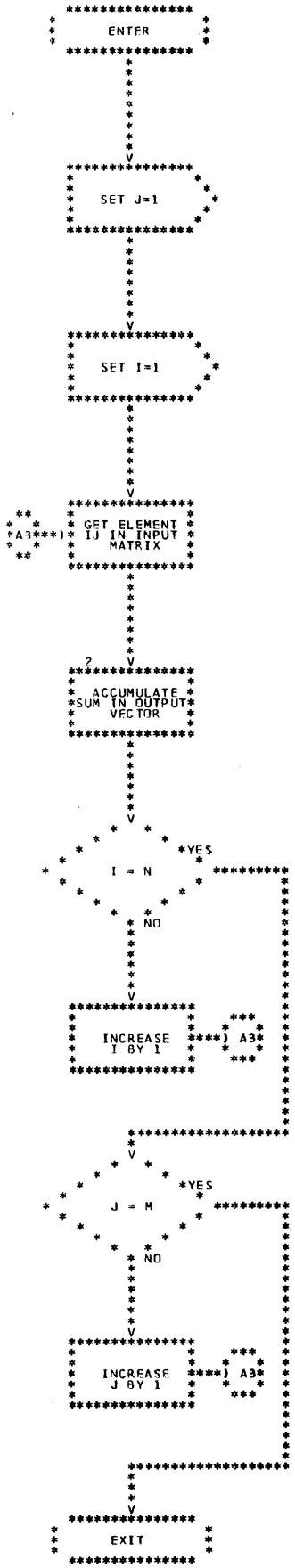


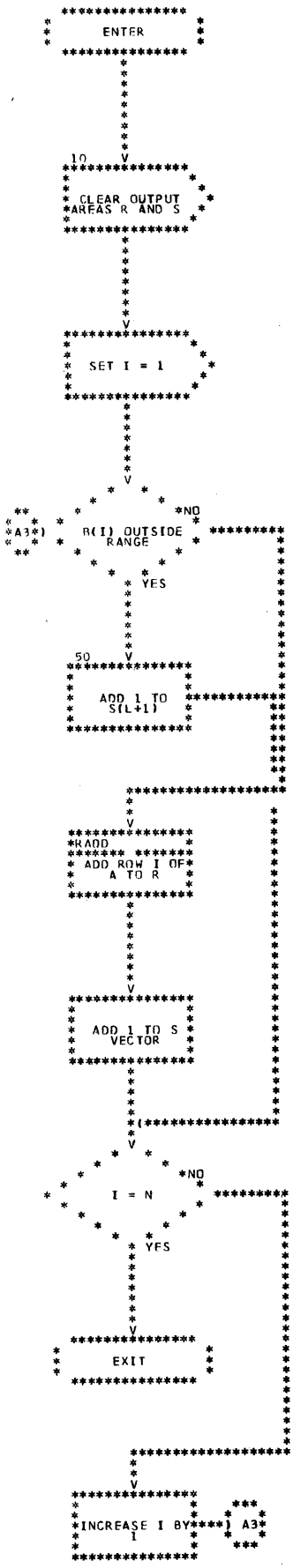
SUBROUTINE CINT

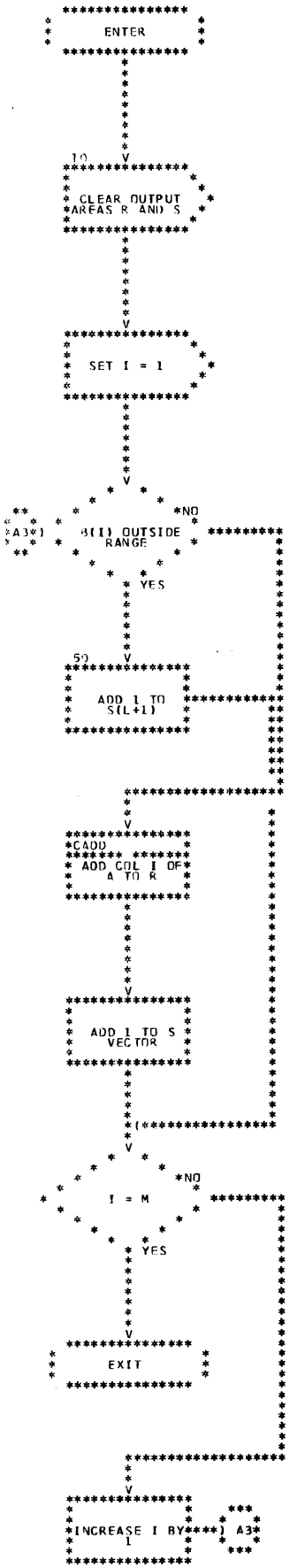


SUBROUTINE RSUM

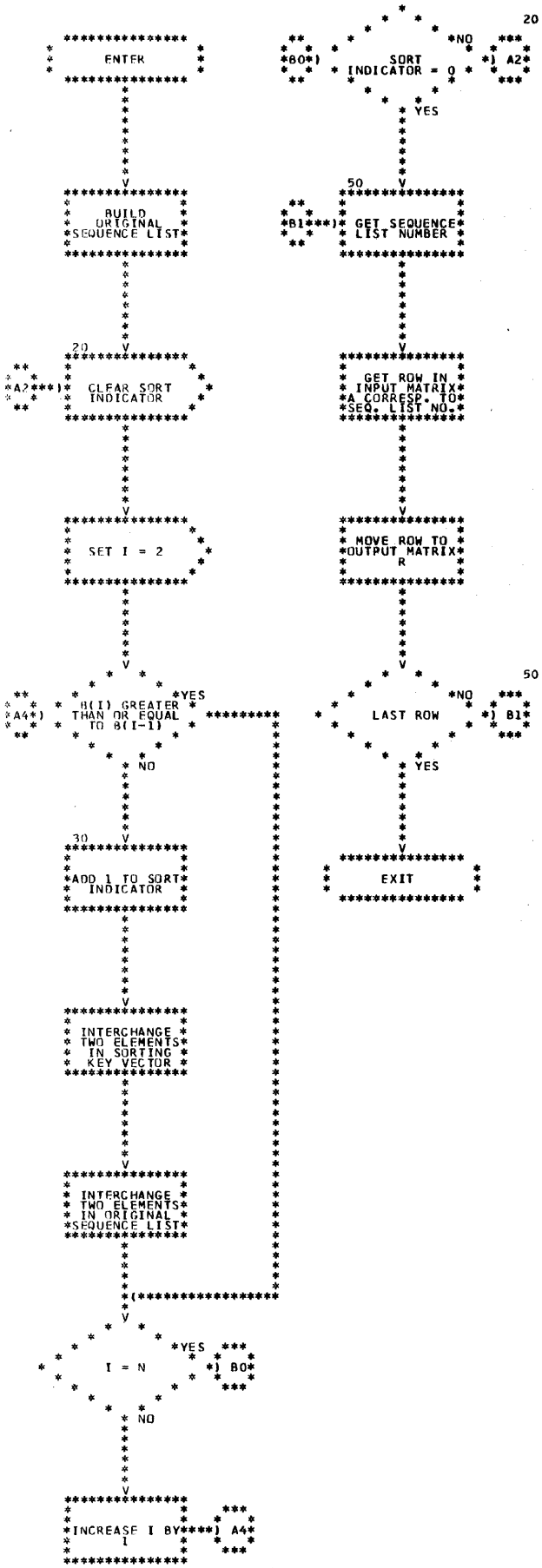



ENTER
SET J=1
SET I=1
GET ELEMENT IJ IN INPUT MATRIX
A3
2
ACCUMULATE SUM IN OUTPUT VECTOR
I = N
YES
NO
INCREASE I BY 1
A3
J = M
YES
NO
INCREASE J BY 1
A3
EXIT

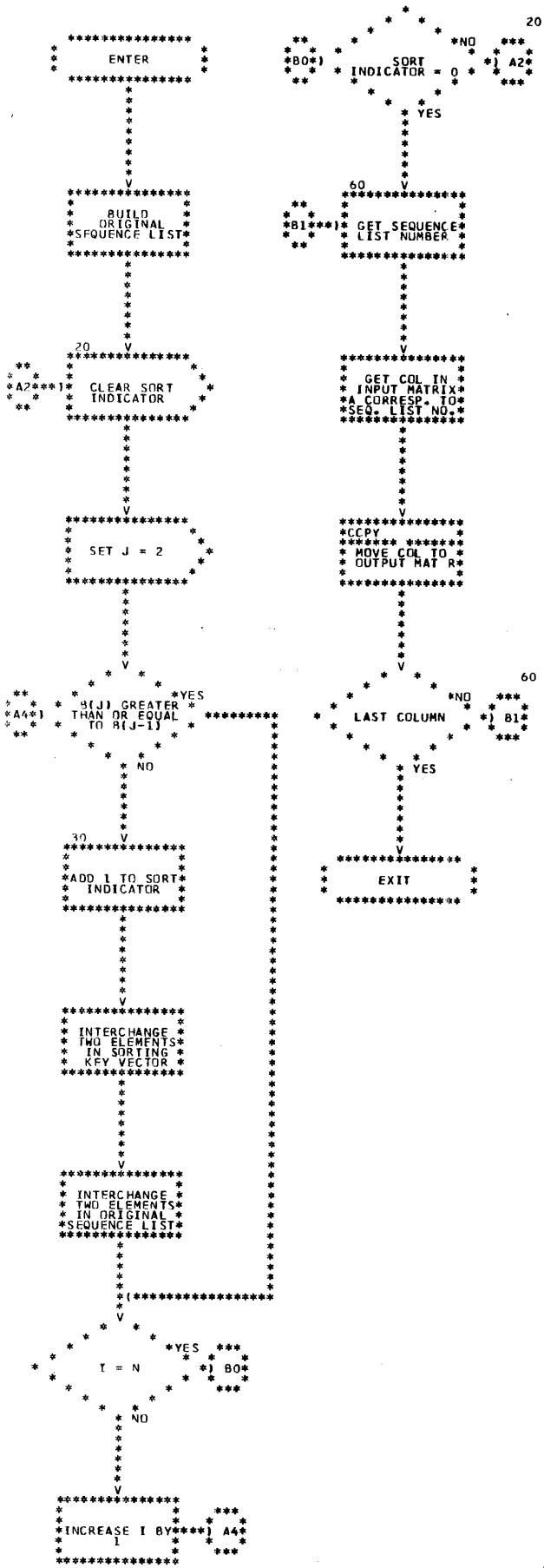

ENTER
10
CLEAR OUTPUT AREAS R AND S
SET I = 1
A3
B(I) OUTSIDE RANGE
NO
YES
50
ADD 1 TO S(L+1)
RADD
ADD ROW I OF A TO R
ADD 1 TO S VECTOR
I = N
NO
YES
EXIT
INCREASE I BY 1
A3


ENTER
TO
CLEAR OUTPUT AREAS R AND S
SET I = 1
NO
B(I) OUTSIDE RANGE
A3
YES
GO
ADD 1 TO S(L+1)
CADD
ADD COL I OF A TO R
ADD 1 TO S VECTOR
I = M
NO
YES
EXIT
INCREASE I BY 1
A3

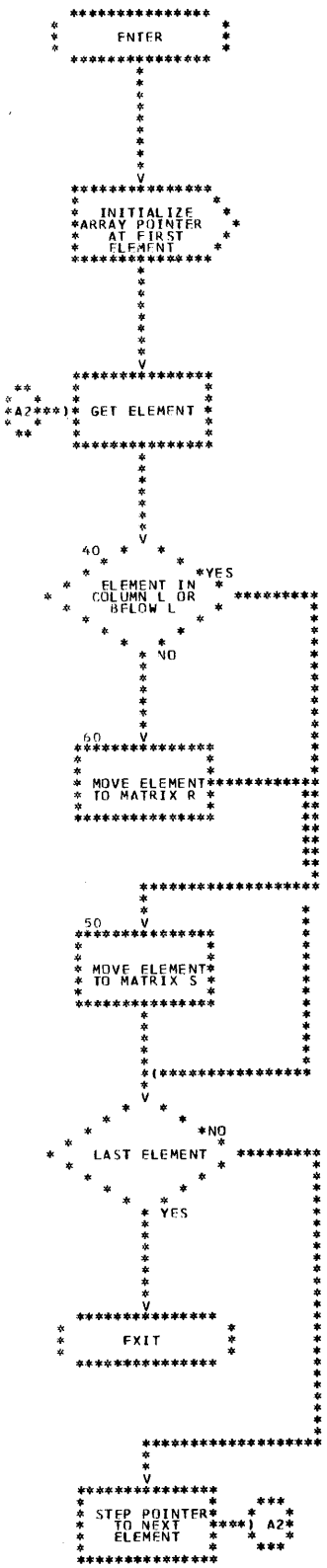
SUBROUTINE RSRT



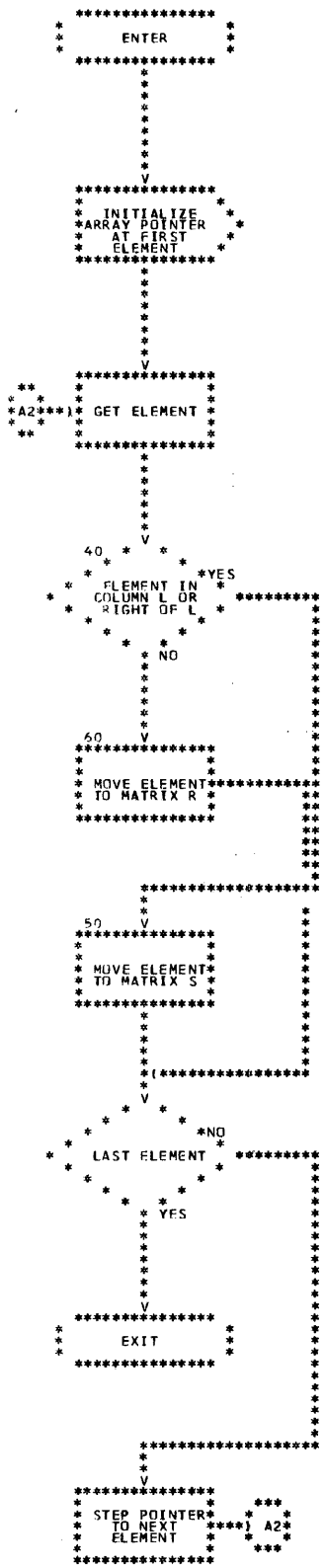
SUBROUTINE CSRT



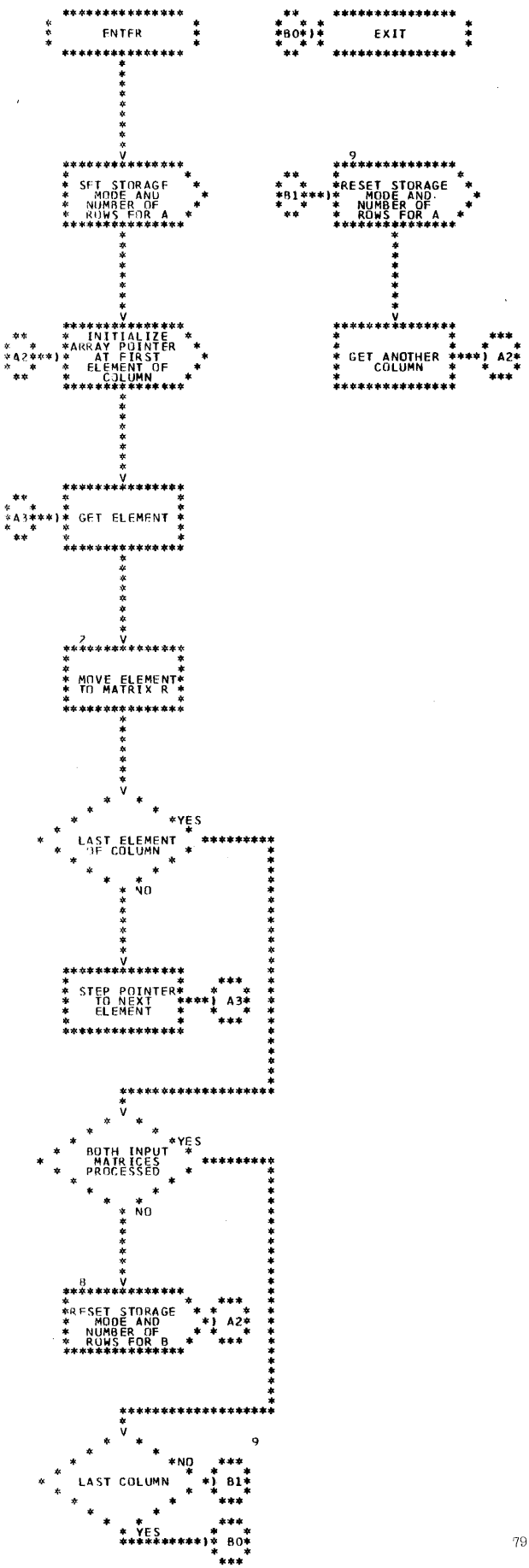
SUBROUTINE RCUT

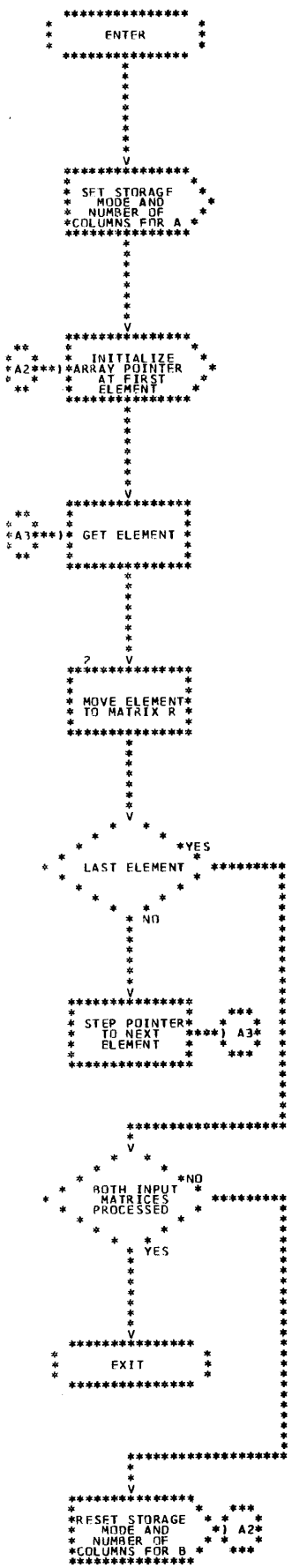


SUBROUTINE CCUT

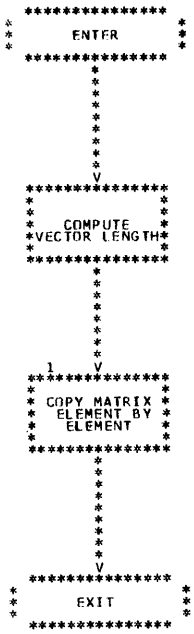


SUBROUTINE RTIE

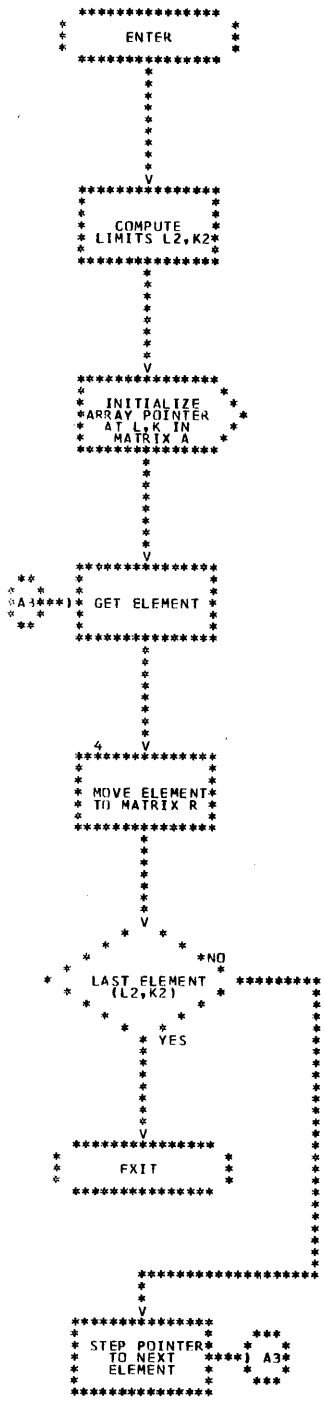



ENTER
SET STORAGE MODE AND NUMBER OF COLUMNS FOR A
INITIALIZE ARRAY POINTER AT FIRST ELEMENT
A2
GET ELEMENT
A3
2
MOVE ELEMENT TO MATRIX R
YES
LAST ELEMENT
NO
STEP POINTER TO NEXT ELEMENT
A3
NO
BOTH INPUT MATRICES PROCESSED
YES
EXIT
RESET STORAGE MODE AND NUMBER OF COLUMNS FOR B
A2

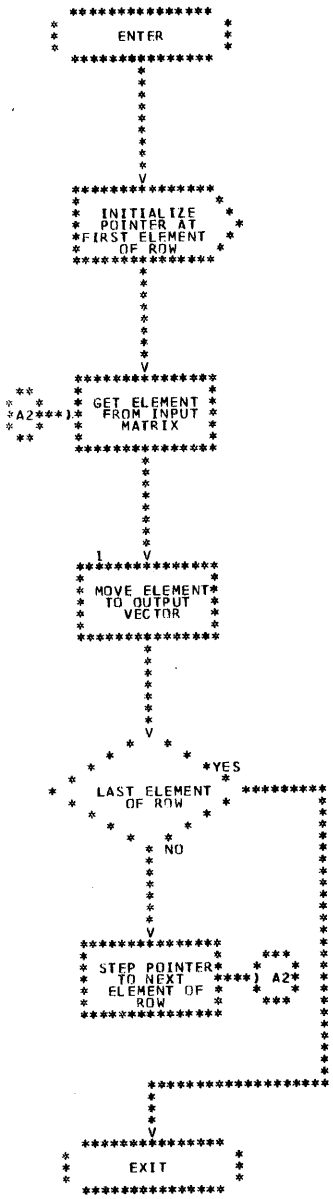
SUBROUTINE MCPY

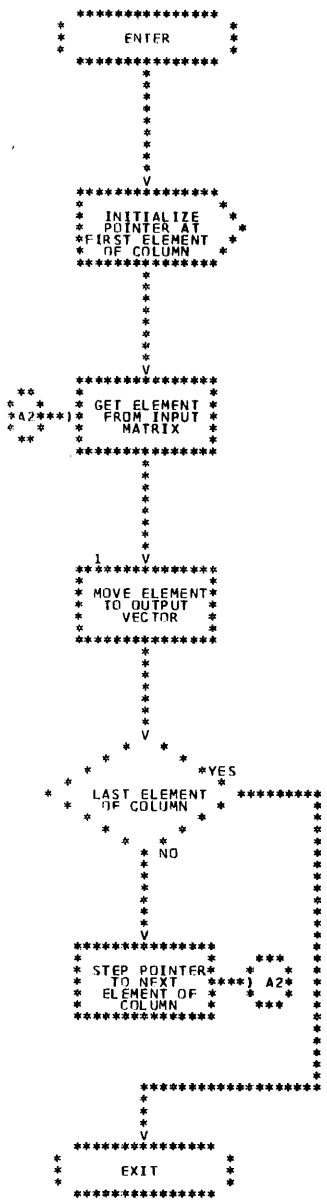


SUBROUTINE XCPY

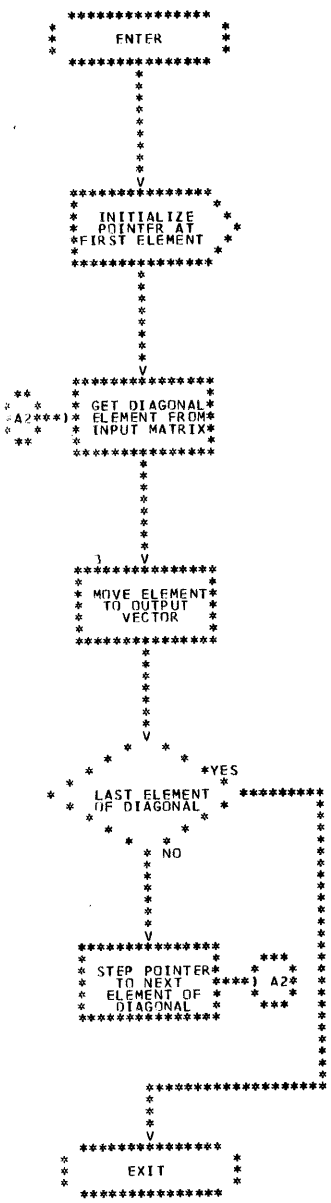


SUBROUTINE RCPY

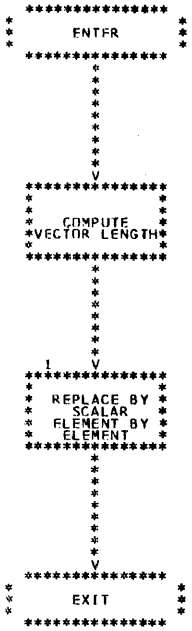



ENTER
INITIALIZE POINTER AT FIRST ELEMENT OF COLUMN
GET ELEMENT FROM INPUT MATRIX
A2
1
MOVE ELEMENT TO OUTPUT VECTOR
YES
LAST ELEMENT OF COLUMN
NO
STEP POINTER TO NEXT ELEMENT OF COLUMN
A2
EXIT

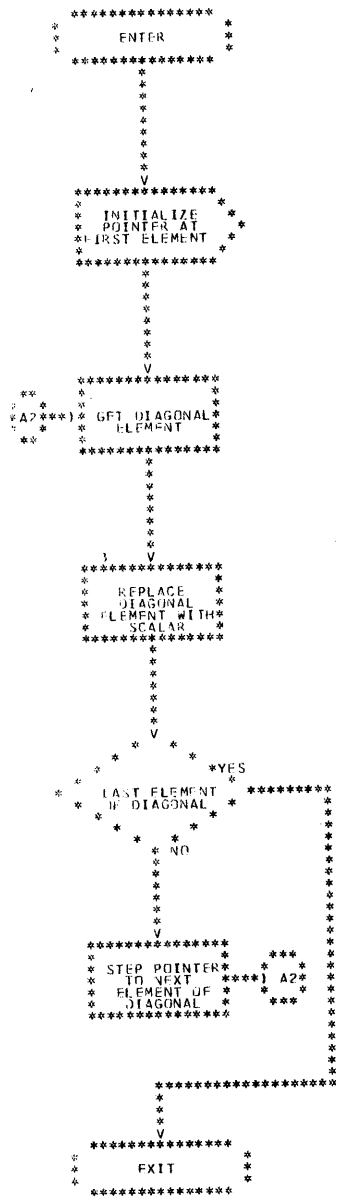
SUBROUTINE DCPY



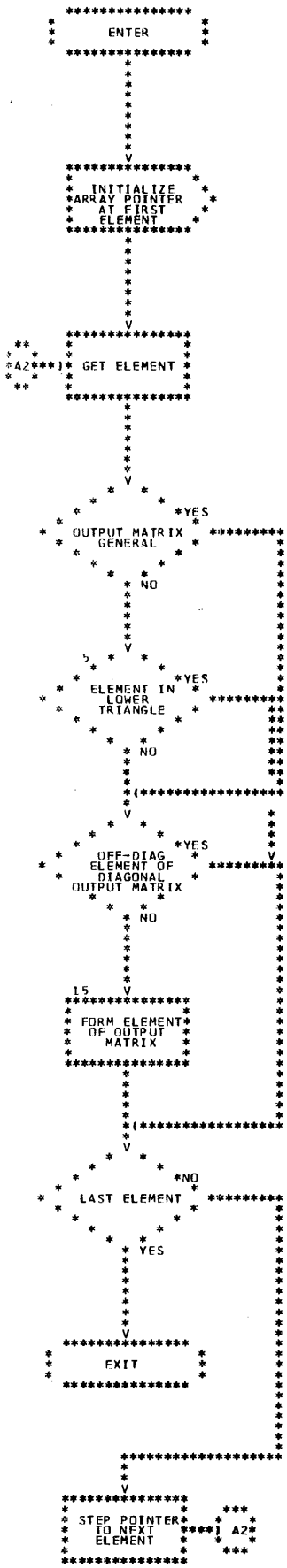
SUBROUTINE SCLA

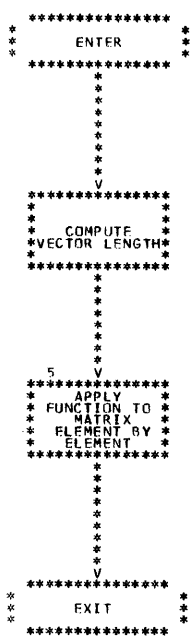


SUBROUTINE OCLA

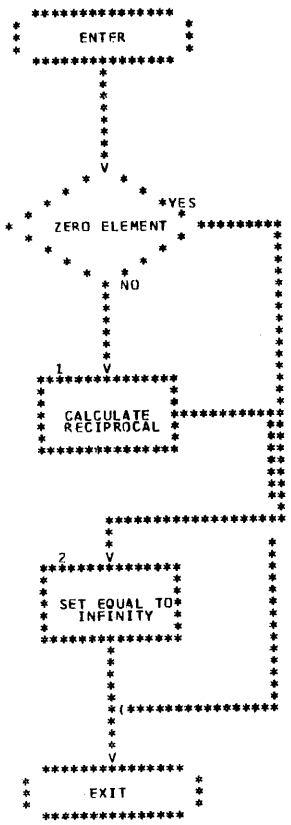


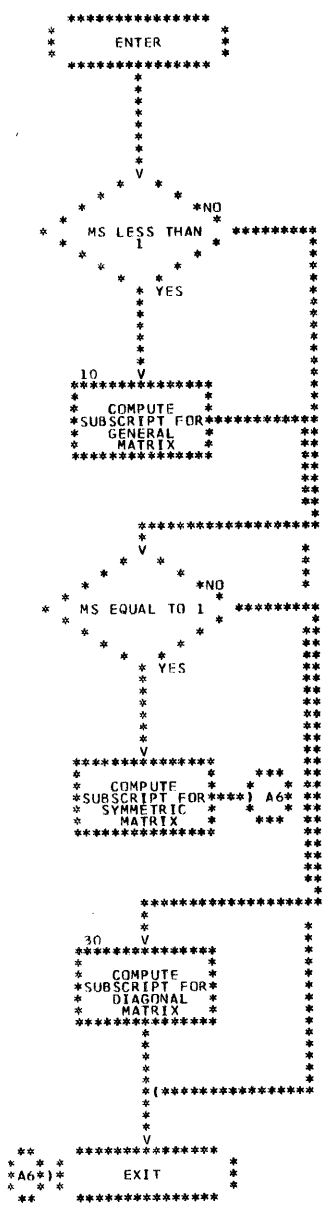
SUBROUTINE MSTR

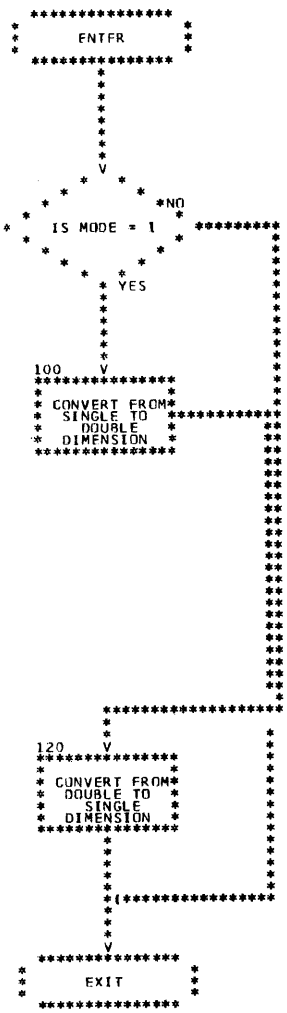


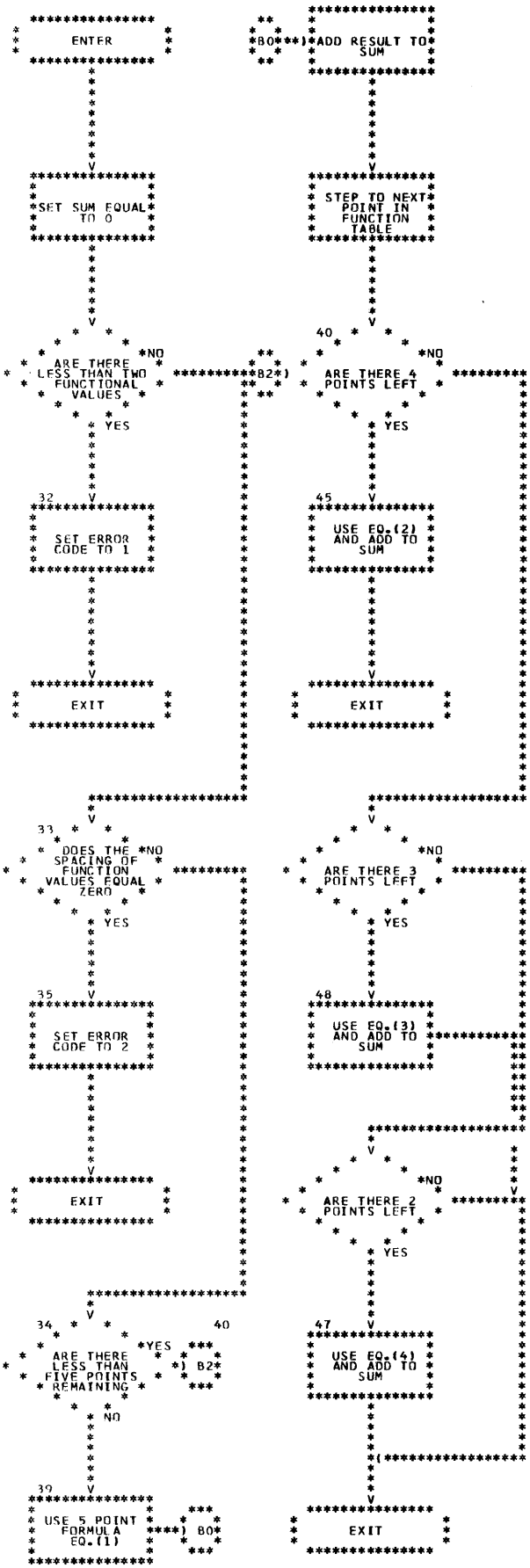

ENTER
COMPUTE VECTOR LENGTH
5
APPLY FUNCTION TO MATRIX ELEMENT BY ELEMENT
EXIT

FUNCTION RECP

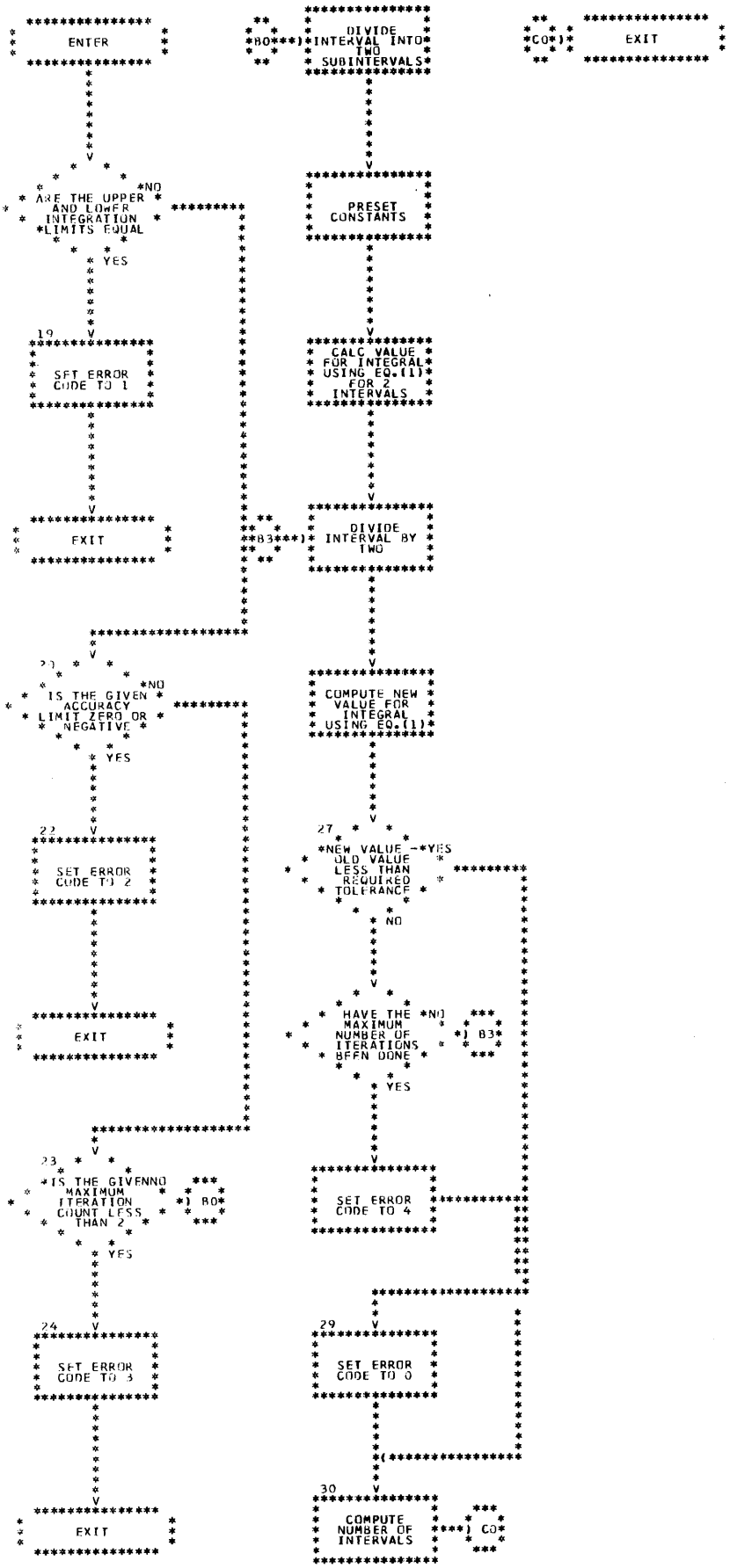



ENTER
MS LESS THAN 1
*NO
YES
10
COMPUTE *SUBSCRIPT FOR GENERAL MATRIX
MS EQUAL TO 1
*NO
YES
COMPUTE *SUBSCRIPT FOR SYMMETRIC MATRIX
A6*
30
COMPUTE *SUBSCRIPT FOR DIAGONAL MATRIX
A6*
EXIT

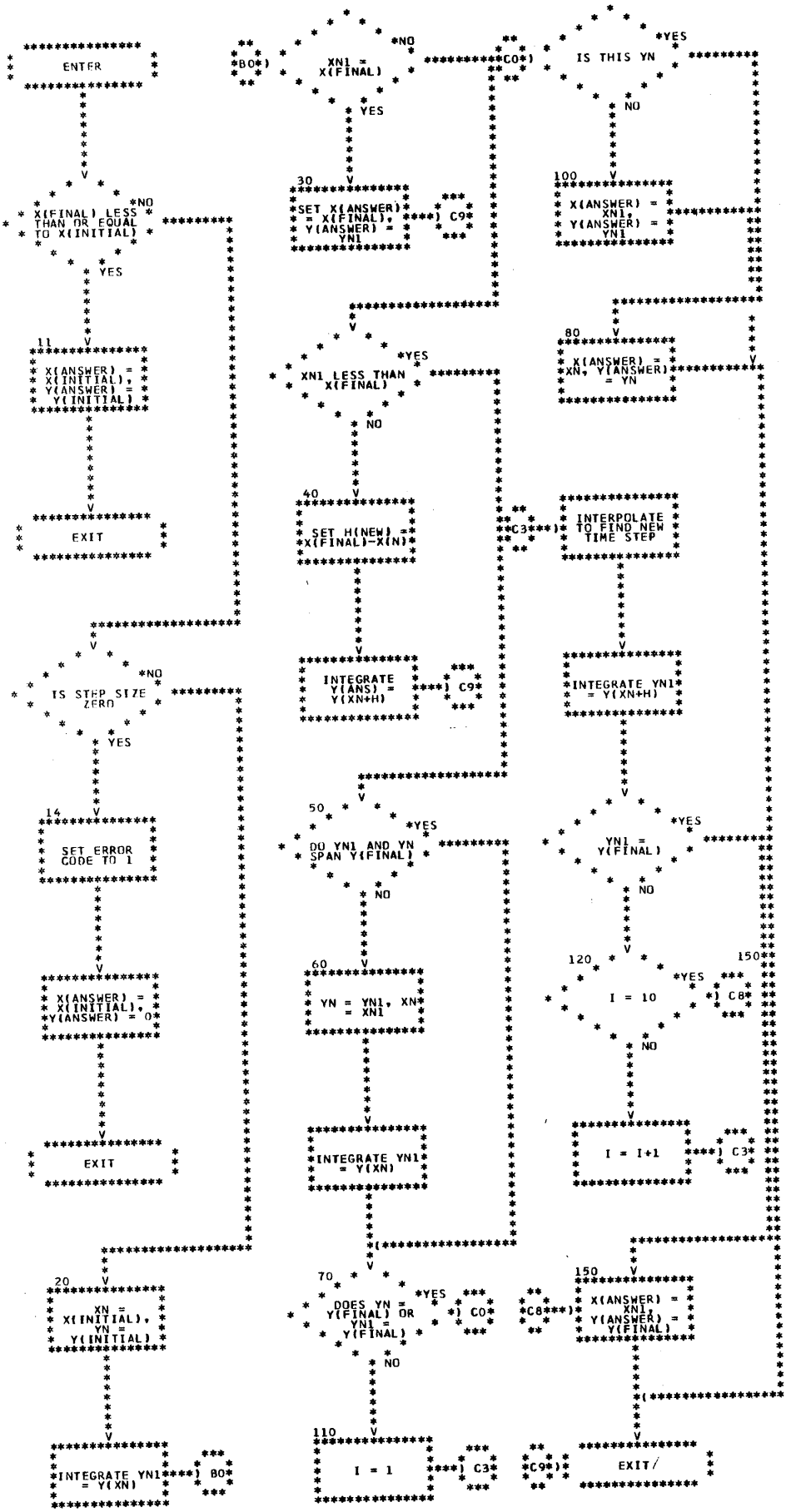

ENTER
IS MODE = 1
NO
YES
100
CONVERT FROM SINGLE TO DOUBLE DIMENSION
120
CONVERT FROM DOUBLE TO SINGLE DIMENSION
EXIT


ENTER
SET SUM EQUAL TO 0
ARE THERE LESS THAN TWO FUNCTIONAL VALUES
YES
SET ERROR CODE TO 1
EXIT
DOES THE SPACING OF FUNCTION VALUES EQUAL ZERO
YES
SET ERROR CODE TO 2
EXIT
ARE THERE LESS THAN FIVE POINTS REMAINING
NO
USE 5 POINT FORMULA EQ.(1)
B0
B2
ADD RESULT TO SUM
STEP TO NEXT POINT IN FUNCTION TABLE
ARE THERE 4 POINTS LEFT
YES
USE EQ.(2) AND ADD TO SUM
EXIT
ARE THERE 3 POINTS LEFT
YES
USE EQ.(3) AND ADD TO SUM
ARE THERE 2 POINTS LEFT
YES
USE EQ.(4) AND ADD TO SUM
EXIT

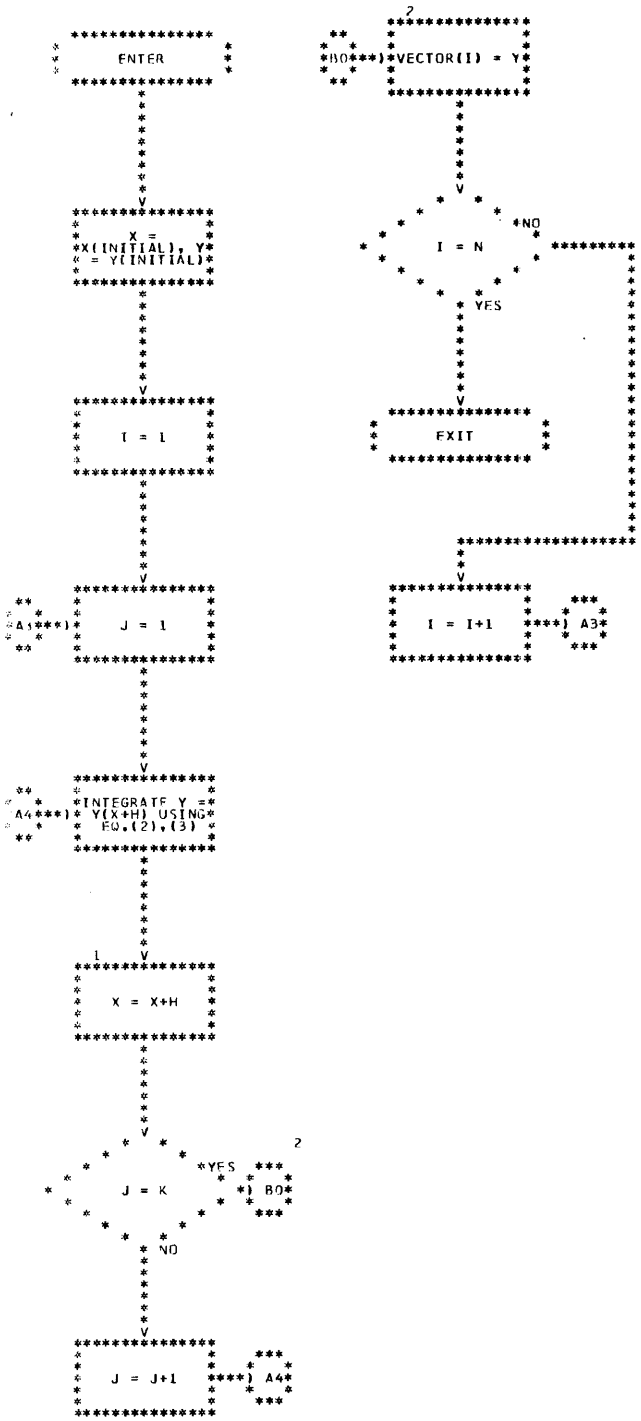
SUBROUTINE SMPSN



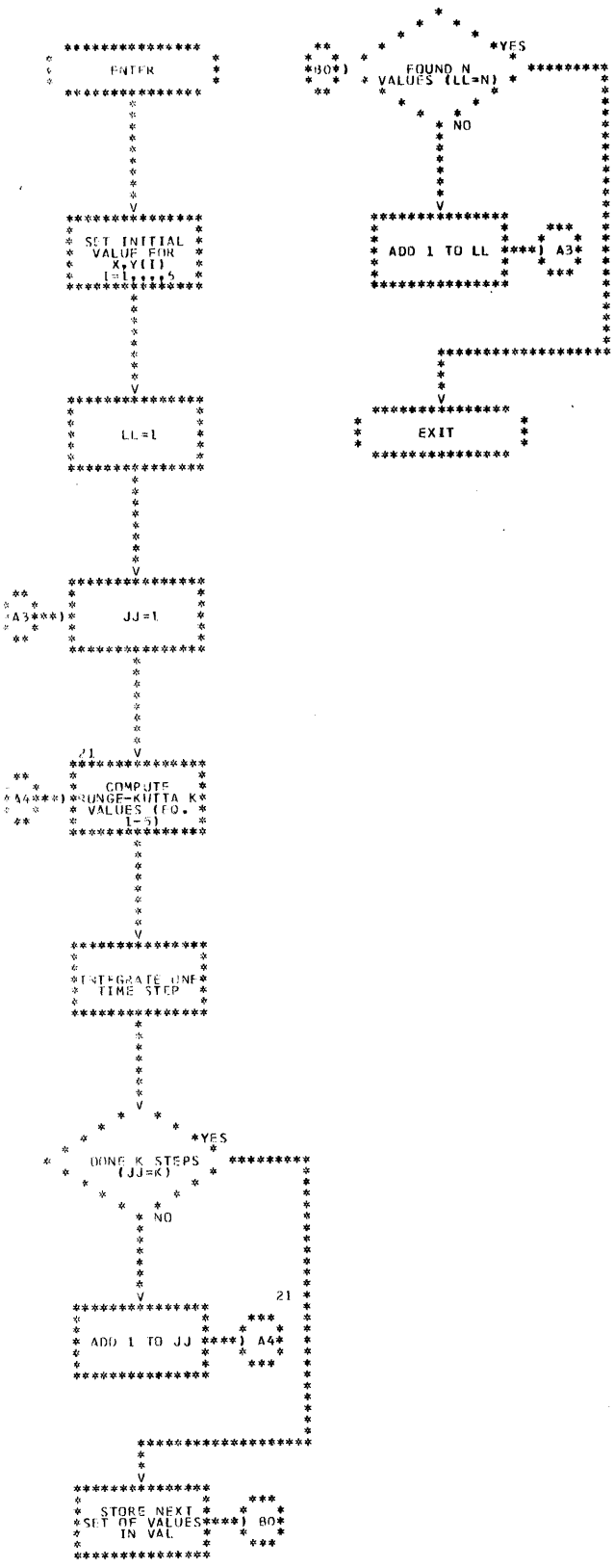
SUBROUTINE RK1

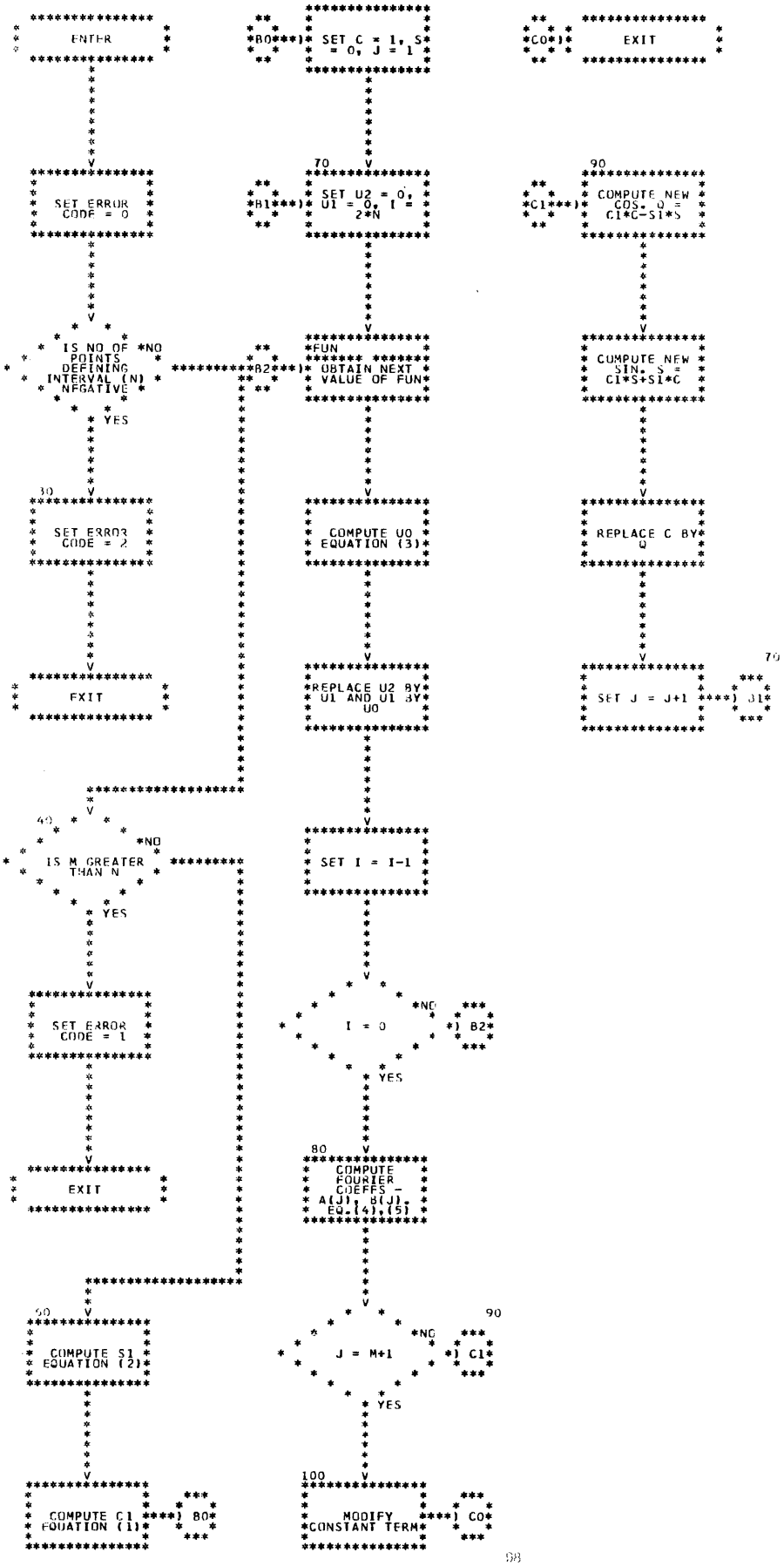


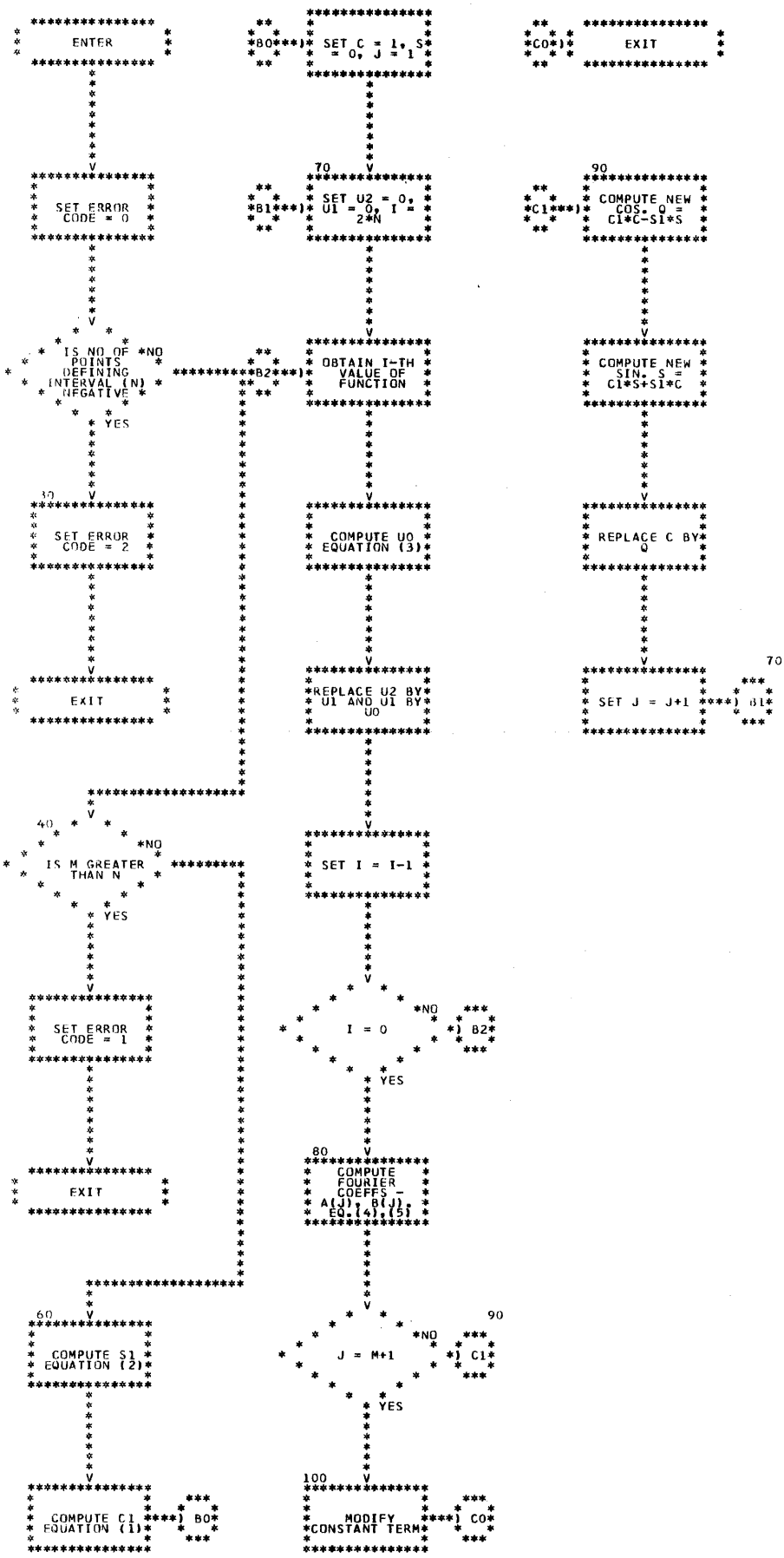
SUBROUTINE RK2

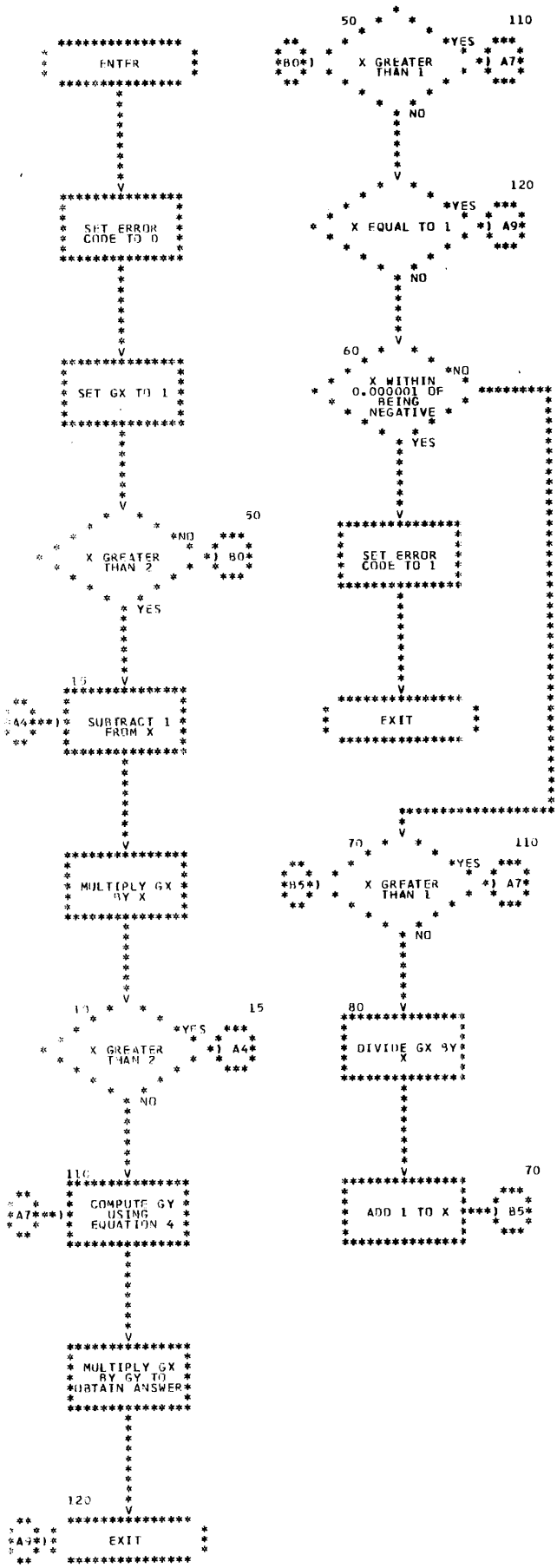


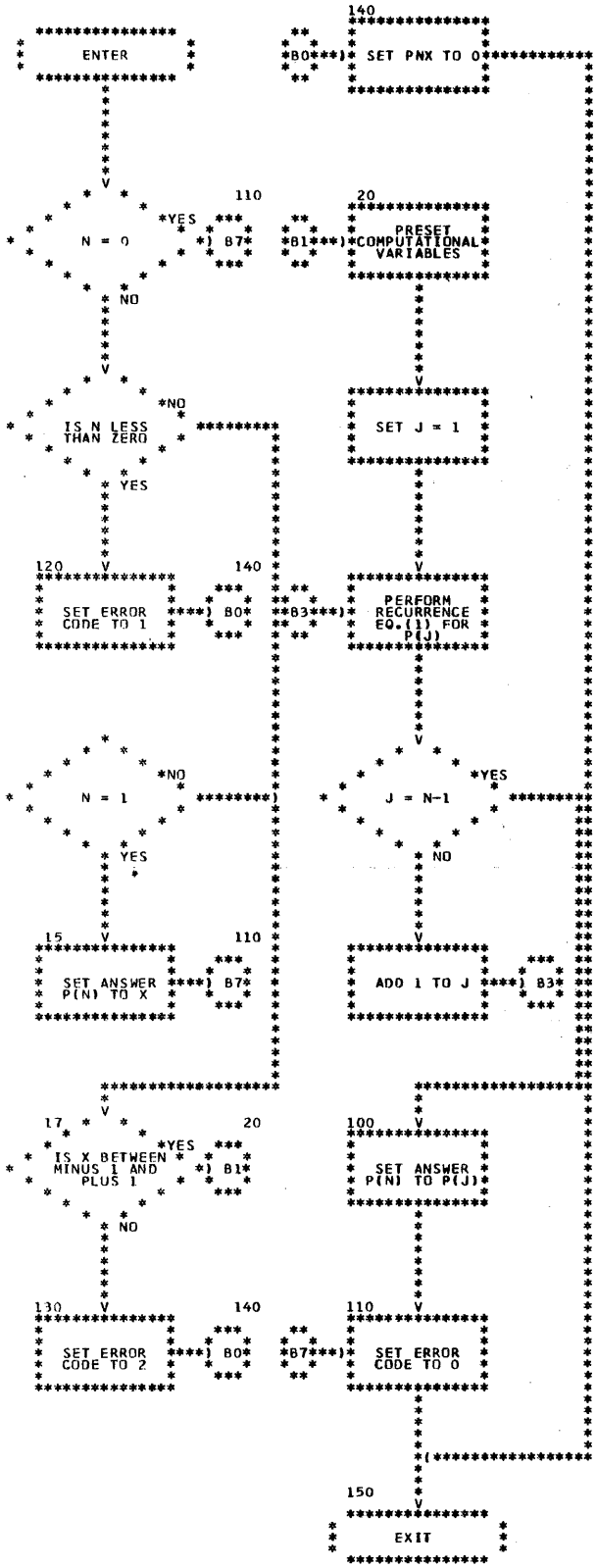
SUBROUTINE RK3



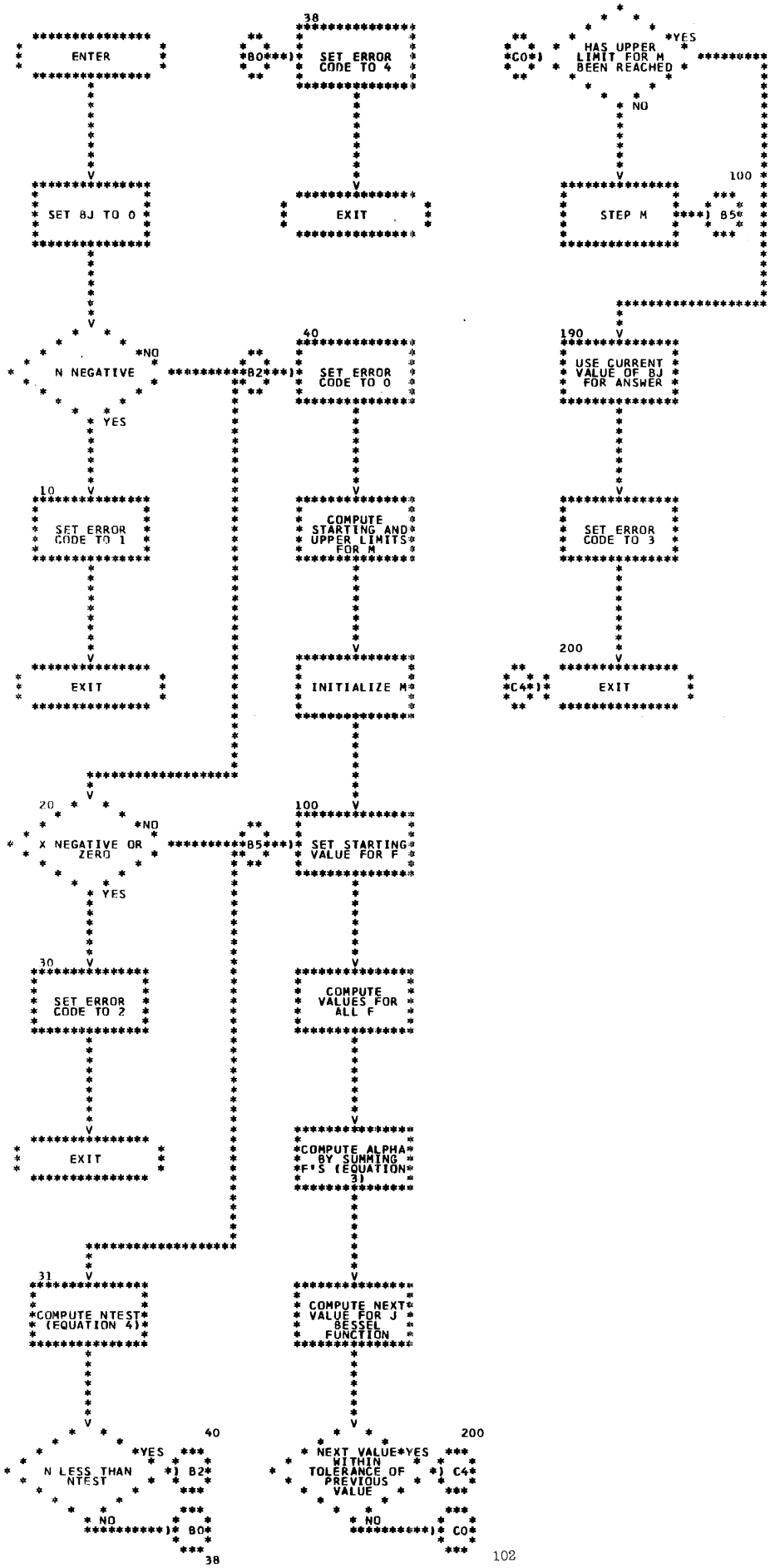

ENTER
SET ERROR CODE = 0
IS NO OF DEFINING INTERVAL (N) NEGATIVE
YES
SET ERROR CODE = 2
EXIT
IS M GREATER THAN N
NO
YES
SET ERROR CODE = 1
EXIT
COMPUTE S1 EQUATION (2)
COMPUTE C1 EQUATION (1)
B0
SET C = 1, S = 0, J = 1
70
SET U2 = 0, U1 = 0, I = 2*N
B1
B2
FUN
OBTAIN NEXT VALUE OF FUN
COMPUTE U0 EQUATION (3)
REPLACE U2 BY U1 AND U1 BY U0
SET I = I-1
I = 0
B2
YES
80
COMPUTE FOURIER COEFFS - A(J), B(J), EQ.(4),(5)
J = M+1
C1
YES
100
MODIFY CONSTANT TERM
C0
EXIT
90
COMPUTE NEW COS. Q = C1*C-S1*S
C1
COMPUTE NEW SIN. S = C1*S+S1*C
REPLACE C BY Q
SET J = J+1
31
79


ENTER
SET ERROR CODE = 0
IS NO OF POINTS DEFINING INTERVAL (N) NEGATIVE
*NO
YES
SET ERROR CODE = 2
EXIT
IS M GREATER THAN N
*NO
YES
SET ERROR CODE = 1
EXIT
COMPUTE S1 EQUATION (2)
COMPUTE C1 EQUATION (1)
*B0
SET C = 1, S = 0, J = 1
*B1
SET U2 = 0, U1 = 0, I = 2*N
*B2
OBTAIN I-TH VALUE OF FUNCTION
COMPUTE U0 EQUATION (3)
REPLACE U2 BY U1 AND U1 BY U0
SET I = I-1
I = 0
*NO
*B2
YES
COMPUTE FOURIER COEFFS - A(J), B(J), EQ.(4),(5)
J = M+1
*NO
*C1
YES
MODIFY CONSTANT TERM
*C0
EXIT
*C1
COMPUTE NEW COS. Q = C1*C-S1*S
COMPUTE NEW SIN. S = C1*S+S1*C
REPLACE C BY Q
SET J = J+1
*B1
10
30
40
60
70
80
90
100

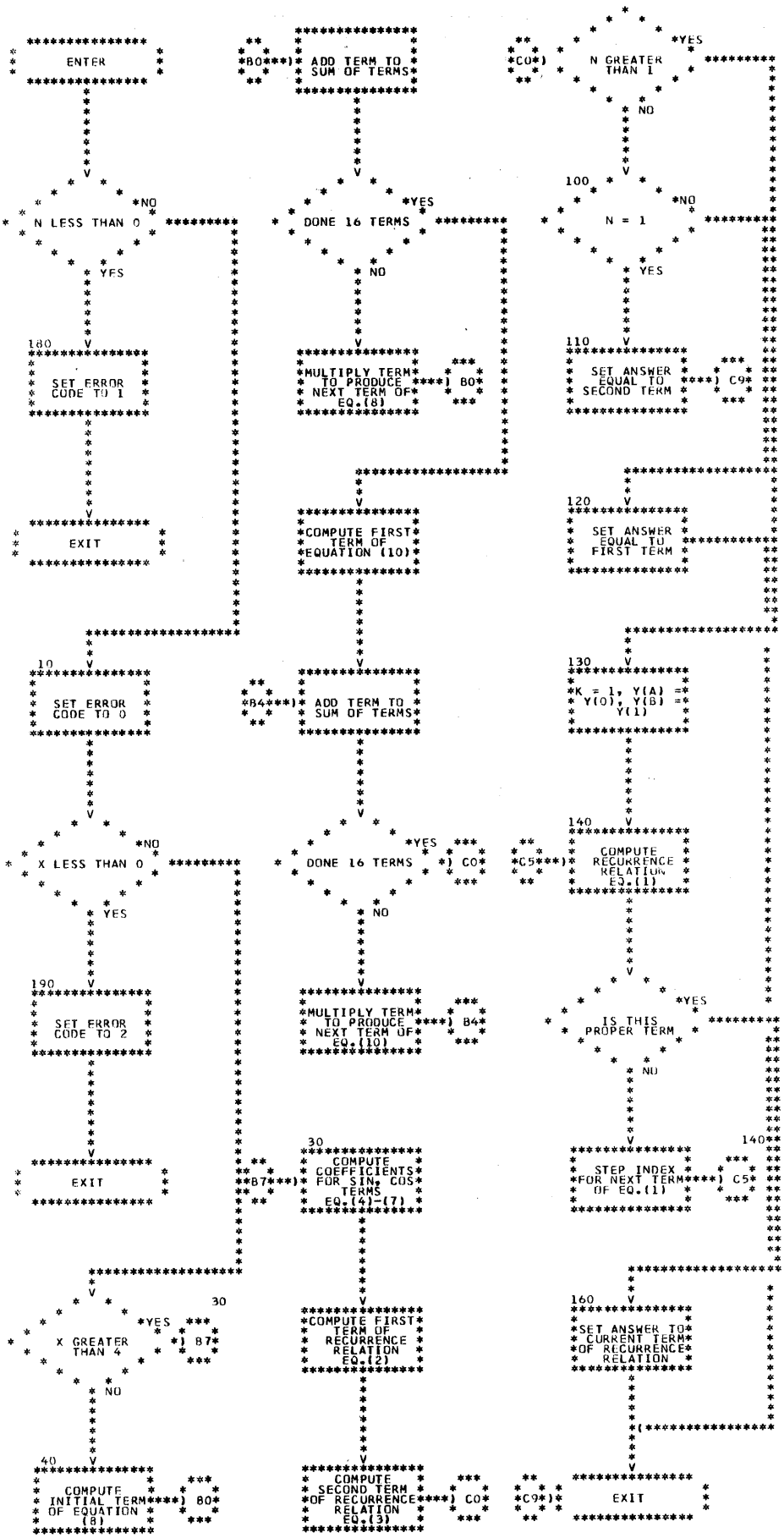

ENTER
SET ERROR CODE TO 0
SET GX TO 1
X GREATER THAN 2
*NO
) B0
50
*YES
SUBTRACT 1 FROM X
*A4***)
MULTIPLY GX BY X
X GREATER THAN 2
*YES
) A4
15
*NO
COMPUTE GY USING EQUATION 4
*A7***)
MULTIPLY GX BY GY TO OBTAIN ANSWER
EXIT
A9)
X GREATER THAN 1
B0)
) A7
110
X EQUAL TO 1
120
X WITHIN 0.000001 OF BEING NEGATIVE
SET ERROR CODE TO 1
EXIT
X GREATER THAN 1
B5)
DIVIDE GX BY X
ADD 1 TO X
) B5
70
80


ENTER
B0
140
SET PNX TO 0
110
YES
N = 0
B7
B1
20
PRESET COMPUTATIONAL VARIABLES
NO
NO
IS N LESS THAN ZERO
SET J = 1
YES
120
SET ERROR CODE TO 1
B0
140
B3
PERFORM RECURRENCE EQ.(1) FOR P(J)
NO
N = 1
YES
J = N-1
YES
NO
15
SET ANSWER P(N) TO X
B7
110
ADD 1 TO J
B3
17
IS X BETWEEN MINUS 1 AND PLUS 1
YES
B1
20
100
SET ANSWER P(N) TO P(J)
NO
130
SET ERROR CODE TO 2
B0
140
B7
110
SET ERROR CODE TO 0
150
EXIT

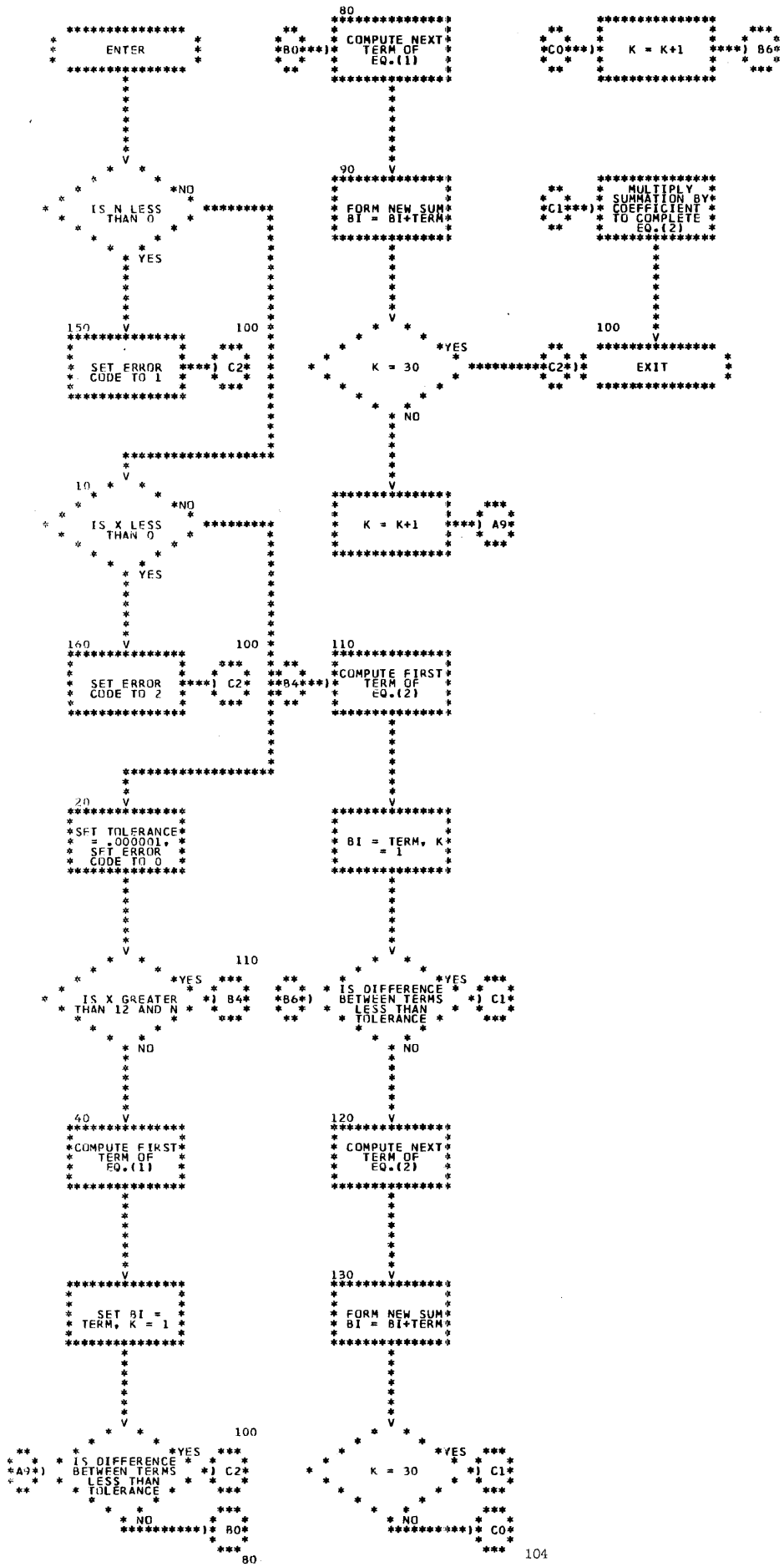
SUBROUTINE BESJ



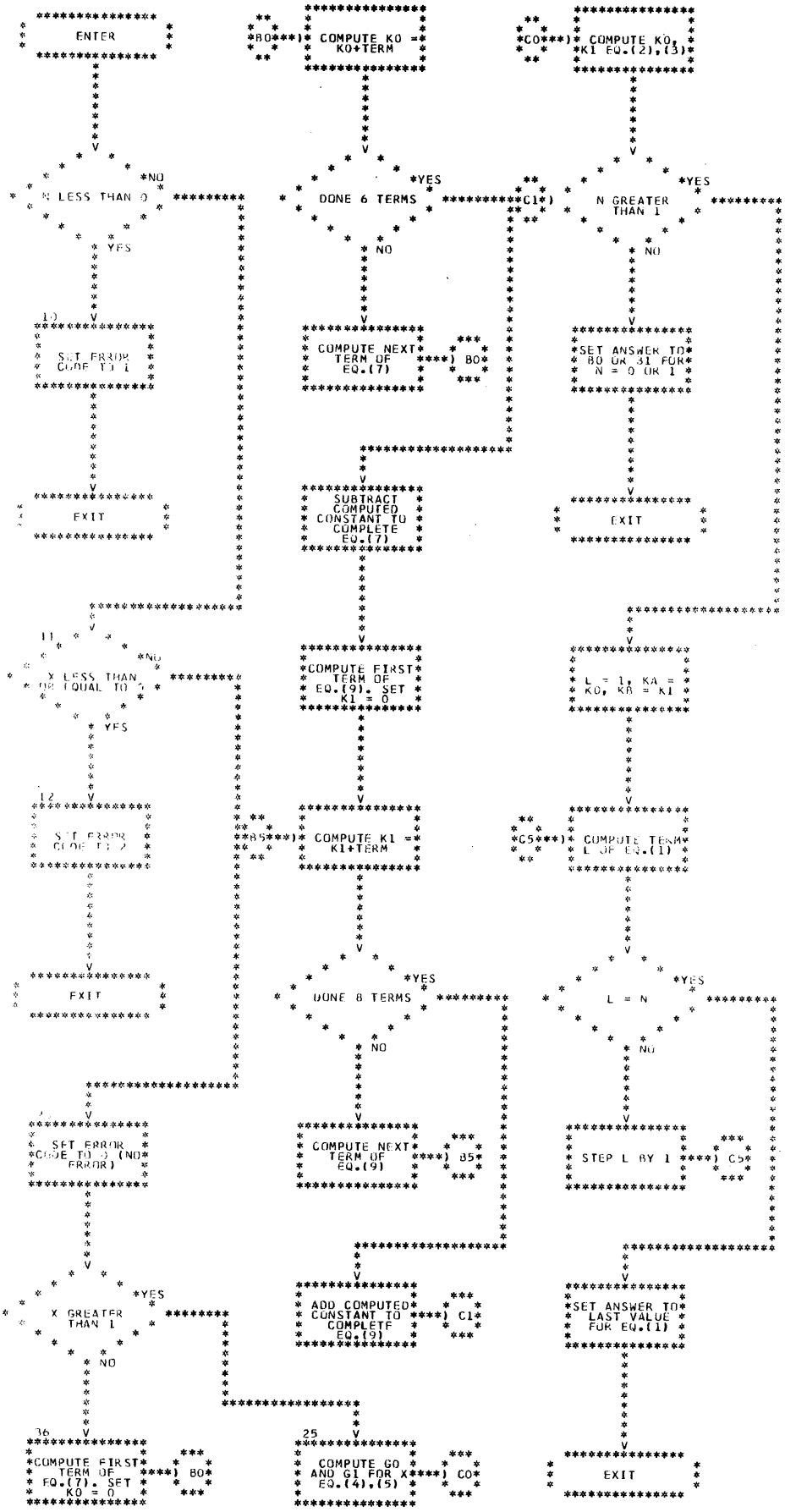
SUBROUTINE BESY

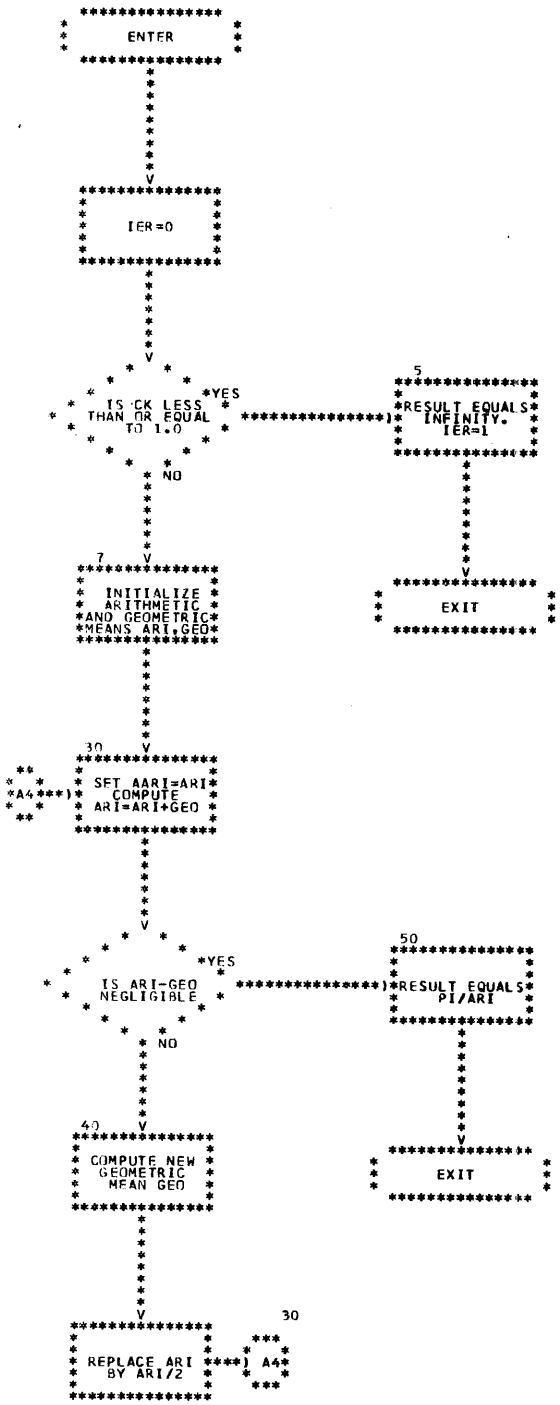


SUBROUTINE BESI

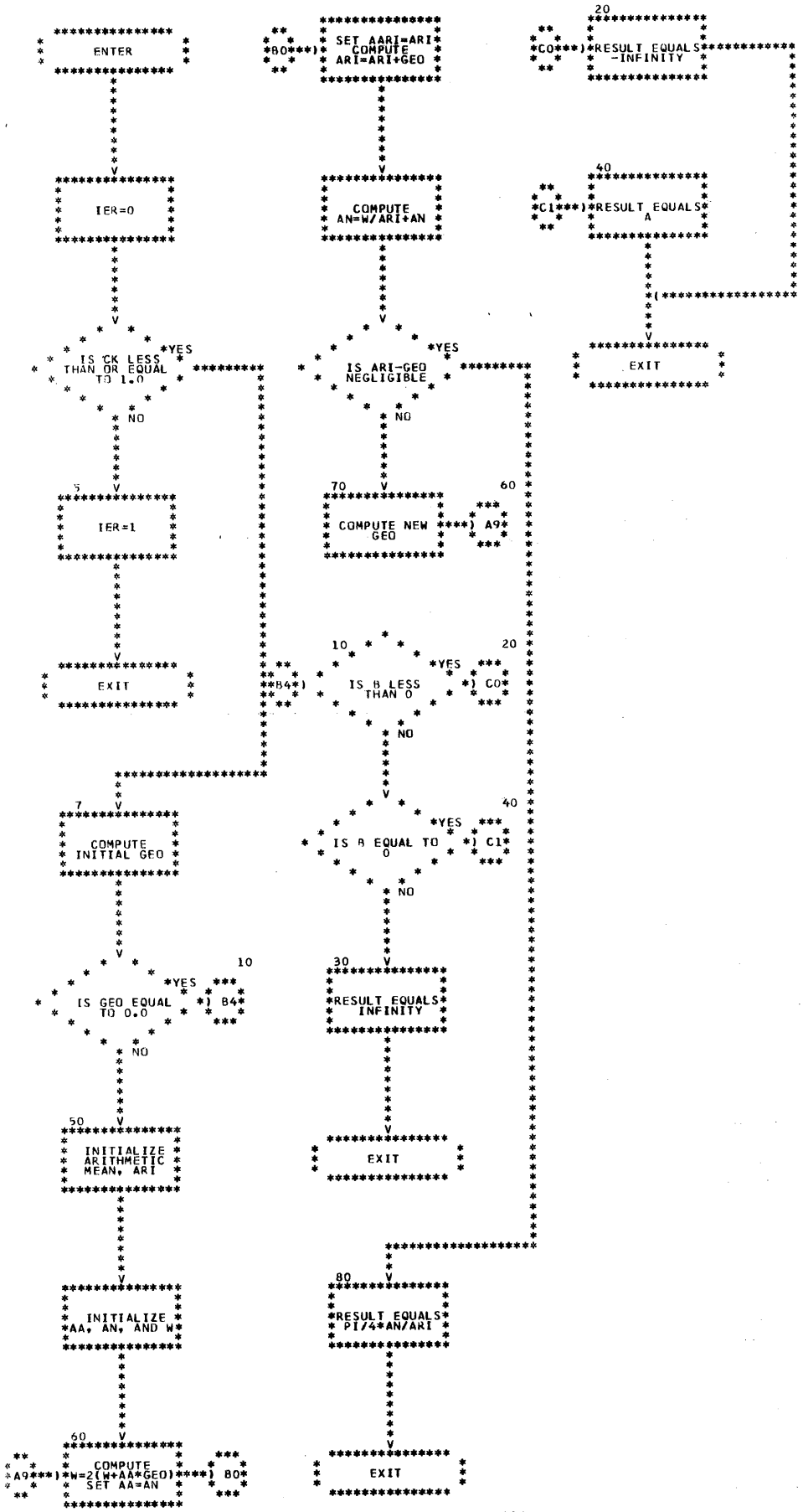


SUBROUTINE BESK

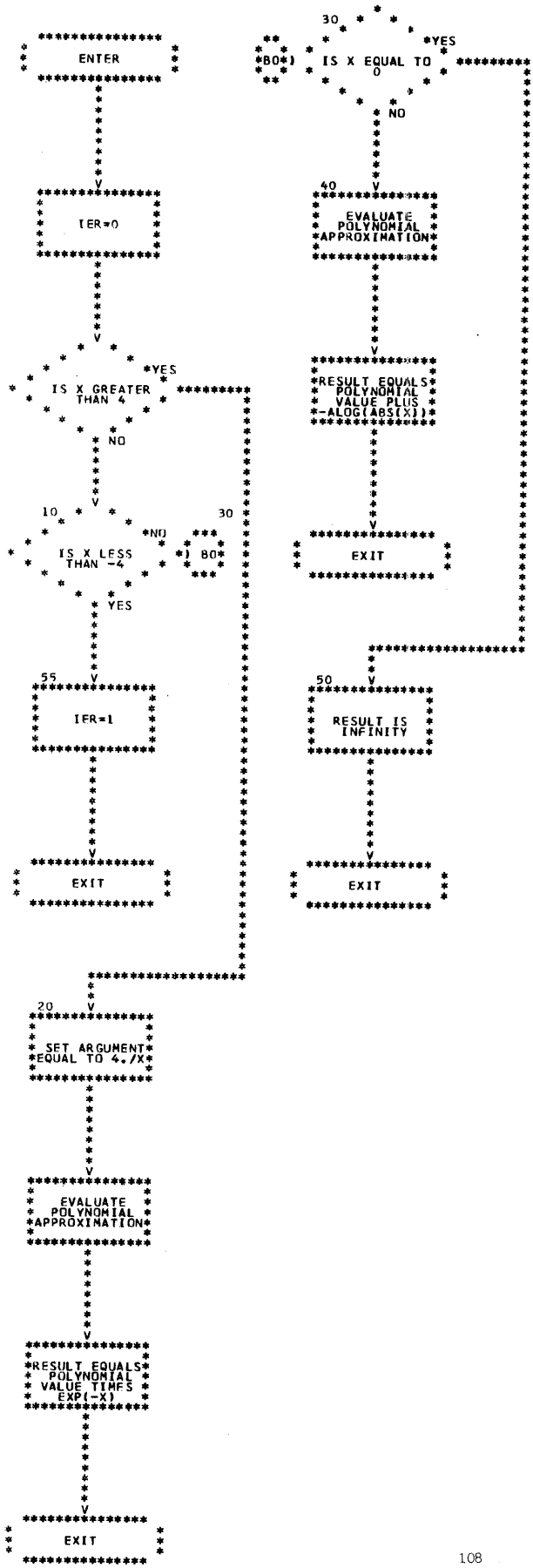



ENTER
IER=0
IS CK LESS THAN OR EQUAL TO 1.0
YES
5
RESULT EQUALS INFINITY. IER=1
NO
7
INITIALIZE ARITHMETIC AND GEOMETRIC MEANS ARI,GEO
EXIT
30
SET AARI=ARI COMPUTE ARI=ARI+GEO
A4
IS ARI-GEO NEGLIGIBLE
YES
50
RESULT EQUALS PI/ARI
NO
40
COMPUTE NEW GEOMETRIC MEAN GEO
EXIT
REPLACE ARI BY ARI/2
30
A4

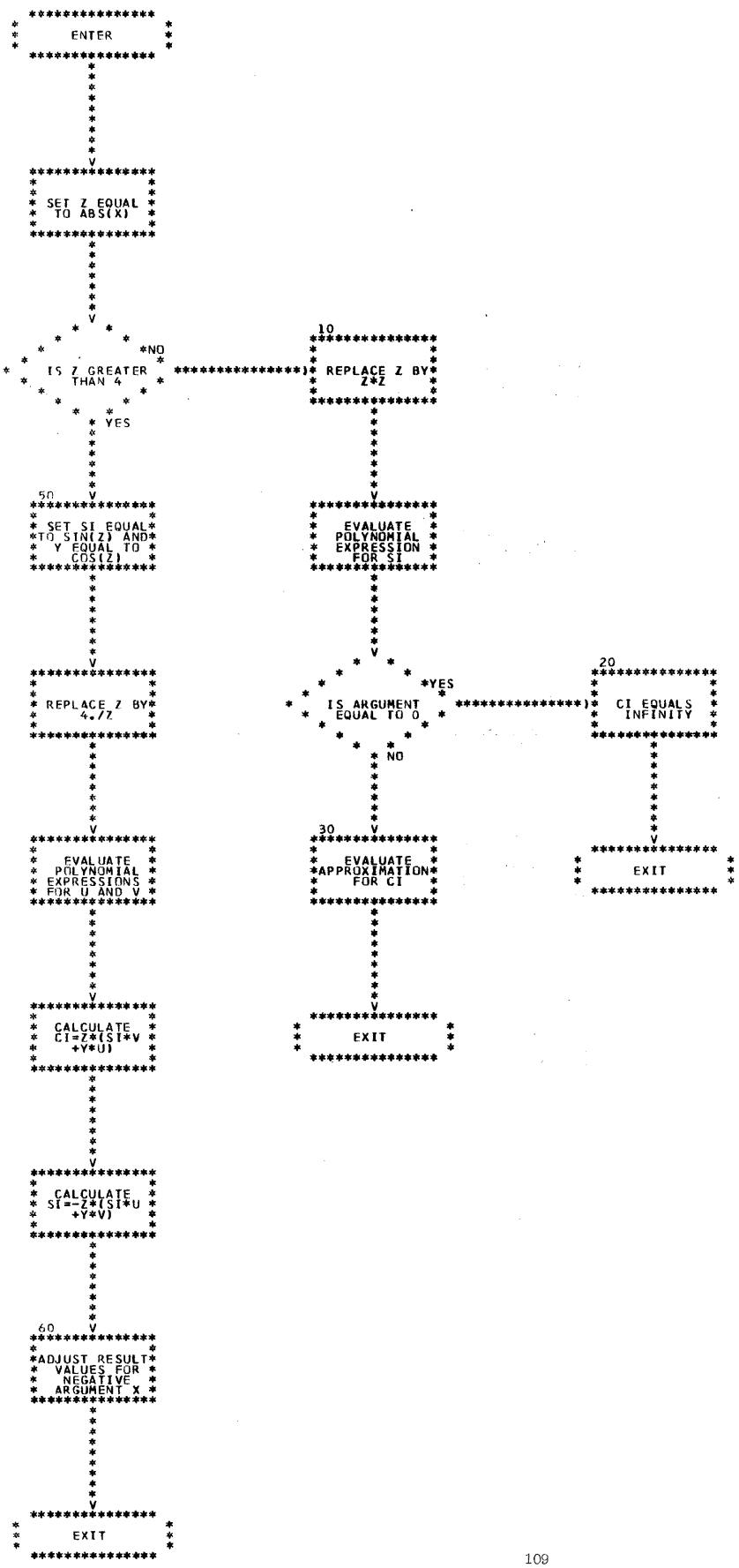
SUBROUTINE CELI2



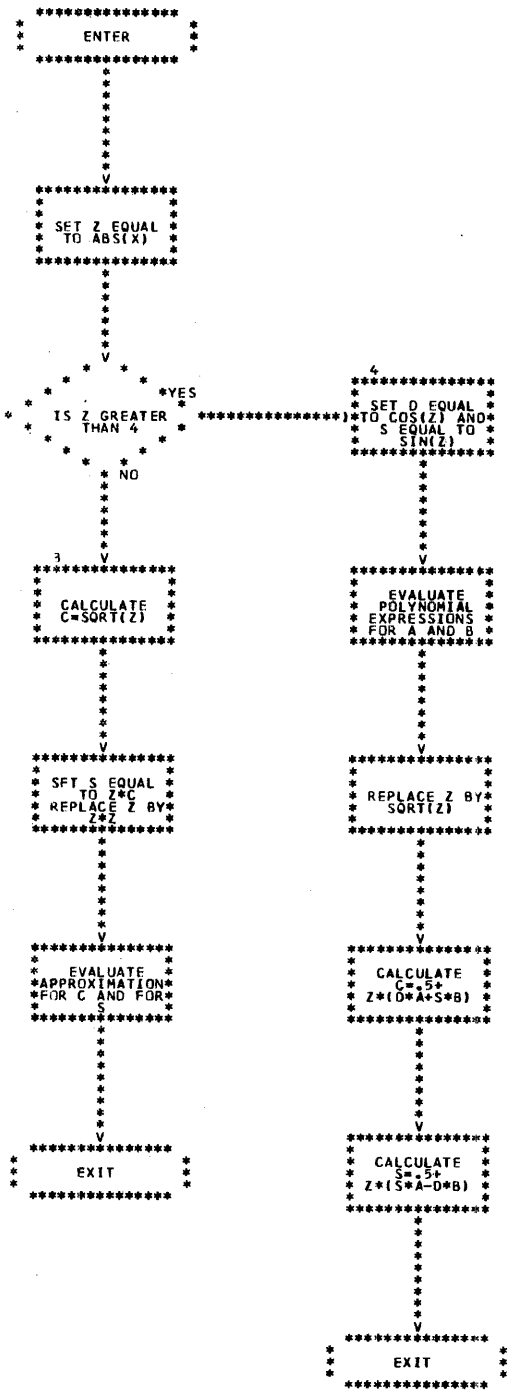
SUBROUTINE EXPI



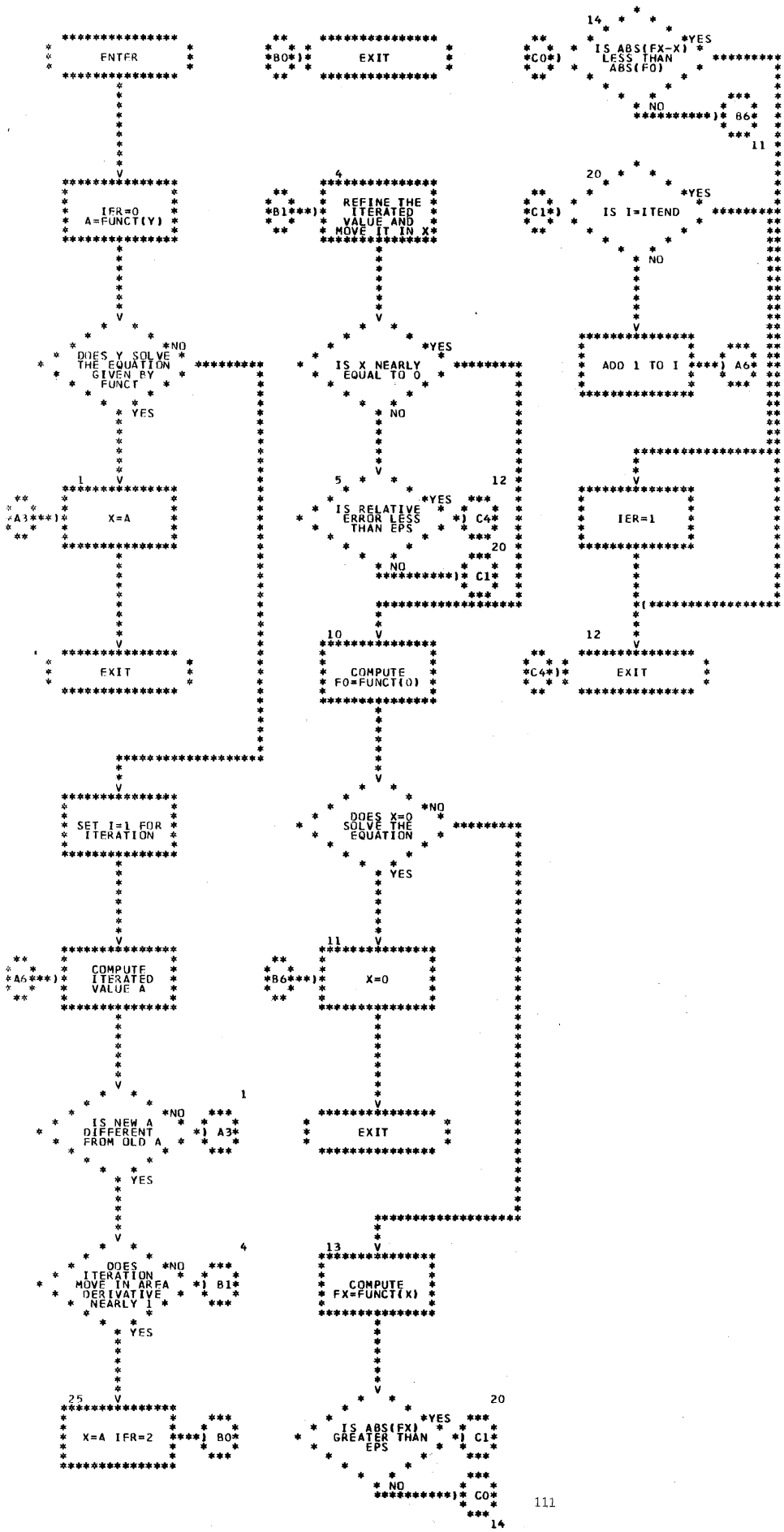
SUBROUTINE SICI

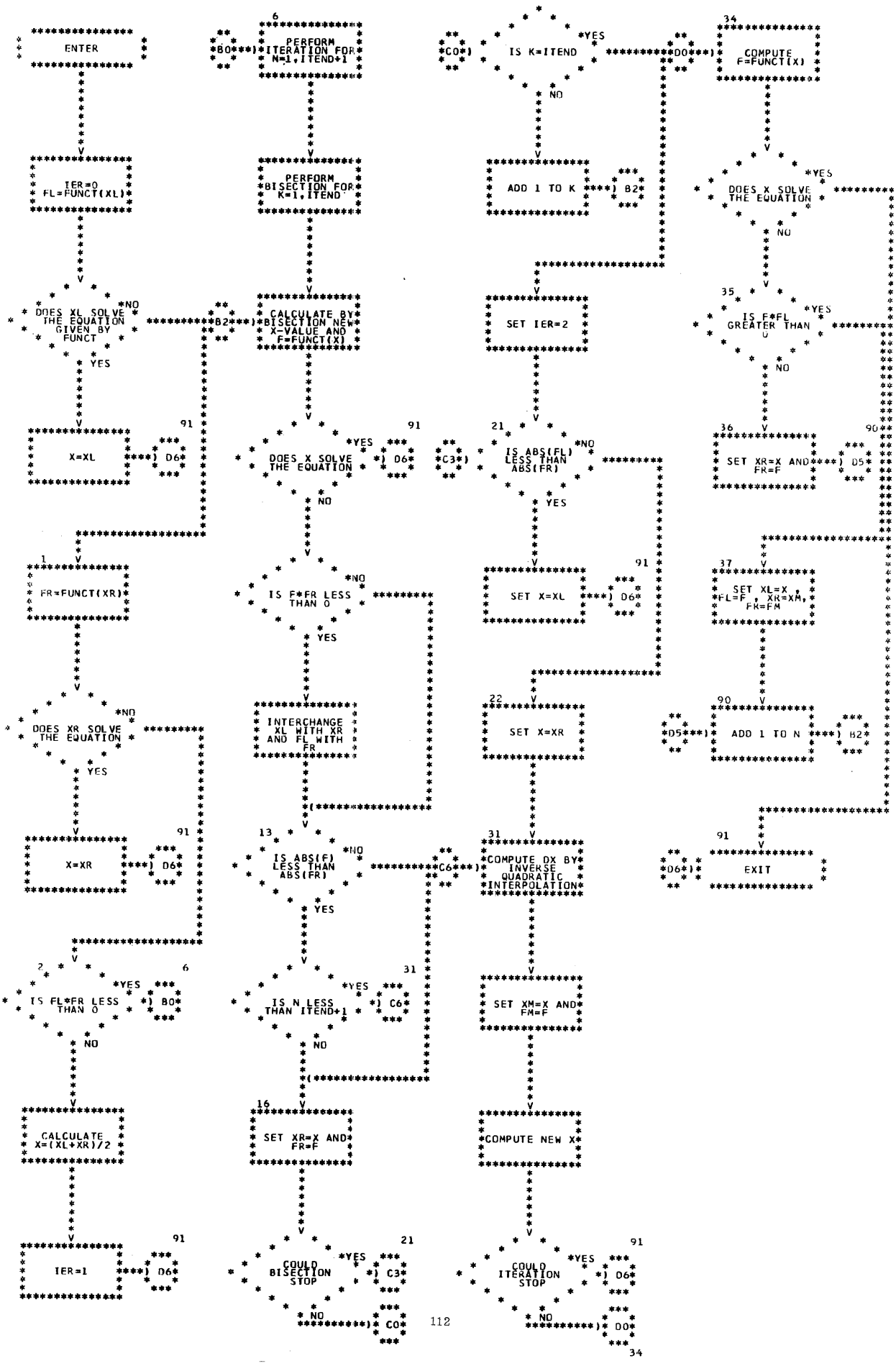


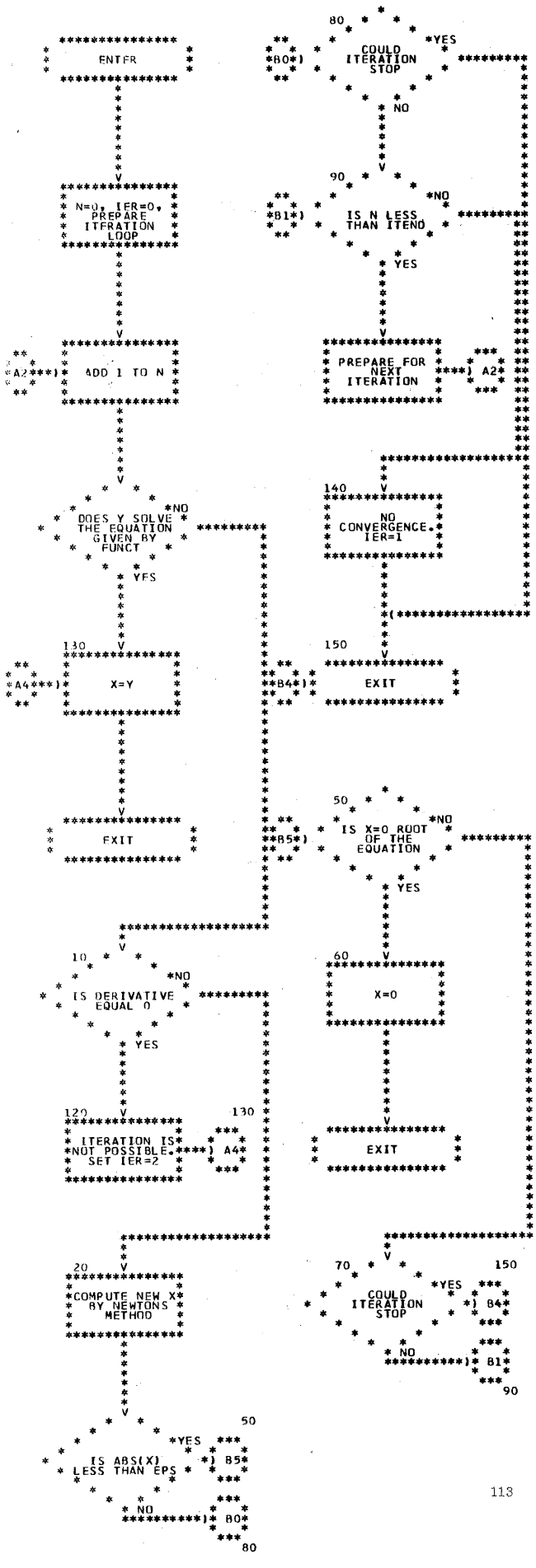
SUBROUTINE CS



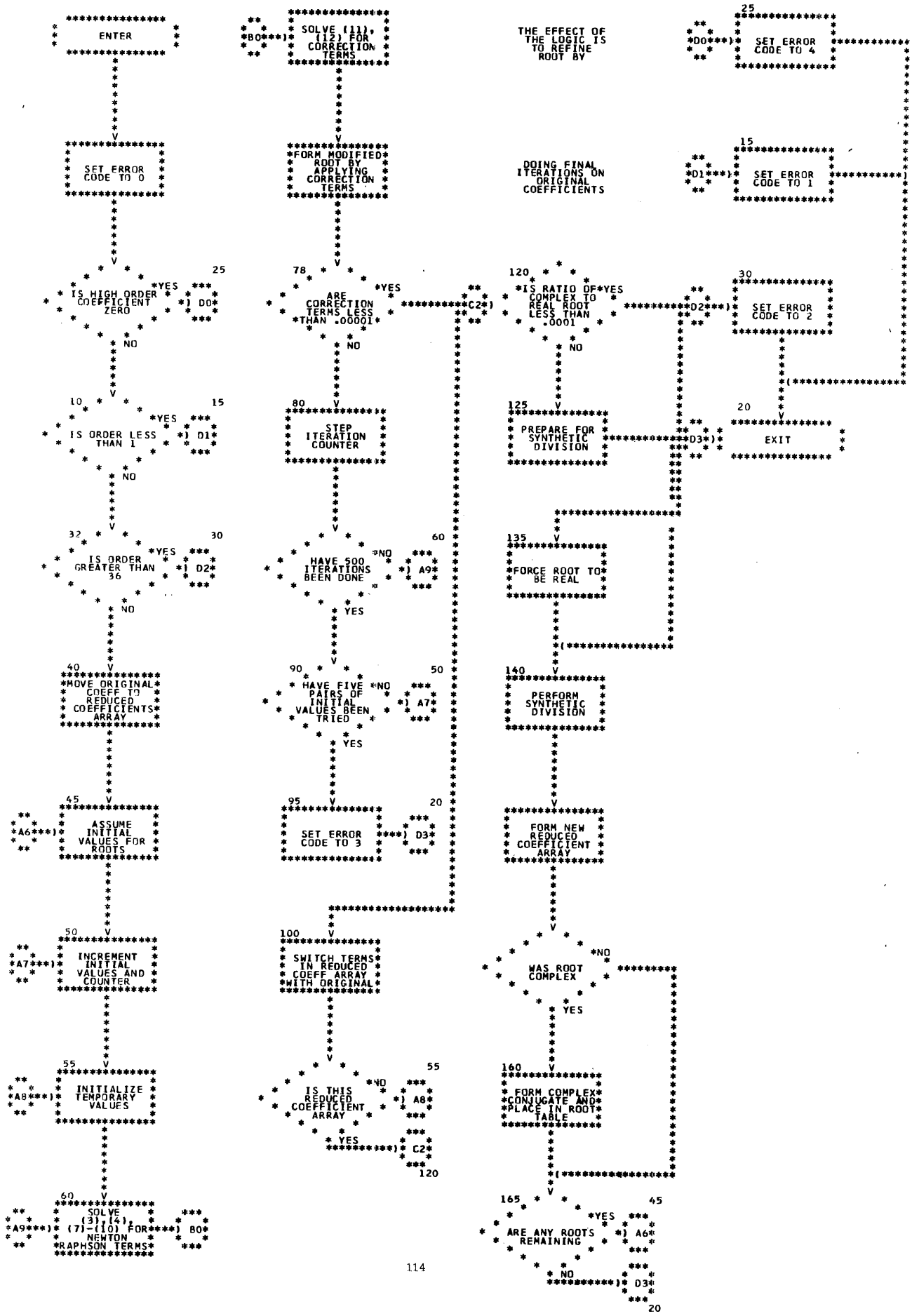
SUBROUTINE RTWIT



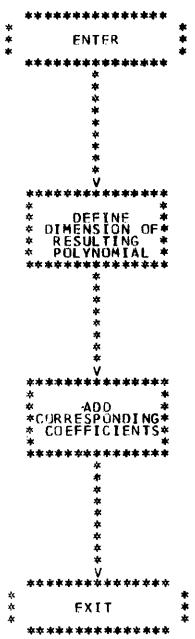

ENTER
IER=0
FL=FUNCT(XL)
DOES XL SOLVE THE EQUATION GIVEN BY FUNCT
YES
NO
X=XL
FR=FUNCT(XR)
DOES XR SOLVE THE EQUATION
YES
NO
X=XR
IS FL*FR LESS THAN 0
YES
NO
CALCULATE X=(XL+XR)/2
IER=1
PERFORM ITERATION FOR N=1,ITEND+1
PERFORM BISECTION FOR K=1,ITEND
CALCULATE BISECTION NEW X-VALUE AND F=FUNCT(X)
DOES X SOLVE THE EQUATION
YES
NO
IS F*FR LESS THAN 0
YES
INTERCHANGE XL WITH XR AND FL WITH FR
IS ABS(F) LESS THAN ABS(FR)
YES
NO
IS N LESS THAN ITEND+1
YES
NO
SET XR=X AND FR=F
COULD BISECTION STOP
YES
NO
IS K=ITEND
YES
NO
ADD 1 TO K
SET IER=2
IS ABS(FL) LESS THAN ABS(FR)
YES
NO
SET X=XL
SET X=XR
COMPUTE DX BY INVERSE QUADRATIC INTERPOLATION
SET XM=X AND FM=F
COMPUTE NEW X
COULD ITERATION STOP
YES
NO
COMPUTE F=FUNCT(X)
DOES X SOLVE THE EQUATION
YES
NO
IS F*FL GREATER THAN 0
YES
NO
SET XR=X AND FR=F
SET XL=X, FL=F, XR=XM, FR=FM
ADD 1 TO N
EXIT
B0
B2
C0
C3
C6
D0
D5
D6


ENTER
N=0, IER=0, PREPARE ITERATION LOOP
ADD 1 TO N
DOES Y SOLVE THE EQUATION GIVEN BY FUNCT
NO
YES
X=Y
EXIT
IS DERIVATIVE EQUAL 0
NO
YES
ITERATION IS NOT POSSIBLE. SET IER=2
COMPUTE NEW X BY NEWTONS METHOD
IS ABS(X) LESS THAN EPS
YES
NO
COULD ITERATION STOP
YES
NO
IS N LESS THAN ITEND
NO
YES
PREPARE FOR NEXT ITERATION
NO CONVERGENCE. IER=1
EXIT
IS X=0 ROOT OF THE EQUATION
NO
YES
X=0
EXIT
COULD ITERATION STOP
YES
NO
10
20
50
60
70
80
90
120
130
140
150
A2
A4
B4
B5
B0
B1

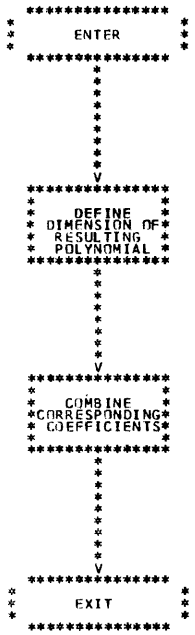
SUBROUTINE POLRT



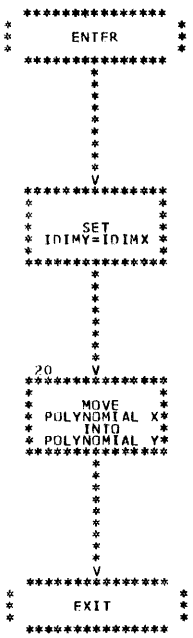
SUBROUTINE PADD



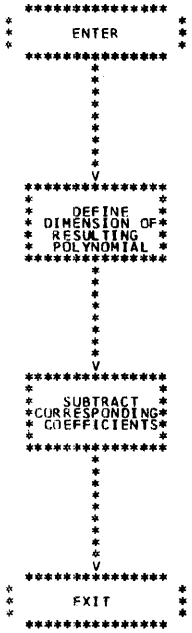
SUBROUTINE PADDM



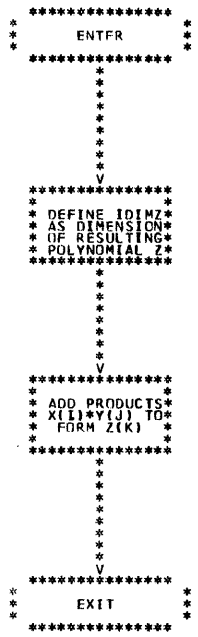
SUBROUTINE PCLA

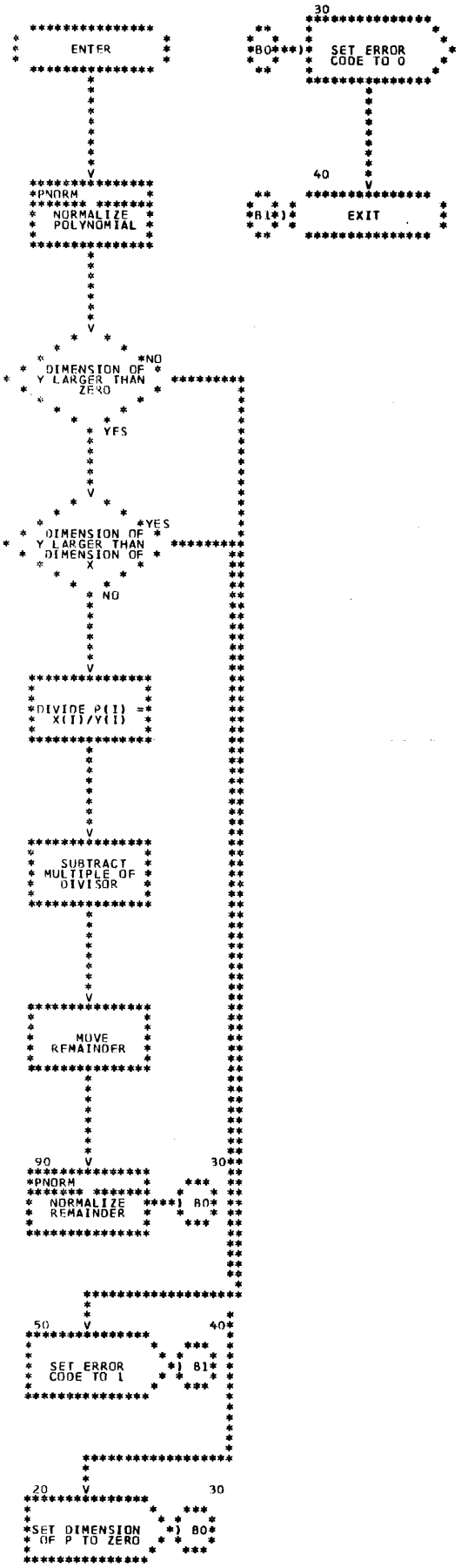


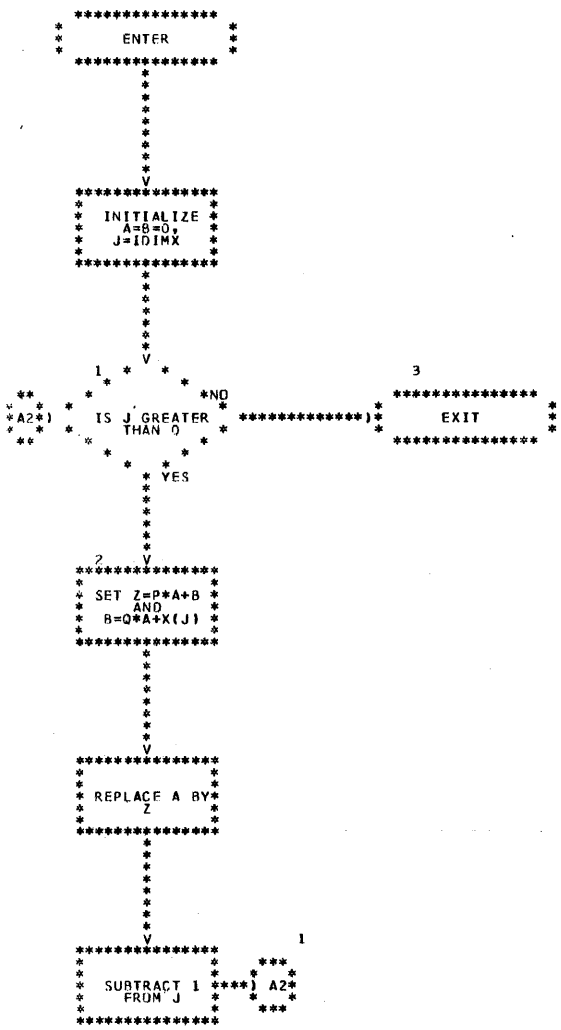
SUBROUTINE PSUB



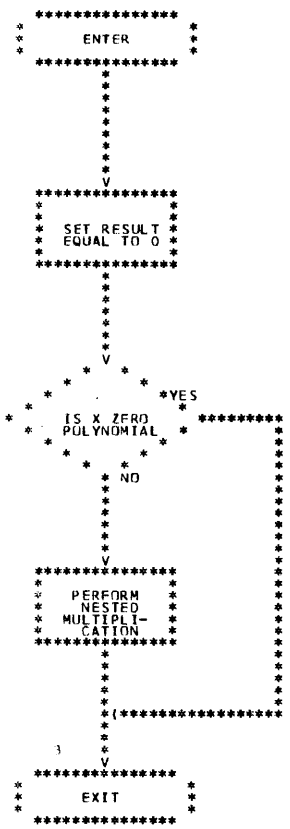
SUBROUTINE PMPY



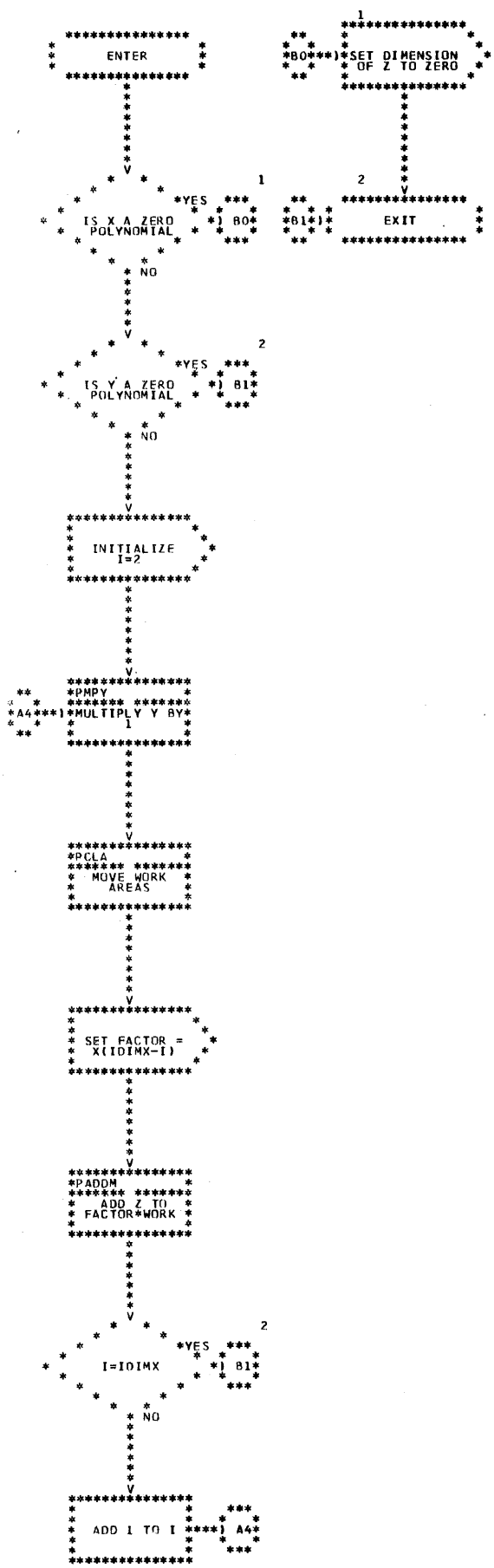

ENTER
*PNORM
NORMALIZE
POLYNOMIAL
*NO
DIMENSION OF
Y LARGER THAN
ZERO
YES
*YES
DIMENSION OF
Y LARGER THAN
DIMENSION OF
X
NO
DIVIDE P(I) =
X(I)/Y(I)
SUBTRACT
MULTIPLE OF
DIVISOR
MOVE
REMAINDER
90
*PNORM
NORMALIZE
REMAINDER
30
*80
50
SET ERROR
CODE TO 1
*81
40
20
SET DIMENSION
OF P TO ZERO
*80
30
SET ERROR
CODE TO 0
40
EXIT
*81


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

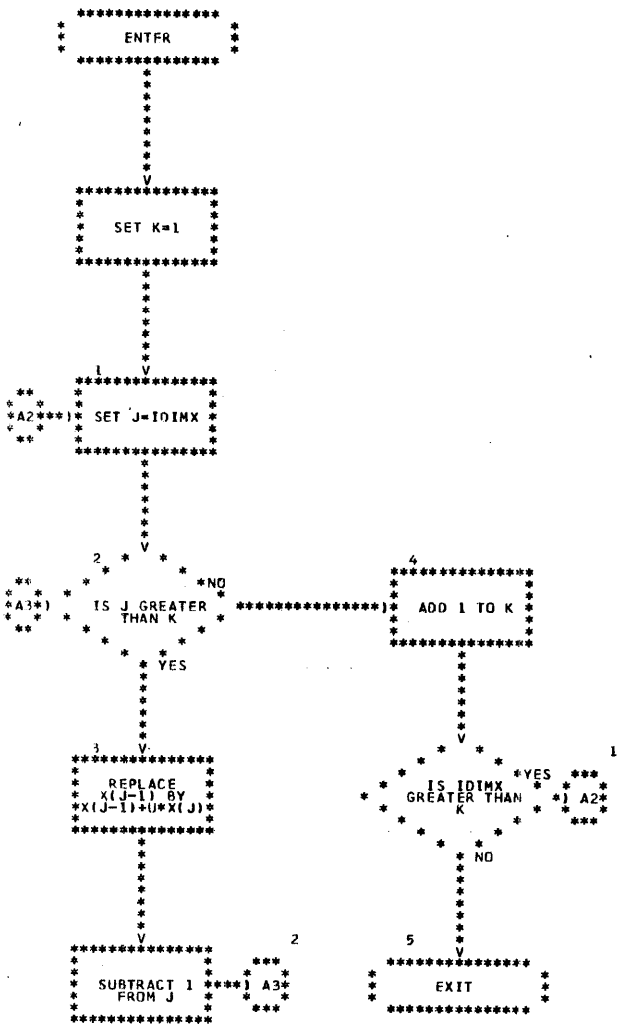
SUBROUTINE PVAL



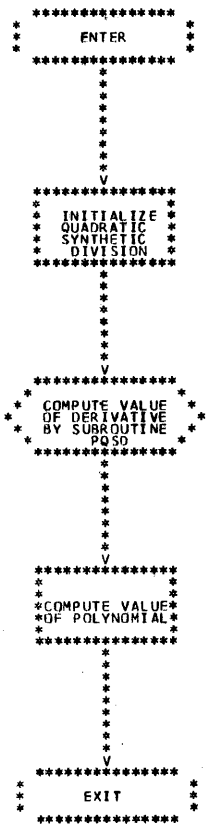
SUBROUTINE PVSUB



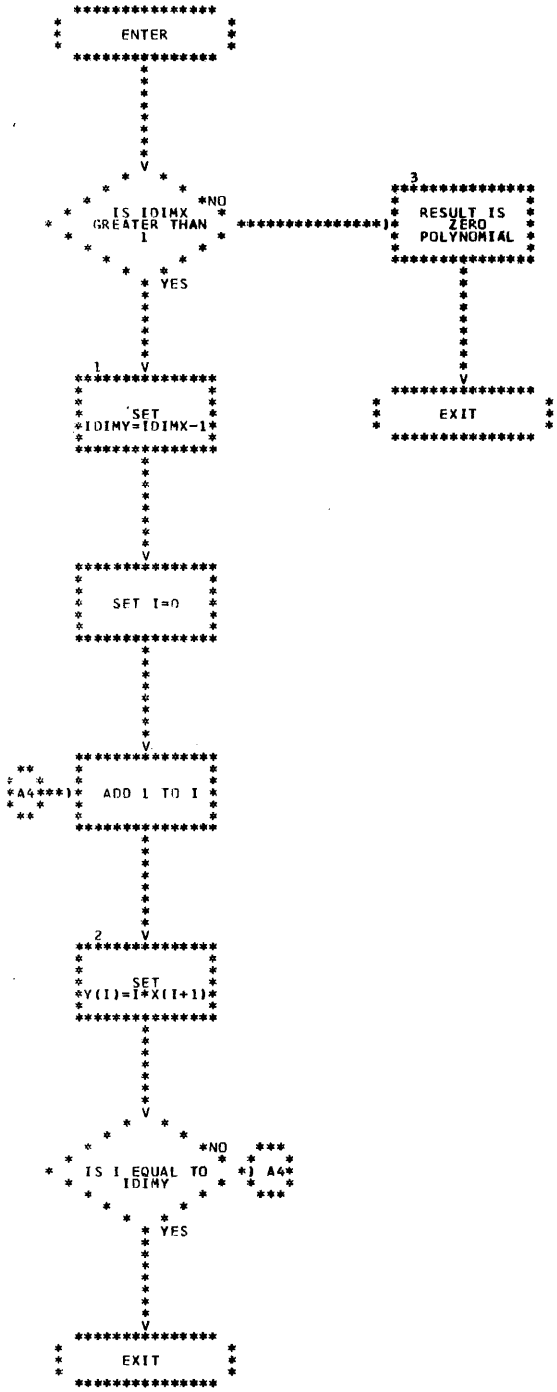
SUBROUTINE PCLD

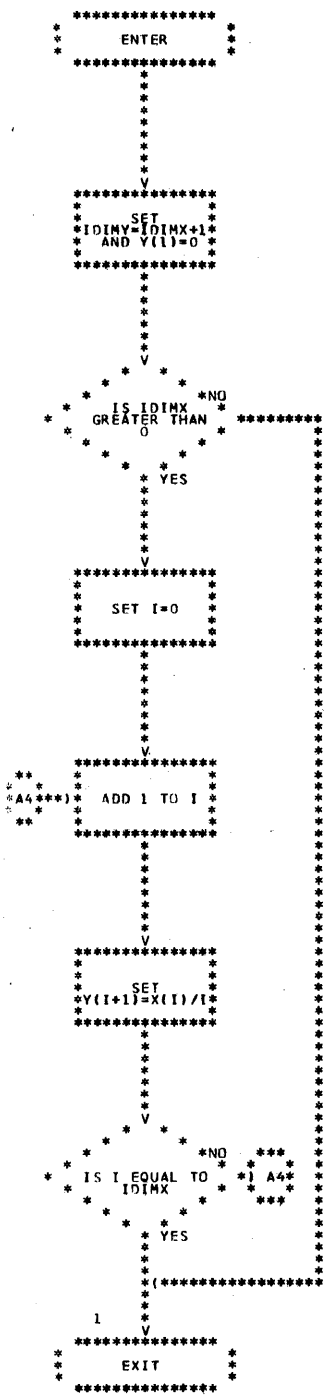


SUBROUTINE PILD

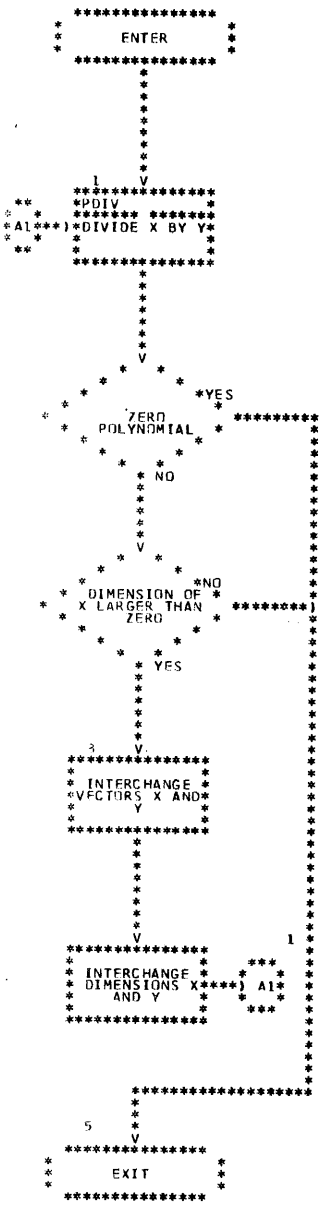


SUBROUTINE POER




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

SUBROUTINE PGCD



SUBROUTINE PNORM