SOFTWARE BULLETIN

SERIES 200

STATISTICS PACKAGE D

GENERAL SYSTEM: SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT: Statistics Package D: A Set of Five Programs that Enable the User to Perform Various Statistical Analyses on Numerical Data.

CHI-SQUARE D
LEAST SQUARES CURVE FITTING D
MEAN, VARIANCE, AND CORRELATION D
STEPWISE REGRESSION ANALYSIS D
RANDOM NUMBER GENERATOR D

DATE: October 15, 1965 FILE NO. 122.7005.001D.00.01

8512
31065

PREFACE

The Statistics Package D programs are written in Fortran language. For a description of Fortran language considerations, refer to the Fortran Compiler D Reference Handbook, File No. 123.1305.001D.00.00.

The number of variables permitted in most of these programs depends on the size of memory available when running the object program. The user may change I/O, Dimensional, and Data statements to fit the particular requirements of his program.

TABLE OF CONTENTS

SECTION I

CHI SQUARE D PROGRAM

I. Introduction

1.1 Purpose

To evaluate CHI SQUARE.

1.2 Method

$$\chi^2 = \sum_{k=1}^{m} \frac{(O_k - E_k)^2}{E_k}$$

where O_k is the observed frequency.

E_k is the expected frequency.

m is the number of categories.

II. Usage as a Routine

2.1 Minimum System Components Required

Fortran Compiler D System.

2.2 Input

The input data is read from card reader. The first card of each problem must be a control card, followed by data cards.

2.2.1 First Control Card Format

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Problem ID.	An integer < 99999
11-20	No. of categories	
21-80	Not used	

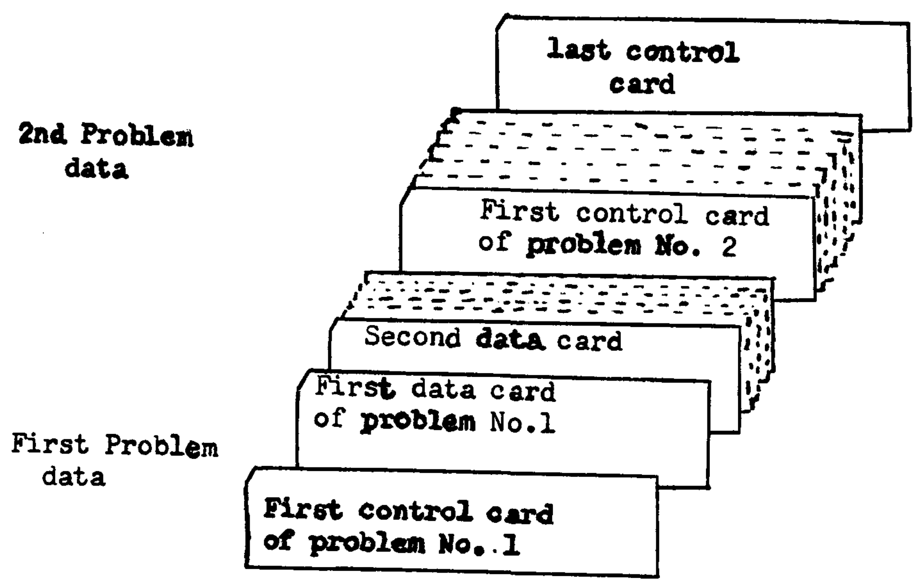
2.2.2 k^{th} Data Card Format

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Input Data ID	This number should be exactly the same as the number which appears in column 6-10 in the control card.
11-20	Category Number	An integer.
21-30	Observed frequency for k^{th} category.	Decimal
31-40	Expected frequency for k^{th} category.	Decimal
41-80	Not used	

2.2.3 Last Control Card Format (This card is needed only at the end. of a set of problems)

Columns	Contents	Description or Remark
1-5	Blank	
6-10	99999	
11-80	Not used	

2.2.4 Card Sequence Layout



2.3 Output

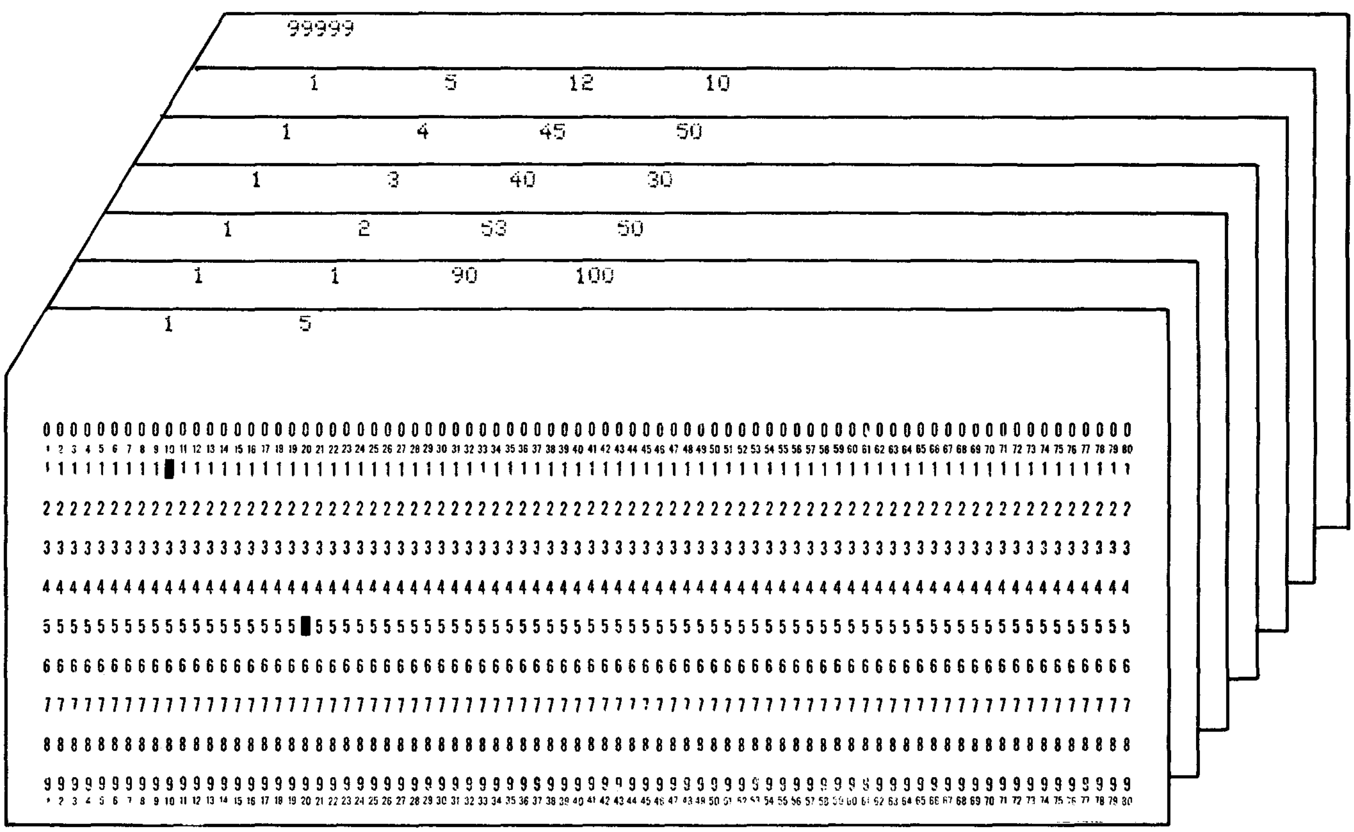
The output is printed on-line.

III. Example

3.1 Problem No. 1

Category	Observed Frequency	Expected Frequency
1	90	100
2	53	50
3	40	30
4	45	50
5	12	10

3.2 Input Card Format



3.3 Output

CHI SQUARE PROGRAM

PROBLEM NO. 1

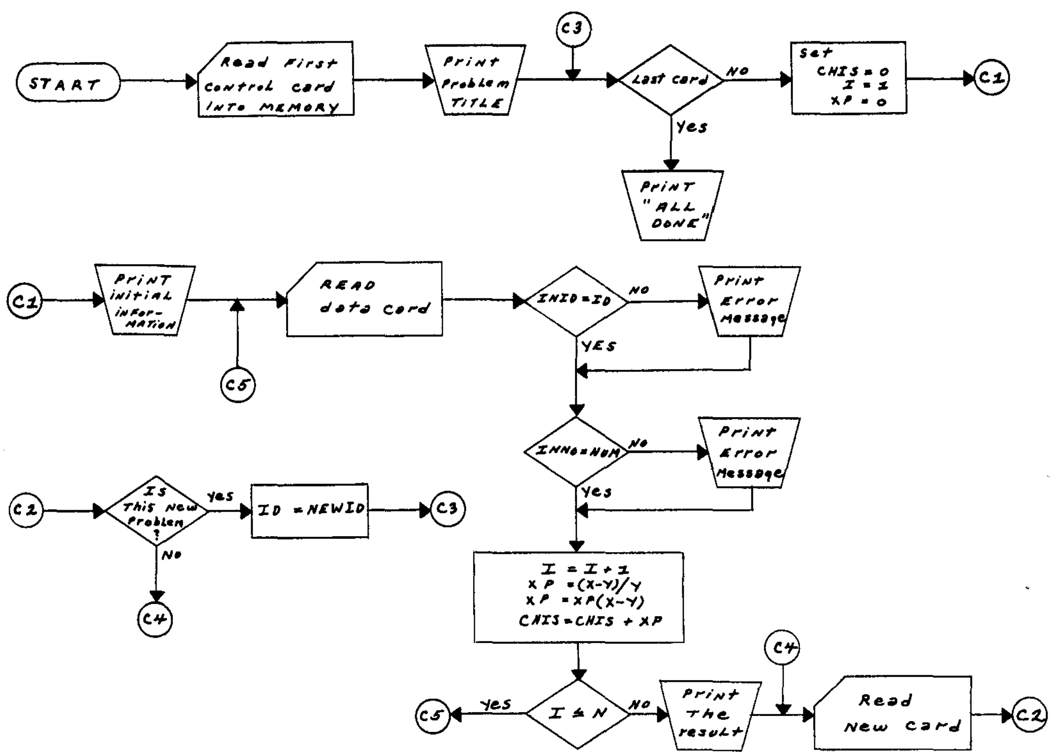
NO. OF CATEGORIES = 5

CHI SQUARE = 5.41333

ALL DONE.

IV. Disclaimer

This program has been checked but no guarantee is expressed or implied to the user concerning its functioning. Honeywell will cooperate in making functional those programs written by them. However, they assume no responsibility for programs distributed but not written by Honeywell.



SECTION II

LEAST SQUARES CURVE FITTING D PROGRAM

I. Purpose

Fits a polynomial of degree n to a set of m observations by the method of least squares. It is written in Fortran language.

II. Method

2.1 Symbol

n = degree of polynomial

m = number of observations

x_j = j-th observation of independent variable

y_j = j-th observation of dependent variable

$\hat{y}_j$ = predicted value of dependent variable

δ_j = deviation between observed and predicted value of j-th observation of dependent variables, i.e. $\delta_j = y_j - \hat{y}_j$

x_j^i = i-th power of x_j

2.2 Description

For a given n observations of (x_j, y_j) to fit a polynomial of the following form:

$$y = b_o + b_1x + b_2x^2 + \ldots + b_nx^n \qquad (1)$$

This program is used to determine the coefficients b_i's by the method of least squares.

The principle of Least Squares seeks to minimize the squares of the residuals

$$\Sigma\delta^2 = \sum_{j=1}^{m} (y_j - \sum_{i=o}^{n} B_i x_j^i)^2$$

where B_i is the estimated value of b_i in (1).

The normal equations are:

$$\begin{aligned}
b_o\Sigma x_j^o + b_1\Sigma x_j + \ldots + b_n\Sigma x_j^n &= \Sigma y_j \\
b_o\Sigma x_j + b_1\Sigma x_j^2 + \ldots + b_n\Sigma x_j^{n+1} &= \Sigma x_j y_j \\
\vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad &\quad\ \vdots \\
b_o\Sigma x_j^n + b_1\Sigma x_j^{n+1} + \ldots + b_n\Sigma x_j^{2n} &= \Sigma x_j^n y_j
\end{aligned} \tag{2}$$

This is a set of (n+1) simultaneous linear equations in b_i which is solved by Gaussian Elimination in this program.

2.3 Matrix Operations

The augmented matrix which is the coefficient matrix of (2) with the right hand side as an additional column is denoted as $A = (a_{ij})$. All (n+1)(n+2) elements will be stored in memory.

The diagonal element is used as a pivot. Denote the pivot as $a_{k,k}$. The new matrix will be generated by the following algorithm:

$$a'_{ij} = \begin{cases} 1/a_{kk} & \text{for } i=j=k \qquad (3) \\ a_{kj}/a_{kk} & \text{for } i=k,\ j\neq k \qquad (4) \\ a_{ij} - a_{ik}a_{kj}/a_{kk} & \text{for } i\neq k,\ j\neq k \qquad (5) \\ -a_{ik}/a_{kk} & \text{for } i\neq k,\ j=k \qquad (6) \end{cases}$$

where a_{ij} is the element of the new matrix.

2.4 Estimates of b_i

After n+1 iterations of the matrix operation in 2.3, the last column gives the estimates

2.5 Calculation of predicted values and deviations.

The predicted value for the dependent variable is

$$\hat{y}_j = \sum_{i=o}^{n} B_i x_j^i$$

The deviation between the actual and predicted values of the dependent variable is

$$\delta_j = y_j - \hat{y}_j$$

III. Minimum System Components Required

3.1 Fortran Compiler D

3.2 One work tape

IV. Input

The input data is read from the card reader. The format will be given in detail in section 8.

V. Output

All output is printed on-line. A sample output printout will be shown in 9.3.

VI. Estimate of Maximum Degree

$(N+1)(N+2)(K+2) \leq M - 13,600$

where N = degree of polynomial

K = length of mantissa of floating point numbers

M = memory configuration

As an example: Assume K = 10

M	N
16K	11
20K	18
24K	27
28K	32
32K	37

This version is written for 16K with N=12, precision 10. In order to adapt this to another memory size, the dimension of V must be changed in the "common" statement of each chain. Example: For 32K, precision 10, the dimension of V should be V(N+1, N+2), i.e., V(39, 40).

VII. Preparation of Input Data

7.1 Size of independent variable

The smallest value of the independent variable x_j must satisfy the following condition

$$x_j^{2n} > 10^{-99}$$

For example, to fit a polynomial of degree 16, the smallest value x_j must be greater than 10^{-3}, roughly.

The largest value of the independent variable x_j must satisfy the condition

$$\sum_{j=1}^{m} x_j^{2n} \leq 10^{99}$$

For example, for a given 100 observations of (x_j, y_j) to fit a polynomial of degree 16, the largest value x_j should be less than 10^3, roughly.

7.2 Number of problem

A maximum of 99998 problems can be run at one time.

7.3 Number of observations

The number of observations that can be handled is one less than 10^5 and subject to the condition that

$$\sum_{j=1}^{m} x_j^{2n} < 10^{99}$$

which is mentioned in 8.2.

7.4 Card Format

8.4.1 First control card of each problem

Columns	Contents	Description or Remark
1-5	Blanks	
6-10	Problem ID.	An integer <99999
11-15	Degree of Polynomial	An integer, right justified
16-20	Total Number of Observations	An integer

Columns	Contents	Description or Remark
21-24	Blanks	
25	Blank or 1	If "1", program will suppress matrix elements printout.
26-80	Not used	

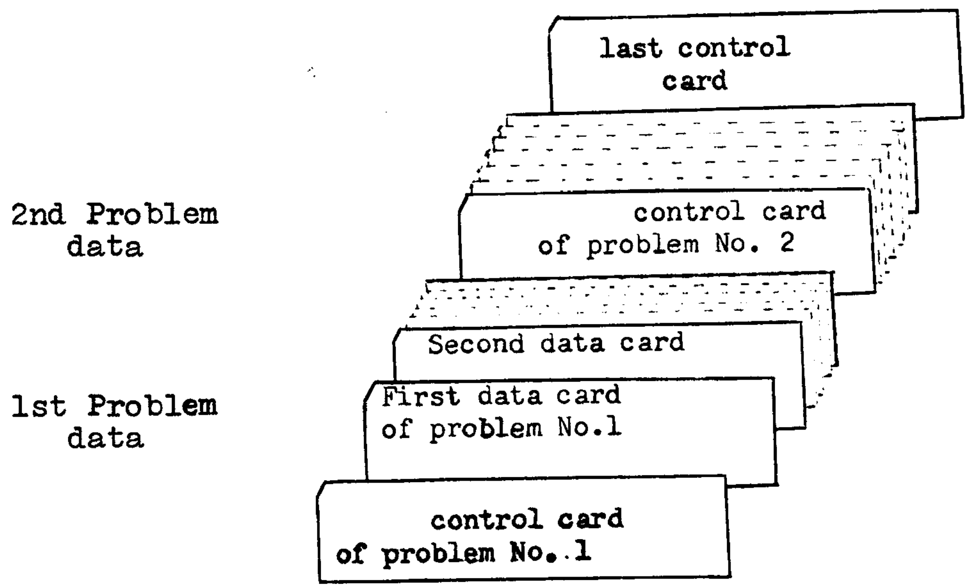
7.4.2 Data card

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Input Data ID	This no. should be exactly the same as the no. which appears in Column 6-10 in the Control Card.
11-20	Observation No.	An integer, 1 stands for first observation, 2 for second observation, etc. Order is irrelevant.
21-30	x_j	Decimal, value of independent variable. The decimal point is assumed at extreme right, if there is no decimal point present.
31-40	y_j	Decimal, value of dependent variable. Decimal point is the same as x_j field.
41-80	Not used	

7.4.3 Last control card

Columns	Contents	Description or Remark
1-5	Blank	
6-10	99999	An integer.
11-80	Not used	

7.4.4 Card Sequence Layout



VIII. Sample Problem

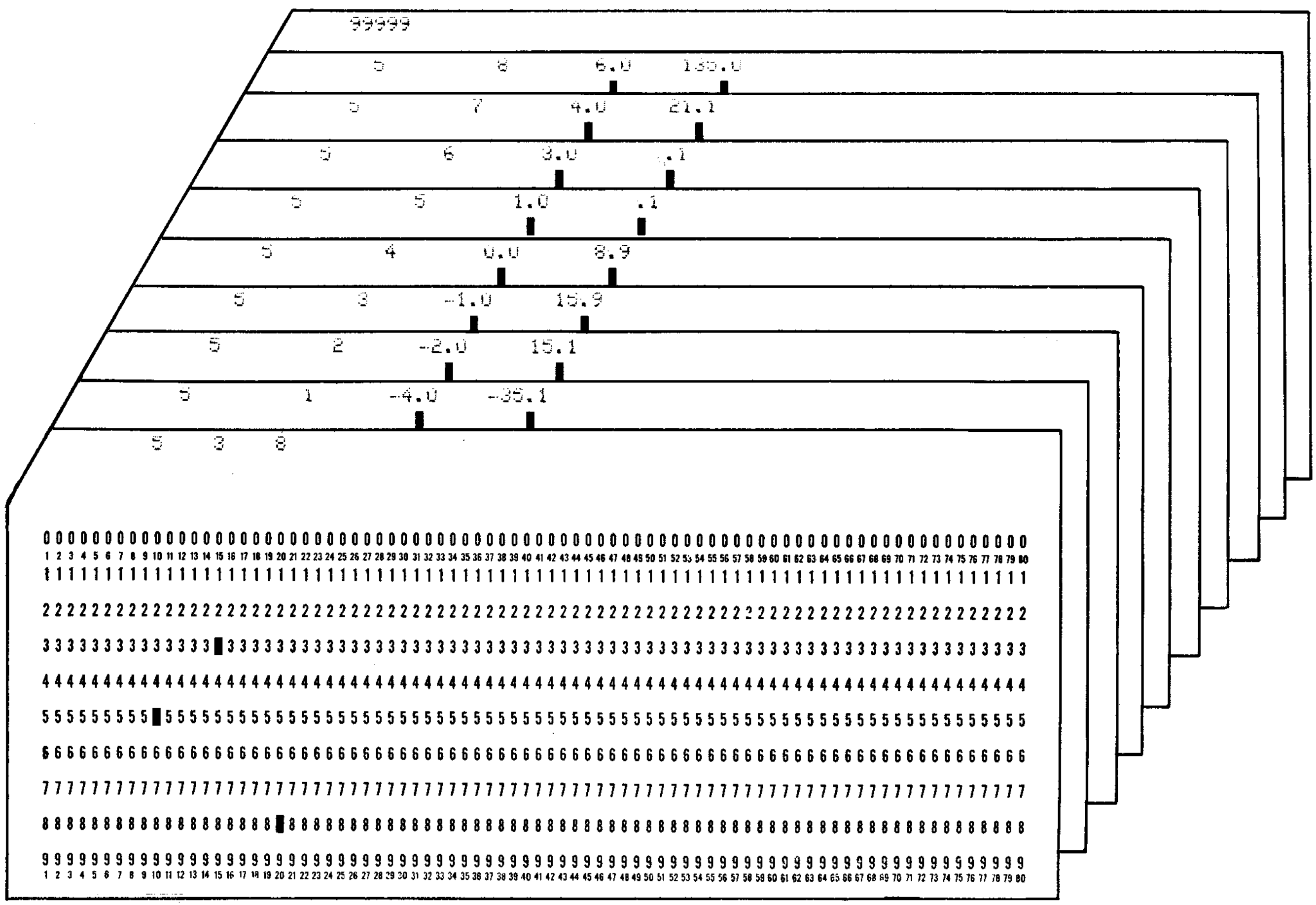
8.1 Find by the method of least squares a formula of the type

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3$$

which will fit the following data (homogeneous)

x	-4	-2	-1	0	1	3	4	6
y	-35.1	15.1	15.9	8.9	.1	.1	21.1	135.0

8.2 Input Data



LEAST SQUARES CURVE FITTING PROGRAM (LESCUF)

PROBLEM NO. 5

NO. OF OBSERVATIONS = 8

DEGREE OF POLYNOMIAL = 3

ELEMENTS OF A-MATRIX

A(1, 1) = 0.800000000E 01	A(1, 2) = 0.700000000E 01	A(1, 3) = 0.830000000E 02
A(1, 4) = 0.235000000E 03	A(1, 5) = 0.161100000E 03	A(2, 2) = 0.830000000E 02
A(2, 3) = 0.235000000E 03	A(2, 4) = 0.190700000E 04	A(2, 5) = 0.989100000E 03
A(3, 3) = 0.190700000E 04	A(3, 4) = 0.798700000E 04	A(3, 5) = 0.471330000E 04
A(4, 4) = 0.556430000E 05	A(4, 5) = 0.326229000E 05	A(,)

CONSTANT TERM = 0.901104386E 01

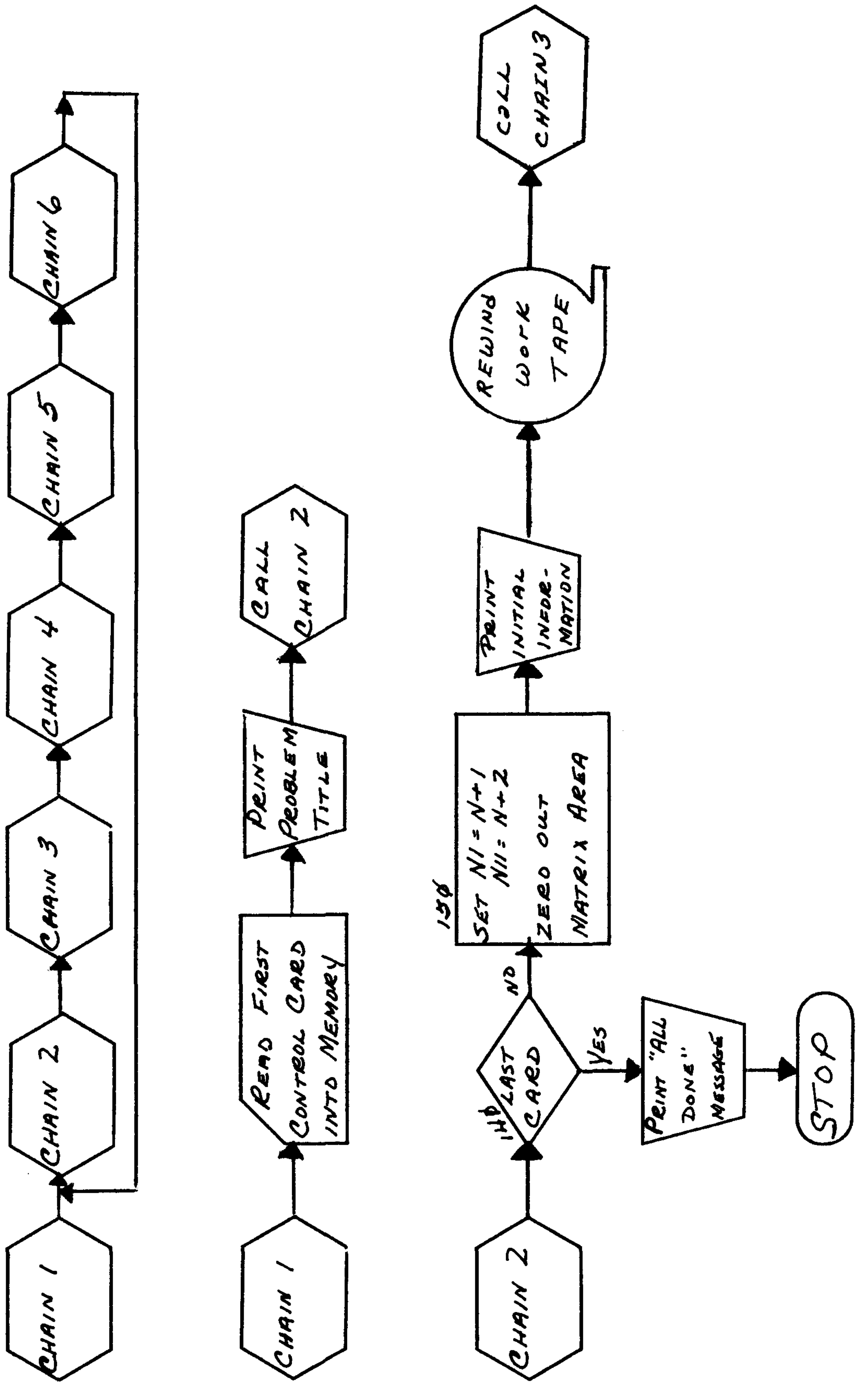
POWER	COEFFICIENT
1	-0.896614319E 01
2	-0.100009408E 01
3	0.999074298E 00

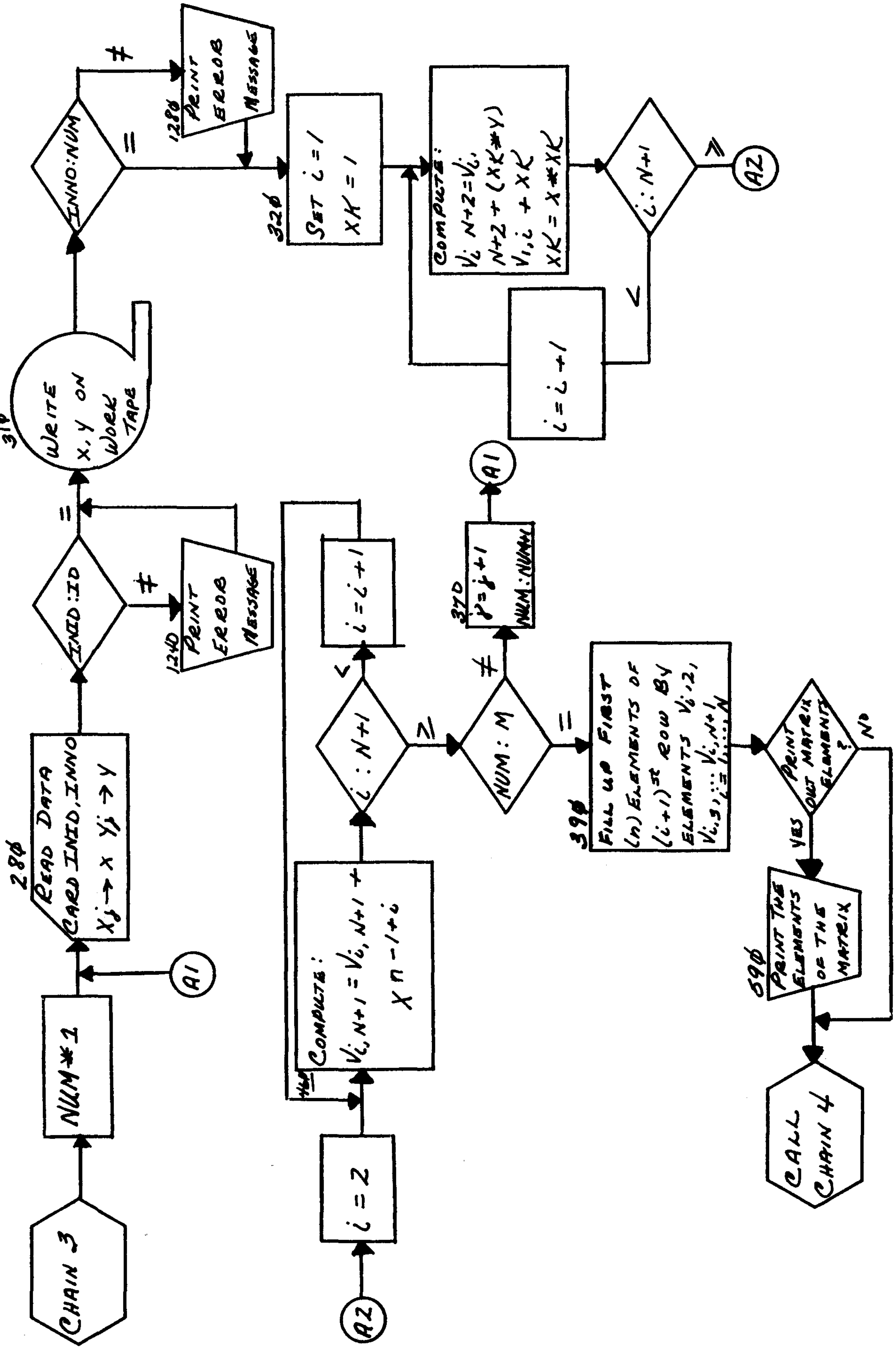
PREDICTED VS ACTUAL RESULTS

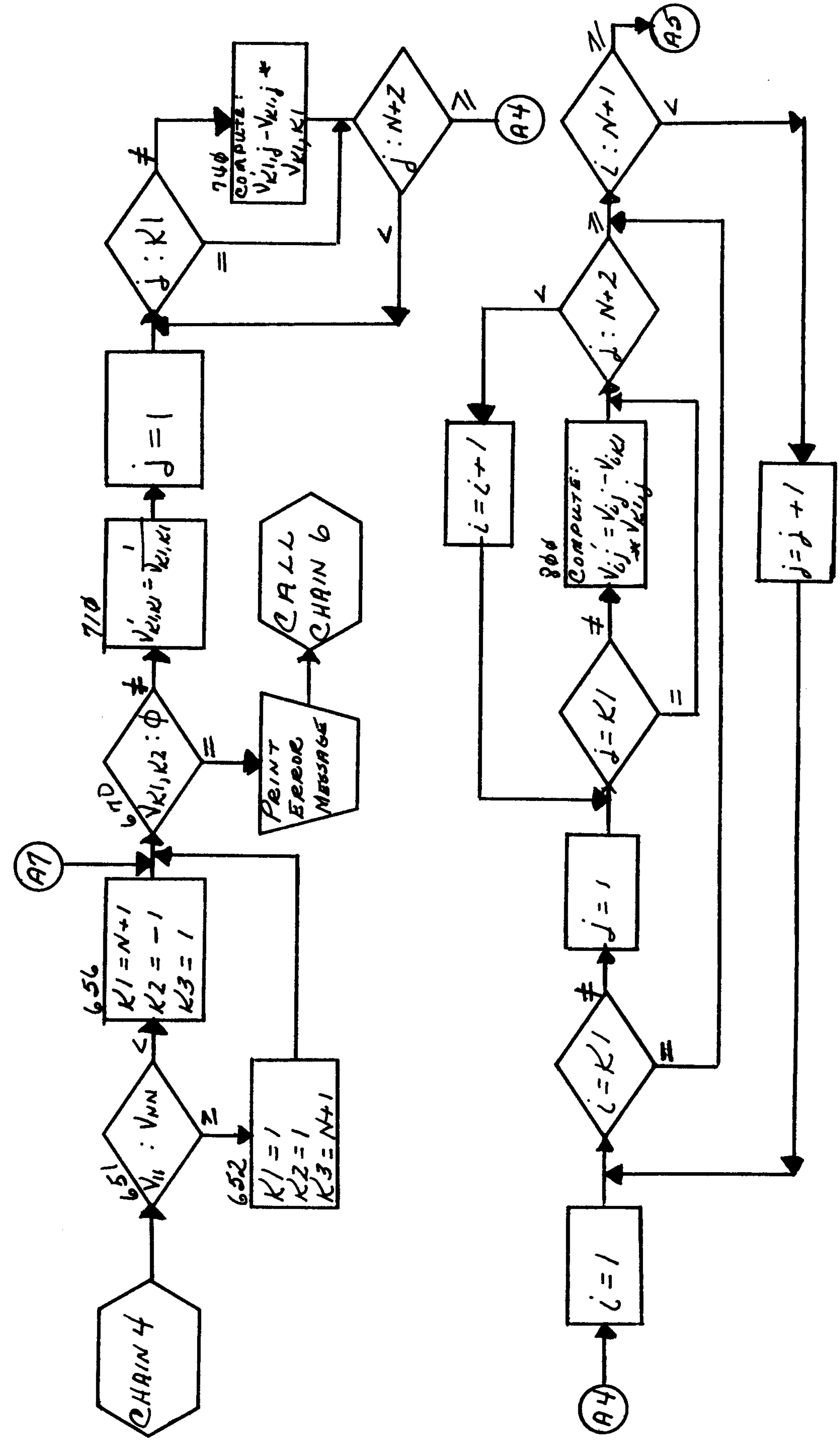
RUN NO.	VALUE OF X	ACTUAL	PREDICTED	DEVIATION
1	-4.000000	-35.100000	-35.066644	-.033356
2	-2.000000	15.100000	14.950360	.149640
3	-1.000000	15.900000	15.978019	-.078019
4	.000000	8.900000	9.011044	-.111044
5	1.000000	.100000	.043881	.056119
6	3.000000	.100000	.086774	.013226
7	4.000000	21.100000	21.085721	.014279
8	6.000000	135.000000	135.010846	-.010846

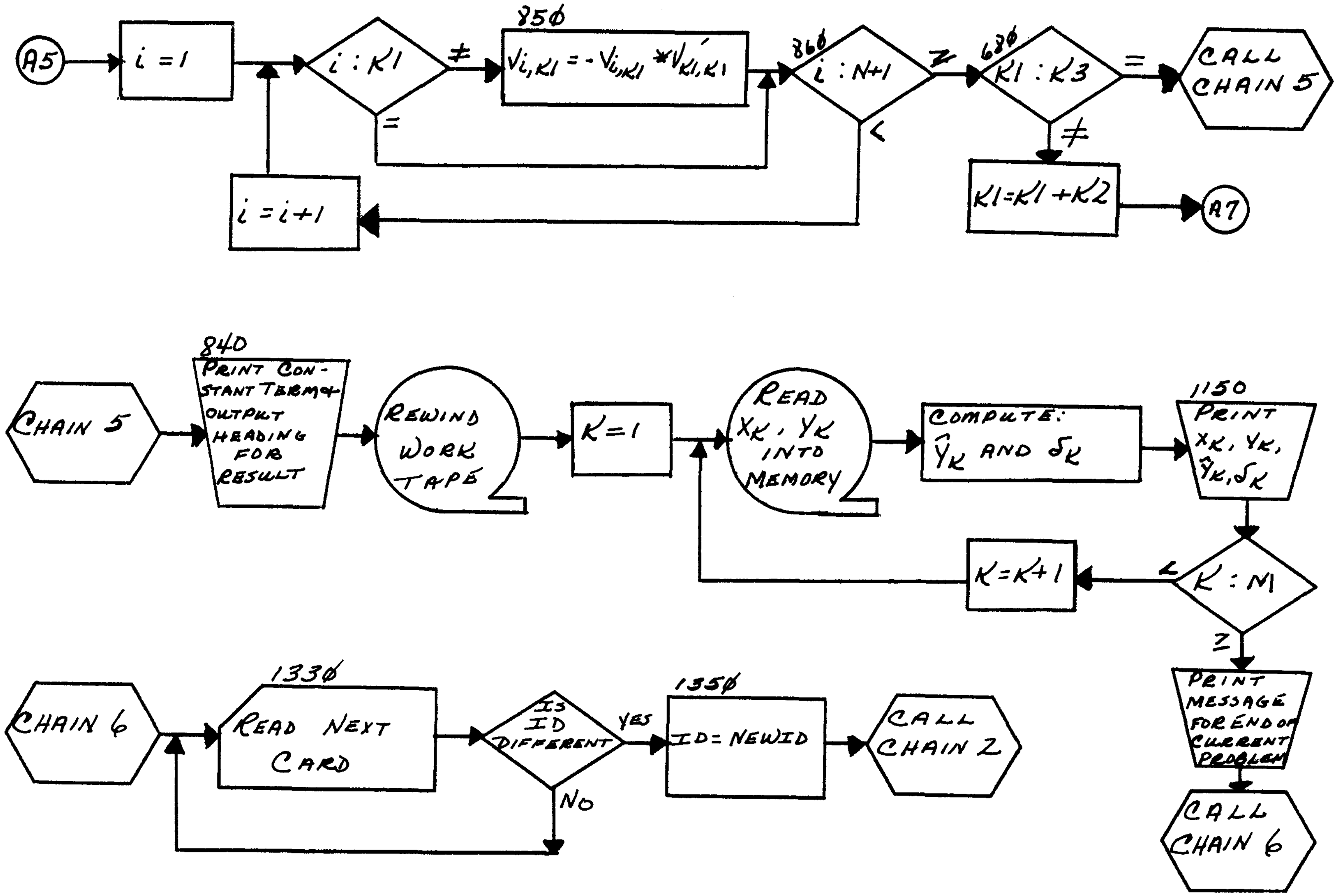
END OF CALCULATION, PROBLEM NO. 5

ALL DONE. THANK YOU.


CHAIN 1
CHAIN 2
CHAIN 3
CHAIN 4
CHAIN 5
CHAIN 6
CHAIN 1
READ FIRST CONTROL CARD INTO MEMORY
PRINT PROBLEM TITLE
CALL CHAIN 2
CHAIN 2
140
LAST CARD
YES
NO
PRINT "ALL DONE" MESSAGE
STOP
150
SET N1 = N+1
N11 = N+2
ZERO OUT MATRIX AREA
PRINT INITIAL INFOR-MATION
REWIND WORK TAPE
CALL CHAIN 3


CHAIN 3
NUM*1
A1
28∅
READ DATA CARD INID, INNO
Xj → X Yj → Y
INID:ID
=
≠
124∅
PRINT ERROR MESSAGE
31∅
WRITE X,Y ON WORK TAPE
INNO:NUM
=
≠
128∅
PRINT ERROR MESSAGE
32∅
SET i = 1
XK = 1
COMPUTE:
XK = X*XK
i : N+1
≥
A2
<
i = i+1
A2
i = 2
COMPUTE:
Vi,N+1 = Vi,N+1 +
X n-1+i
i : N+1
<
i = i+1
≥
NUM : M
≠
37∅
=
A1
39∅
FILL UP FIRST (n) ELEMENTS OF (i+1)st ROW BY ELEMENTS
PRINT OUT MATRIX ELEMENTS ?
YES
NO
59∅
PRINT THE ELEMENTS OF THE MATRIX
CALL CHAIN 4


CHAIN 4
651
$V_{11} : V_{NN}$
652
K1 = 1
K2 = 1
K3 = N+1
656
K1 = N+1
K2 = −1
K3 = 1
A7
670
$V_{K1,K2} : \phi$
PRINT ERROR MESSAGE
CALL CHAIN 6
710
$V'_{K1,K1} = \frac{1}{V_{K1,K1}}$
j = 1
j : K1
740
COMPUTE:
$V'_{K1,j} = V_{K1,j} * V_{K1,K1}$
j : N+2
A4
A4
i = 1
i = K1
j = 1
j = K1
800
COMPUTE:
$V'_{i,j} = V_{i,j} - V_{i,K1} * V_{K1,j}$
j : N+2
i = i + 1
i : N+1
j = j + 1
A5


A5
i = 1
i : K1
≠
=
850
i = i+1
860
i : N+1
≥
<
680
K1 : K3
=
≠
CALL CHAIN 5
K1=K1+K2
A7
CHAIN 5
840
PRINT CONSTANT TERM & OUTPUT HEADING FOR RESULT
REWIND WORK TAPE
K=1
READ XK, YK INTO MEMORY
COMPUTE: ŶK AND δK
1150
PRINT XK, YK, ŶK, δK
K : M
<
≥
K=K+1
PRINT MESSAGE FOR END OF CURRENT PROBLEM
CALL CHAIN 6
CHAIN 6
1330
READ NEXT CARD
IS ID DIFFERENT
YES
NO
1350
ID= NEWID
CALL CHAIN 2

SECTION III

MEAN, VARIANCE AND CORRELATION D PROGRAM

I. Purpose

This program is used for computing mean, variance, covariance, standard deviation and correlation coefficient of the variables which are stored in two groups on the tape, the X-group and Y-group.

II. Method

2.1 Symbol

m = number of observations

p = number of variables in X-group

q = number of variables in Y-group

x_{ij} = i-th observation of variable x_j

y_{ij} = i-th observation of variable y_j

z_{ij} = i-th observation of variable z_j

$x_j = z_j$ (for $1 \leq j \leq p$) j-th variable in X-group

$y_k = z_{k+p}$ (for $1 \leq k \leq q$) k-th variable in Y-group

$\bar{z}_j$ = mean of variable z_j

σz_j^2 = variance of z_j

σz_j = standard deviation of z_j

$cov(x_j, y_k)$ = covariance of x_j and y_k

γ_{jk} = coefficient of correlation between x_j and y_k

2.2 Formula

For each variable, we compute

$$\bar{z}_j = \Sigma z_{ij}/m$$

$$\sigma z_j^2 = (\Sigma\ z_{ij}^2 - m\bar{z}_j^2)/(m-1)$$

$$\sigma z_j = \sqrt{\sigma z_j^2}$$

For each pair of variables, i.e. one of which is in the X-group and one in Y-group, compute

$$\mathrm{cov}(x_j,\ y_k) = (\sum_i x_{ij}\ y_{ik} - m\bar{x}_j\ \bar{y}_k)/(m-1)$$

and $\gamma_{jk} = \mathrm{cov}(x_j,\ y_k)/(\sigma x_j \cdot \sigma y_k)$

III. Minimum System Components Required

3.1 Fortran Compiler D System.

IV. Input

The input data is read from the card reader. The input data format will be given in detail in section 8.6.

V. Output

All output is printed on-line. A sample output will be shown in section 9.

VI. Subroutine

The square root routine must be supplied from the library of subroutines.

VII. High Speed Memory Requirement

	Memory Space	Notation Used in Program	Remarks
A	p · q	VECTOR(I,J)	matrix A space has multiple uses as: 1) sum of cross products 2) covariance of variables 3) correlation of coefficients
x_i	p	Z(I) for $1 \leq I \leq p$	Z(I) space has multiple uses as: 1) inputs x_i, y_i 2) residual sum of squares
y_i	q	Z(I) for $p \leq 1 \leq I \leq p+q$	
$\sum_j X_{ji}$	p	SUM Z(I) for $1 \leq I \leq p$	
$\sum_j Y_{ji}$	q	SUM Z(I) for $p+1 \leq I \leq p+q$	
$\sum_j X_{ji}^2$	p	SUM ZZ(I) for $1 \leq I \leq p$	
$\sum_j Y_{ji}^2$	q	SUM ZZ(I) for $p+1 \leq I \leq p+q$	
σ_i	p	SIGMA(I) for $k \leq I \leq p$	
σ_j	q	for $p+1 \leq I \leq p+q$	
$\bar{X}_i$	p	Z BAR(I) for $1 \leq I \leq p$	
$\bar{Y}_i$	q	for $p+1 \leq I \leq p+q$	

VIII. Preparation of Input Data

8.1 Size of data

If the largest sum of squares or the sum of the cross products is greater than 10^9, the printout will lose the most significant digits. However, the statistical results will be correct. The size of data should depend on the number of observations.

8.2 Number of variables

$[p \cdot q + 5(p+q)]\,(d+2) \leq M - 14{,}000$

where M is the memory size

p is the number of variable in X-group

q is the number of variable in Y-group

d is the number of digits accuracy.

As an example: $p = q$ $d = 10$

Memory Size	Estimated Maximum Value For p
16K	10
24K	25
32K	34

* Remark: This version is written for 16K machine with precision 10. In order to adapt to other size machines, the arrays in the "common" statement of each chain must be changed according to the above estimated formula.

8.3 Number of problems

A maximum of 99998 problems can be run at one time.

8.4 Number of data points

The number of data points (observations) that can be handled is less than 10^6 and is limited by the restriction that the largest sum of squares must be less than 10^9. This program handles an equal number of data points for each variable.

8.5 Decimal point

The decimal point of input data need not be entered if the number is an integer. The decimal point will be assumed to be at the extreme right. For example, columns 21 to 30 are used for the first variable, say x_1, within a data card. If a "15" is punched in columns 29 and 30, then x_1 is 15. If "15" is punched in columns 28 and 29, then x_1 will be 150. A number with a decimal point can be punched anywhere within the specified columns for that variable. For example a "25.4" in columns 19 to 23 is the same as if punched in 26 to 30.

8.6 Tape format

8.6.1 First control card of each problem

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Problem Identification	An integer <99999
11-15	Number of observations	An integer
16-20	Number of variables in X- and Y-groups.	An integer
21-25	Number of variables in Y-group.	An integer * If this no. is zero, the program will not calculate correlation.
26-80		Not used.

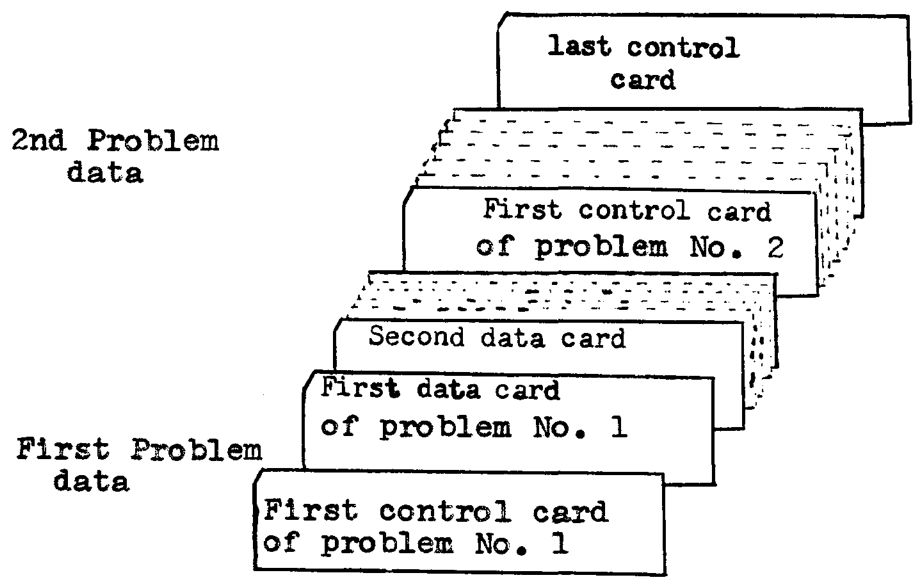
8.6.2 Data card

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Input Data ID	This no. should be exactly the same as the no. which appears in Col. 6-10 in the Control Card.
11-15	Logical Sequence No.	An integer. 1 stands for first 5 variables data card, 2 for second 5 variables data card, etc., if the number of variables is less than 5, a 1 is sufficient for all data cards.
16-20	Observation No.	An integer. 1 stands for first observation, 2 for second observation, etc.
21-30	z_1	Decimal, value of first data within a card
31-40	z_2	Decimal, value of second data within a card
41-50	z_3	Decimal, value of third data within a card
51-60	z_4	Decimal, value of fourth data within a card
61-70	z_5	Decimal, value of fifth data within a card
71-80		Not used

8.6.3 Last Control Card

Columns	Contents	Description or Remark
1-5	Blanks	
6-10	The integer 99999	
11-80	Not used	

8.6.4 Card Sequence Layout



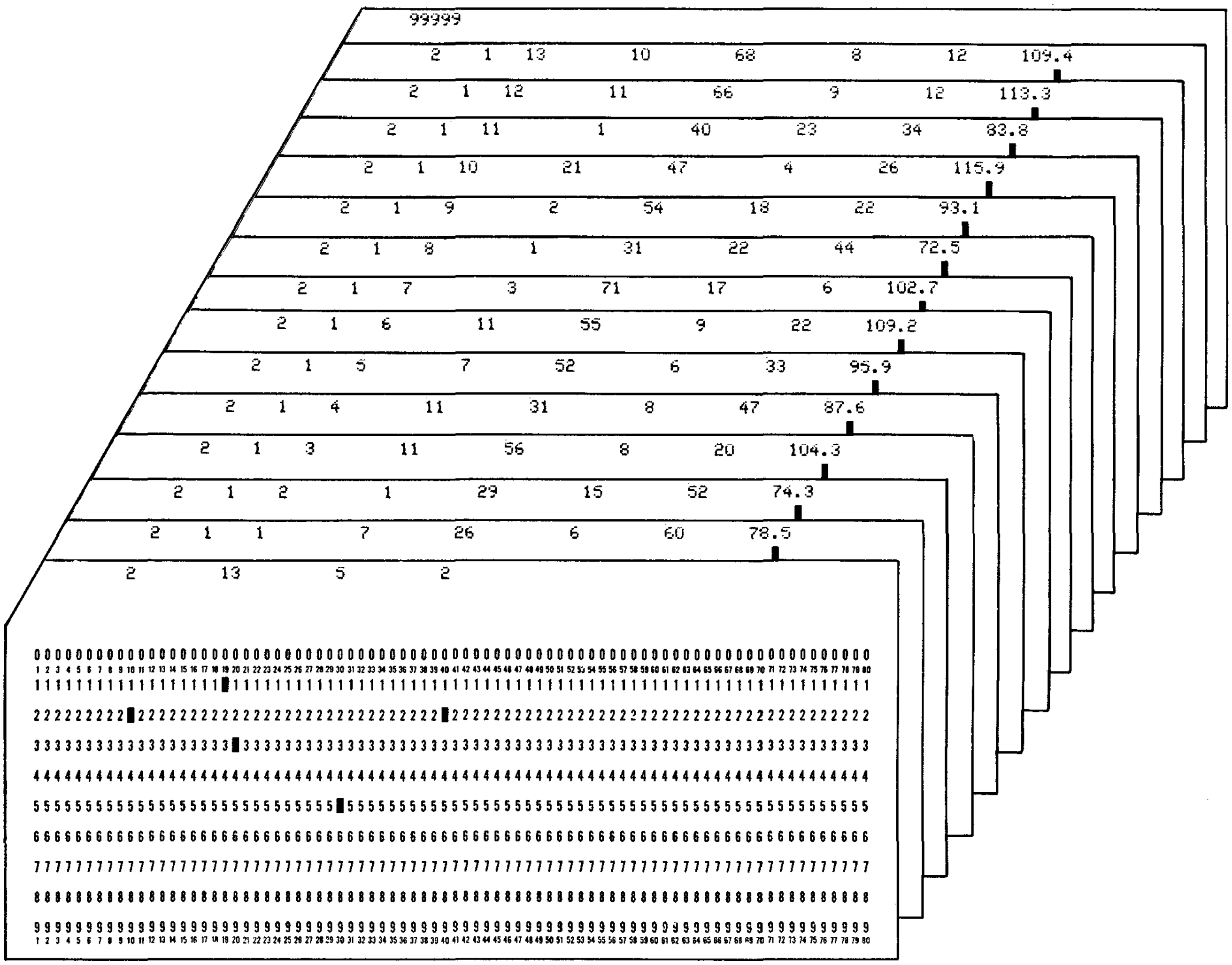
IX. Sample Problem

9.1 Problem 2

Given the following data, compute the mean and variance of each variable and covariance and correlation coefficient between the variables in X-group and Y-group.

Run No.	x_1	x_2	x_3	y_1	y_2
1	7.	26.	6.	60.	78.5
2	1.	29.	15.	52.	74.5
3	11.	56.	8.	20.	104.3
4	11.	31.	8.	47.	87.6
5	7.	52.	6.	33.	95.9
6	11.	55.	9.	22.	109.2
7	3.	71.	17.	6.	102.7
8	1.	31.	22.	44.	72.5
9	2.	54.	18.	22.	93.1
10	21.	47.	4.	26.	115.9
11	1.	40.	23.	34.	83.8
12	11.	66.	9.	12.	113.3
13	10.	68.	8.	12.	109.4

9.2 Input Data Cards For Card Image



9.3 Output Data

```
***STATISTICS PROGRAM - MEAN, VARIANCE AND CORRELATION***



 PROBLEM NO.          2

 NO. OF OBSERVATIONS =          13

 NO. OF VARIABLES IN X-GROUP =   3

 NO. OF VARIABLES IN Y-GROUP =   2

      VARIABLE            SUM             SUM OF SQUARES           MEAN

       X(  1)          97.000000          1139.000000           7.461538
       X(  2)         626.000000         33050.000000          48.153846
       X(  3)         153.000000          2293.000000          11.769231
       Y(  1)         390.000000         15062.000000          30.000000
       Y(  2)        1240.499999        121088.089997          95.423077

                             SUM OF CROSS PRODUCTS

 X(  1) VS Y(  1) =    2620.000000       X(  2) VS Y(  1) =   15739.000000
 X(  3) VS Y(  1) =    4628.000000       X(  1) VS Y(  2) =   10031.999999
 X(  2) VS Y(  2) =   62027.799999       X(  3) VS Y(  2) =   13981.499999

      VARIABLE          VARIANCE          STANDARD DEVIATION

       X(  1)          34.602564             5.882394
       X(  2)         242.141026            15.560881
       X(  3)          41.025641             6.405126
       Y(  1)         280.166667            16.738180
       Y(  2)         226.313590            15.043723

                             COVARIANCES OF VARIABLES

 X(  1) VS Y(  1) =     -24.166667       X(  2) VS Y(  1) =    -253.416667
 X(  3) VS Y(  1) =       3.166667       X(  1) VS Y(  2) =      64.663462
 X(  2) VS Y(  2) =     191.079487       X(  3) VS Y(  2) =     -51.519231

                             CORRELATION COEFFICIENTS

 X(  1) VS Y(  1) =       -.245445       X(  2) VS Y(  1) =       -.972955
 X(  3) VS Y(  1) =        .029537       X(  1) VS Y(  2) =        .730717
 X(  2) VS Y(  2) =        .816253       X(  3) VS Y(  2) =       -.534671


 END OF CALCULATION. PROBLEM NO.          2

ALL DONE. THANK YOU.
```

9.4 Special Comments on Error Conditions

If any or all of the following comments appear on output

9.4.1 DATA ID DOES NOT AGREE WITH CONTROL CARD. INID = xxxxx xxxxx NOC = xxxxx INNO = xxxxx

9.4.2 DATA CARD OUT OF LOGICAL SEQUENCE. CHECK. INID = xxxxx xxxxx NOC = xxxxx INNO = xxxxx

9.4.3 DATA CARD OUT OF LOGICAL SEQUENCE. INID = xxxxx xxxxx NOC = xxxxx INNO = xxxxx

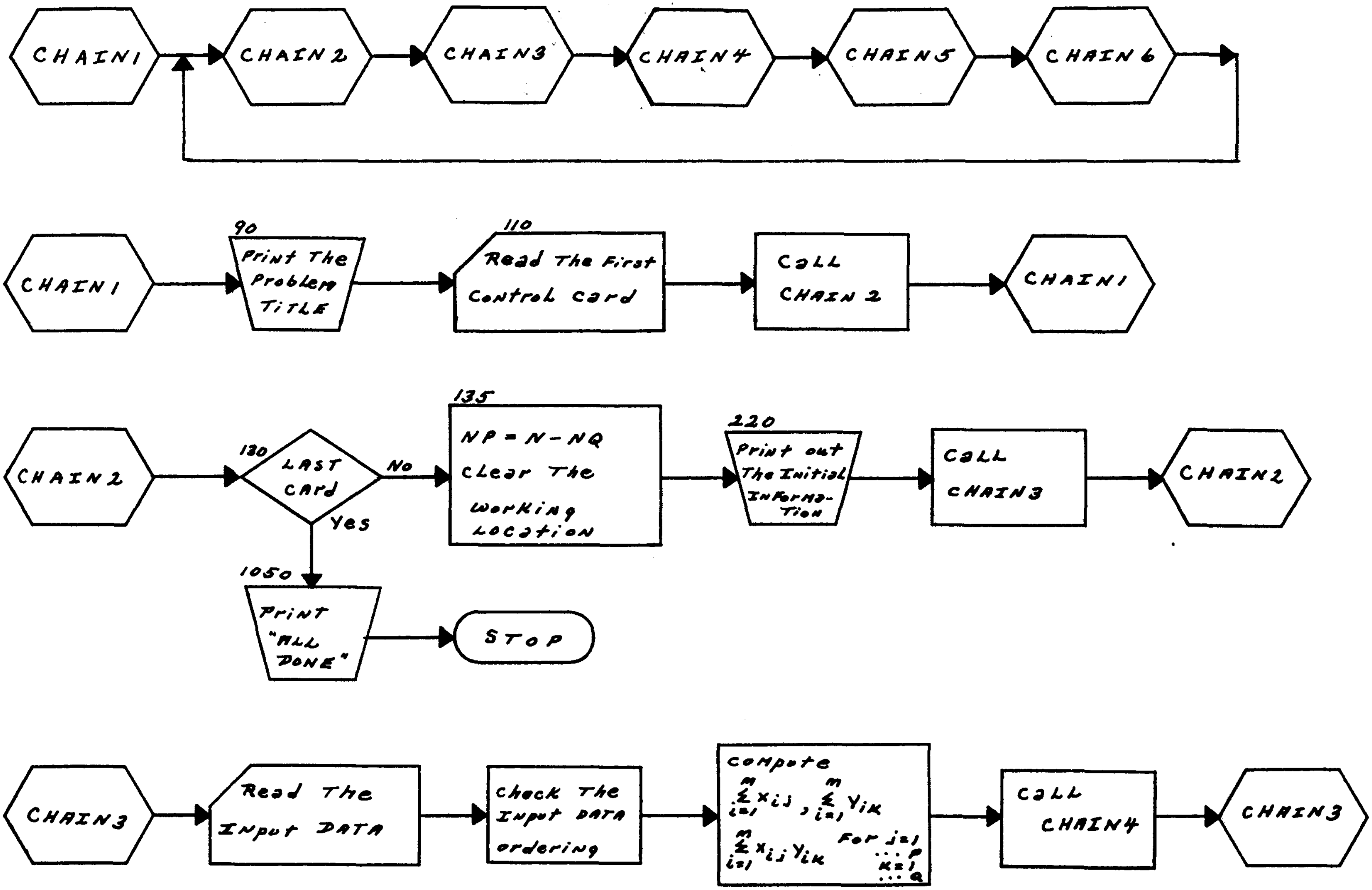
then these errors are caused by the mispunching of the identification number of test data of a problem or the misplacing of data cards. The program will continue but the user should recheck the input data deck. The numbers, INID, NOC, and INNO will help the program to pinpoint the mispunched card or misplaced card. After correcting the input data, the program should be re-run.

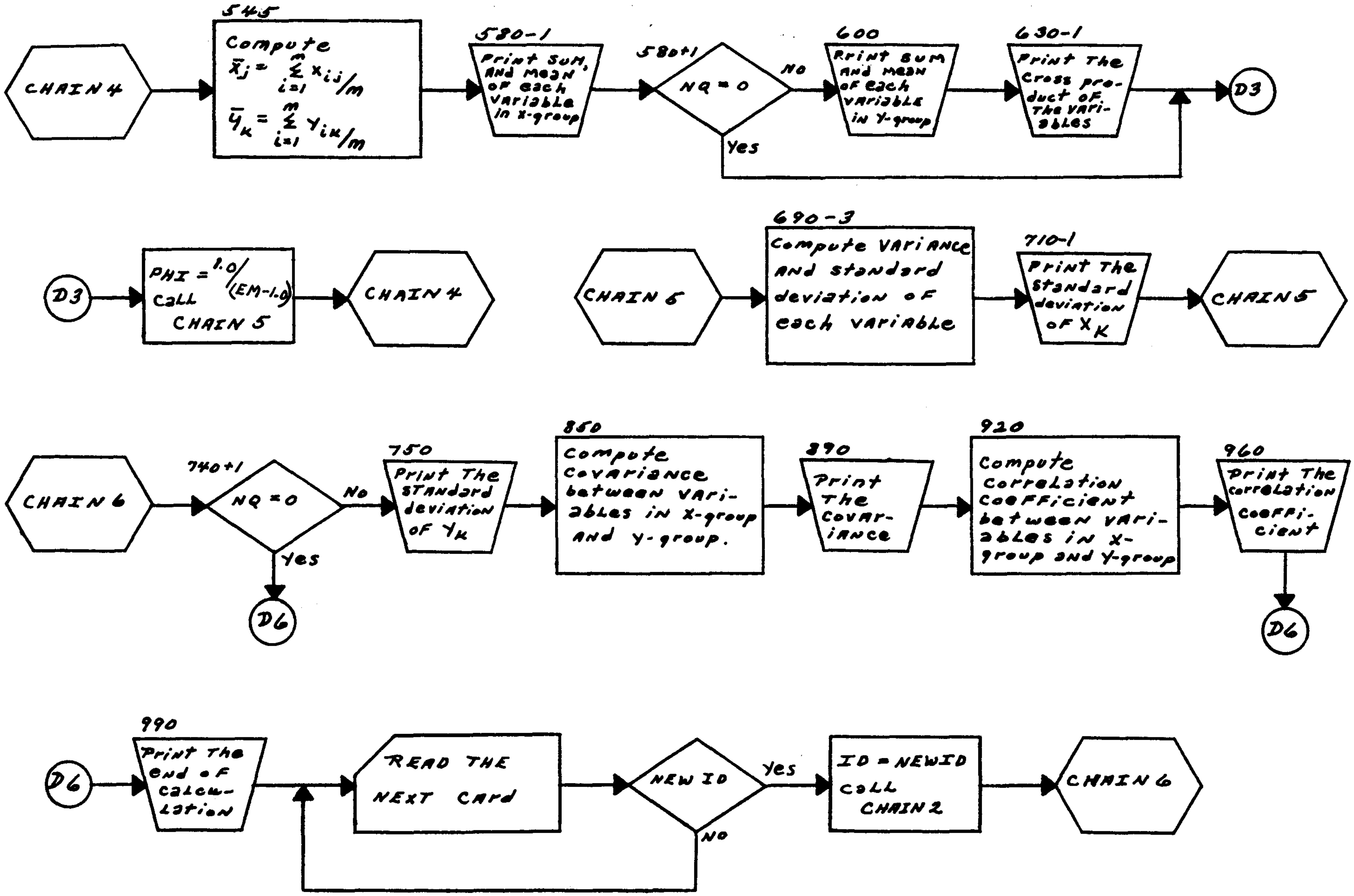
X. Timing

There are (m+1)(p+q) + pq additions, (m+3)(p+q) + 3pq multiplications, pq + 3 divisions and p + q square root subroutine calls.

XI. Disclaimer

This program has been checked but no guarantee is expressed or implied to the user concerning its functioning. Honeywell will cooperate in making functional those programs written by them. However, they assume no responsibility for programs distributed but not written by Honeywell.


CHAIN1
CHAIN2
CHAIN3
CHAIN4
CHAIN5
CHAIN6
CHAIN1
90
Print The Problem TITLE
110
Read The First Control card
Call CHAIN 2
CHAIN1
CHAIN2
130
LAST card
No
Yes
135
NP = N - NQ
Clear The working Location
220
Print out The Initial Informa-tion
Call CHAIN3
CHAIN2
1050
Print "ALL DONE"
STOP
CHAIN3
Read The Input DATA
Check The Input DATA ordering
compute
$\sum_{i=1}^{m} x_{ij}$, $\sum_{i=1}^{m} y_{ik}$
$\sum_{i=1}^{m} x_{ij} y_{ik}$ For j=1 ... p, k=1 ... q
Call CHAIN4
CHAIN3


CHAIN 4
545
Compute
$\bar{x}_j = \sum_{i=1}^{m} x_{ij}/m$
$\bar{y}_k = \sum_{i=1}^{m} y_{ik}/m$
580-1
Print sum, and mean of each variable in x-group
580+1
NQ = 0
No
Yes
600
Print sum and mean of each variable in y-group
630-1
Print the cross product of the variables
D3
D3
PHI = 1.0/(EM-1.0)
Call CHAIN 5
CHAIN 4
CHAIN 6
690-3
Compute variance and standard deviation of each variable
710-1
Print the standard deviation of X_K
CHAIN 5
CHAIN 6
740+1
NQ = 0
No
Yes
D6
750
Print the standard deviation of Y_K
850
Compute covariance between variables in x-group and y-group.
890
Print the covariance
920
Compute correlation coefficient between variables in x-group and y-group
960
Print the correlation coefficient
D6
D6
990
Print the end of calculation
READ THE NEXT CARD
NEW ID
Yes
No
ID = NEWID
Call CHAIN 2
CHAIN 6

SECTION IV

STEP-WISE MULTIPLE REGRESSION ANALYSIS D PROGRAM

I. Purpose

This program is used to find the best fit of an equation of the following form:

$$y = B_o + B_1 x_1 + B_2 x_2 + \dots + B_{n-1} x_{n-1} \qquad (1)$$

where y is the dependent variable and $x_1, x_2, \dots,$ are independent variables.

II. Method*

2.1 Symbols

n = number of independent variables and dependent variables
m = number of sets of observations
F1 = F value for entering the variable
F2 = F value for removing the variable
TOL = Value of tolerance, an arbitrary constant
x_{ik} = k-th observation of i-th variable
$x_{nk} = y_k$ = k-th observation of dependent variable
$\bar{x}_i$ = mean of i-th variable
$\bar{x}_n$ = mean of dependent variable
s_{ij} = residual sum of squares and cross products of i-th and j-th variable.
$\sigma_i = \sqrt{s_{ii}}$
γ_{ij} = simple correlation coefficient of i-th and j-th variable.
B_i = true value of coefficient of i-th variable
b_o = constant term of regression equation
b_i = estimated coefficient of i-th variable
s_y = standard error of dependent variable
s_{bi} = standard error of coefficient of i-th variable
$\hat{y}_k$ = predicted value of dependent variable for k-th observation.
δ_k = deviation between observed and predicted value of k-th observation of dependent variable, i.e. $y_k - \hat{y}_k = \delta_k$
N_{min}, N_{max} = subscripts of selected independent variables

* Ralston, A. and Wilf, H. S., Mathematical Method for Digital Computers, John Wiley, New York, 1960. pp. 191-203

N_{out} = subscripts of an independent variable just removed from regression equation

V_{min} = variance increase by deleting a variable N_{min}

V_{max} = variance reduction by adding a variable N_{max}

ϕ = the degree of freedom

2.2 Description:

Let y be an estimate of y; the estimating equation may then be expressed as:

$$y = b_o + b_1 x_1 + b_2 x_2 + \dots + b_{n-1} x_{n-1} \qquad (2)$$

where the b's are to be determined by means of available data.

For a given m sets of values of n variables, it is always possible to obtain b's, the estimates of the coefficients B's. Geometrically, the problem is one of finding the equation of a hyperplane which best fits a set of m points in n dimensions. The problem now is to find the set of b's in (2) that will minimize the sum $\sum_{k=1}^{m} \delta_k^2$.

In the step wise procedure, we obtain the regression equation by adding one variable at a time. The intermediate regression equations are:

$$\hat{y} = b'_o + b'_1 x_{k1} \qquad (3)$$

$$\hat{y} = b''_o + b''_1 x_{k1} + b''_2 x_{k2} \qquad (4)$$

$$\vdots$$

The variable added is that one which makes the greatest improvement in "goodness of fit". The coefficients are the best values when the regression equation is fitted by the specific variables which have already been entered in the equation. A variable, say x_{k2}, may be a significant variable in an early stage and thus enter the equation. But after several other variables have entered the regression equation, the x_{k2} may be insignificant. This variable x_{k2} will be removed from the regression equation before entering an additional variable. The final regression equation includes only those variables which are significant.

The b's with the superscripts are intermediate estimated values of the coefficients of variables which are in the regression equation at that stage. There is no relation between the b's with the lower and the higher superscripts. The b_k of equation (2) will be equal to the b with the superscript of the coefficient of x_2 in the final regression equation. And $b_i = 0$ for the corresponding i-th variable is not in the final regression equation. For example, we expect to have the estimate of y:

$$\hat{y} = b_o + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_5 x_5$$

for n = 6

If we have the final regression equation of the following form:

$$\hat{y} = b_o''' + b_1''' x_2 + b_2''' x_4 + b_3''' x_5$$

then $b_o = b_o'''$, $b_2 = b_1'''$, $b_4 = b_2'''$, $b_5 = b_3'''$

and $b_1 = b_3 = 0.$

2.3 Basic Calculations and Formulas

2.3.1 Calculation of sums of squares and cross products

Sums of variables $= \sum_{k=1}^{M} X_{ik}$ for i = 1, 2, ... n

Sums of squares and cross products $= \sum_{k=1}^{m} x_{ik} \; x_{jk}$ for i=1, 2, ... n; j=1, 2, ... n

2.3.2 Calculation of means and residual sums of squares and cross products

Means $\bar{x}_i = \sum_{k=1}^{m} x_{ik} / m \quad$ for i=1, 2, ... n (5)

Residual sum of squares and cross products

$$s_{ij} = \sum_{k=1}^{m} x_{ik} x_{jk} - \sum_{k=1}^{m} x_{ik} \sum_{k=1}^{m} x_{jk} / m$$

$$= \sum_{k=1}^{m} x_{ik} x_{jk} - \bar{x}_i \sum_{k=1}^{m} x_{jk} \quad \text{for } i=1, 2, \ldots n;\ j=1, 2, \ldots n \quad (6)$$

2.3.3 Calculation of partial correlation coefficients

$$\sigma_i = \sqrt{s_{ii}} \quad \text{for } i = 1, 2, \ldots n \quad (7)$$

$$\gamma_{ij} = s_{ij} / \sigma_i \sigma_j \quad \text{for } i = 1, 2, \ldots n,\ j=1,2,\ldots n \quad (8)$$

$$\gamma_{ji} = \gamma_{ij} \text{ and } \gamma_{ii} = 1.0 \quad \text{for } i=1, 2, \ldots n \quad (9)$$

2.3.4 Normal Equations

By minimizing the sum $\sum_{k=1}^{m} \delta_k^2$, we have the normal equations

$$\sum_{j=1}^{n-1} \gamma_{ij} b_j = \gamma_{in} \quad \text{for } i=1, 2, \ldots, n-1 \quad (10)$$

They comprise a set of n-1 simultaneous linear equations in b_j which can be solved by any method. In this program we adopt the Gaussian elimination method. In the stepwise procedure, we obtain the final regression by adding one variable at a time. How to enter or remove a variable from the regression will be discussed in 2.3.5.a.

2.3.5 Calculation of new matrix

The original matrix consists of coefficients of correlation, i.e. $A = (\gamma_{ij})$. It is symmetric. All n^2 elements will be stored in memory.

2.3.5.a. Calculation of V_i

V_i can be computed from matrix elements as follows:

$V_i = a_{in} \; a_{ni} / a_{ii}$ for each $a_{ii} > TOL$

Where a_{ij} indicates an element of Matrix A.

A tolerance is placed on a_{ii} since there is a possibility of degeneracy in the matrix when an independent variable is approximately a linear combination of other independent variables. A variable is not allowed to enter the regression unless the a_{ii} associated with it is greater than a preassigned value. The value of TOL is generally assigned a value which lies between .001 and .0001. In this program the value is assigned by the user in the control card. See 8.6.1.

Another reason for establishing a limit for a_{ii} is to prevent arithmetic overflow.

The method used to determine whether an x_i is in the regression at this step involves ascertaining the sign of V_i. This is due to the mathematical method of elimination which is used on the symmetric matrix.

If V_i is negative, X_i is in the regression at this stage. The smallest value of V_i is found for all X_i in the regression. Denote this as V_{min} and the corresponding subscript i as N_{min}. If V_i is positive, X_i is not in the regression yet. The largest value of V_i is found. Denote this as V_{max} and the corresponding subscript i as N_{max}.

The following criteria are used in selecting the x_i variable to add to or remove from the regression.

(1) If $(|V_{min}| \cdot \phi / a_{n,n}) < F2$, the corresponding variable x_i is insignificant at this time, therefore x_i will be removed from the regression before adding any other variable and $k_1 = N_{min}$.

(2) If $(V_{max} \cdot \phi / (a_{n,n} - V_{max})) > F1$, the corresponding variable x_i is significant and will be added to the regression. We then choose $k_1 = N_{max}$.

The pivot $a_{k1,k1}$ will be chosen according to the above rules. F1 and F2 are F values which should be preselected for a problem. These values are based on the approximate confidence level. The program will run into a loop if these values are not chosen correctly.

2.3.5.b. Matrix Operations

Given an element $a_{k,k}$ which is the pivot, selected according to the criteria in 2.3.5.a, where k corresponds to the subscript of one of the independent variables. The new matrix will be generated by the following algorithm. Denote the new matrix elements as a_{ij}.

$$\text{The new } a'_{ij} = \begin{cases} 1/a_{kk} & \text{for } i=j=k \quad (12) \\ a_{kj}/a_{kk} & \text{for } i=k,\ j\neq k \quad (13) \\ a_{ij} - a_{ik}\, a_{kj} / a_{kk} & \text{for } i\neq k,\ j\neq k \quad (14) \\ -a_{ik}/a_{kk} & \text{for } i\neq k,\ j=k \quad (15) \end{cases}$$

2.3.6 Calculation of the coefficients of regression equation.

After each variable entered or removed from the regression, the following are computed:

The standard error of the dependent variable:

$$s_y = \sigma_n \sqrt{a_{nn} / \phi} \tag{16}$$

The regression coefficients:

$$b_i = a_{in} \cdot \sigma_n / \sigma_i \tag{17}$$

The constant term:

$$b_o = \bar{y} - \sum b_i \quad x_i \tag{18}$$

The standard deviation of the regression coefficients:

$$s_{b_i} = s_y \cdot \sqrt{a_{ii} / \sigma_i} \tag{19}$$

where i corresponds to the subscript of the variable x_i in the regression at this stage and n the subscript of the dependent variable. a_{ij} is the element of the matrix at each stage.

2.3.7 Calculation of predicted values and deviations

The computation of the predicted values for dependent variables is based on the final regression equation. The deviation between the actual and predicted values of the dependent variable is

$$\delta_k = y_k - \hat{y}_k \qquad \text{for } k = 1, 2, \ldots m$$

3. Minimum System Components Required

3.1 Fortran Compiler D

3.2 One work tape.

4. Input

The input data is read from the card reader. The input data format will be given in detail in a later section.

5. Output

All output is printed on-line.

6. Subroutine

The square root routine must be supplied from the library of subroutines.

7. High Speed Memory Requirements

	Memory Space	Notation Used In Program	Remarks
A	N^2	VECTOR (I,J)	Matrix A space has multiple use as: 1) sums of squares and cross products 2) residual sums of squares and cross products 3) simple correlation coefficients 4) partial correlation coefficients 5) an inverse matrix
$\sum_i x_{ik}$	N	SUMX(I)	
σ_i	N	SIGMA(I)	
x_{ik}	N	X(I)	
$\bar{x}_i$	N	XBAR(I)	
b_i	N	BETA(I)	
s_{b_i}	N	SB(I)	
DEV	N	STDEV(I)	

7.1 Estimate of data size

$(N^2+7N)(d+2) \leq M - 16{,}500$

where M is the memory size

d is the digits of accuracy

As an example: Assume d = 10

Memory Size	Data Size
20K	$N \leq 14$
24K	$N \leq 22$
28K	$N \leq 28$
32K	$N \leq 33$

* Remark: This version is written for 16K machine with precision d = 10. In order to adapt this to other size machines, the arrays in the "common" statements of each chain must be changed according to the above estimated formula.

8. Preparation of Input Data

8.1 Size of numbers

The largest sum of squares must be less than 10^9. The size of numbers depends on the number of data points. The number that is punched on a data card should not be more than 10 characters, including sign and decimal point.

8.2 Number of variables

This program will accept a maximum of N independent variables calculated according to the estimated formula in section 7 and one dependent variable of a homogeneous set of data.

8.3 Number of problems

A maximum of 99998 problems can be run at one time.

8.4 Number of data points

The number of data points (observations) that can be handled is less than 10^6 and is limited by the restriction that the largest sum of squares must be less than 10^9. This program handled an equal number of homogeneous data points for each variable.

8.5 Decimal point

The decimal point of input data need not be entered if the number is an integer. The decimal point will be assumed to be at the extreme right. For example, columns 21 to 30 are used for the variable x_1. If a "25" is punched in columns 28 and 29, then x_1 will be 250. A number with a decimal point can be punched in anywhere within the specified columns, for that variable. For example, a 25. in column 21 to 23 is the same as if punched in 28 to 30.

8.6 Card format

8.6.1 Control card of each problem

Columns	Contents	Description or Remark
1-5		Not used
6-10	Problem ID.	An integer < 99999
11-15	Number of Observations	An integer
16-20	Number of variables	An integer (see 7 about its size limitation)
21-30	F1	Decimal
31-40	F2	Decimal
41-50	TOL	Decimal
51-80		Not used

8.6.2 Data card

Columns	Contents	Description or Remark
1-5	Blank	
6-10	Input Data ID	This no. should be exactly the same as the no. which appears in Column 6-10 in the Control Card.

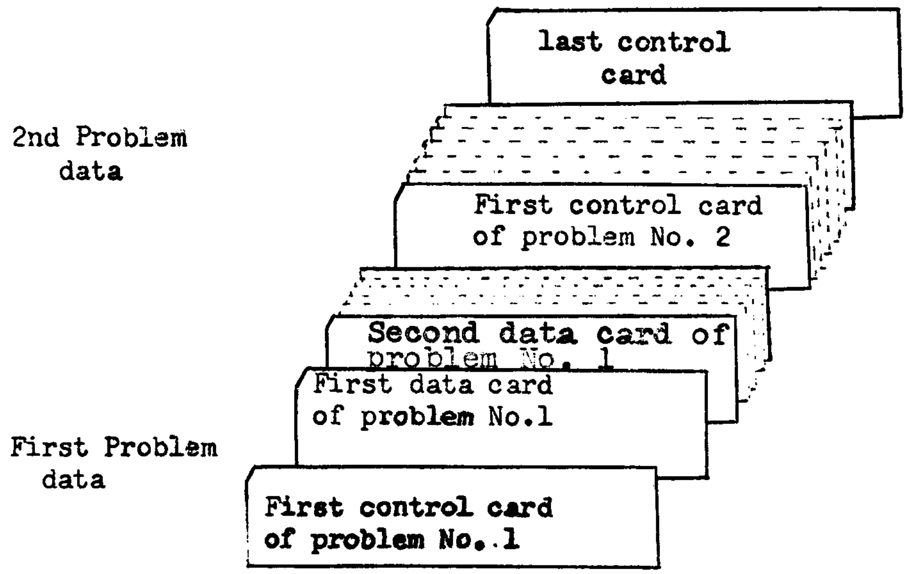
Columns	Contents	Description or Remark
11-15	Logical Sequence Number	An integer. 1 stands for first 5 variables data card, 2 for second five variables data card, etc; if number of variables is less than 5, then a 1 is sufficient for all data cards.
16-20	Observation No.	An integer. 1 stands for first observation, 2 for second observation, etc.
21-30	x_1	Decimal, value of first data
31-40	x_2	Decimal, value of second data
41-50	x_3	Decimal, value of third data
51-60	x_4	Decimal, value of fourth data
61-70	x_5	Decimal, value of fifth data
71-80		Not used

8.6.3 Last control card

This card is needed at the end of a set of problems.

Columns	Contents	Description or Remark
1-5		Not used
6-10	The integer 99999	
11-80		Not used

8.6.4 Card Sequence Layout



9. Sample Problem for Step-wise Regression

9.1 Problem 2

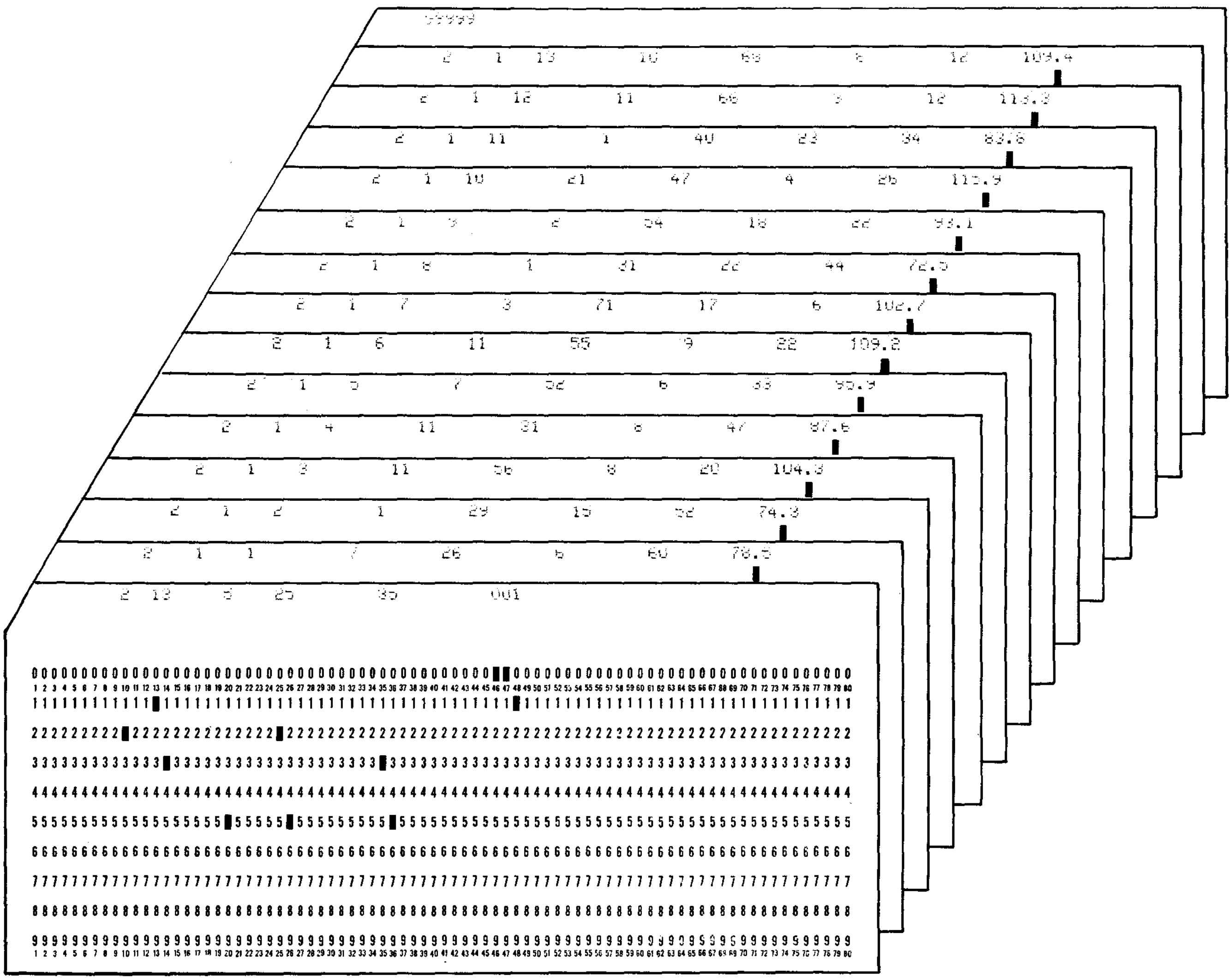
Find the best fit of

$$y = b_o + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4$$

with the following available data of 13 observations of values of 5 variables:

	x_1	x_2	x_3	x_4	Y
1	7.	26.	6.	60.	78.5
2	1.	29.	15.	52.	74.5
3	11.	56.	8.	20.	104.3
4	11.	31.	8.	47.	87.6
5	7.	52.	6.	33.	95.9
6	11.	55.	9.	22.	109.2
7	3.	71,	17.	6.	102.7
8	1.	31.	22.	44.	72.5
9	2.	54.	18.	22.	93.1
10	21.	47.	4.	26.	115.9
11	1.	40.	23.	34.	83.8
12	11.	66.	9.	12.	113.3
13	10.	68.	8.	12.	109.4

9.2 Input Data Cards



9.3 Output Data

STEP-WISE REGRESSION ANALYSIS PROGRAM(STRAP1)

PROBLEM NO. 2

NO. OF OBSERVATIONS = 13

NO. OF VARIABLES = 5

F LEVEL TO ENTER VARIABLE = 2.500

F LEVEL TO REMOVE VARIABLE = 2.500

SUM OF VARIABLES

X(1) = 97.000	X(2) = 626.000	X(3) = 153.000
X(4) = 390.000		
Y = 1240.499		

RAW SUMS OF SQUARES AND CROSS PRODUCTS

X(1) VS X(1) = 1139.000000	X(1) VS X(2) = 4922.000000	X(1) VS X(3) = 769.000000
X(1) VS X(4) = 2620.000000	X(2) VS X(2) = 33050.000000	X(2) VS X(3) = 7201.000000
X(2) VS X(4) = 15739.000000	X(3) VS X(3) = 2293.000000	X(3) VS X(4) = 4628.000000
X(4) VS X(4) = 15062.000000	X(	
X(1) VS Y = 10031.999940	X(2) VS Y = 62027.799970	X(3) VS Y = 13981.499940
X(4) VS Y = 34733.299970		
Y VS Y = 121088.089900		

AVERAGE VALUE OF VARIABLES

X(1) =	7.461	X(2) =	48.153	X(3) =	11.769
X(4) =	30.000				
Y =	95.423				

RESIDUAL SUMS OF SQUARES AND CROSS PRODUCTS

X(1) VS X(1) =	415.230768	X(1) VS X(2) =	251.076922	X(1) VS X(3) =	-372.615384
X(1) VS X(4) =	-289.999999	X(2) VS X(2) =	2905.692305	X(2) VS X(3) =	-166.538461
X(2) VS X(4) =	-3040.999993	X(3) VS X(3) =	492.307692	X(3) VS X(4) =	38.000000
X(4) VS X(4) =	3362.000000	X(			
X(1) VS Y =	775.961538	X(2) VS Y =	2292.953844	X(3) VS Y =	-618.230768
X(4) VS Y =	-2481.699996				
Y VS Y =	2715.763076				

STANDARD DEVIATIONS OF VARIABLES

X(1) =	5.882	X(2) =	15.560	X(3) =	6.405
X(4) =	16.738				
Y =	15.043				

CORRELATION COEFFICIENTS

X(1) VS X(1) =	1.000000	X(1) VS X(2) =	.228579	X(1) VS X(3) =	-.824133
X(1) VS X(4) =	-.245445	X(2) VS X(2) =	1.000000	X(2) VS X(3) =	-.139242
X(2) VS X(4) =	-.972954	X(3) VS X(3) =	1.000000	X(3) VS X(4) =	.029537
X(4) VS X(4) =	1.000000	X(			
X(1) VS Y =	.730717	X(2) VS Y =	.816252	X(3) VS Y =	-.534670
X(4) VS Y =	-.821305				
Y VS Y =	1.000000				

STANDARD ERROR OF Y = 15.043722

```
STEP NO.    1
  VARIABLE ENTERING         X=  4
  F LEVEL =      24.8711
  STANDARD ERROR OF Y =        8.9639

  CONSTANT TERM =    117.56793

  VARIABLE NO.          COEFFICIENT          STD ERR OF COEFF
     X=  4                   -.73816                   .15459

STEP NO.    2
  VARIABLE ENTERING         X=  1
  F LEVEL =     119.0462
  STANDARD ERROR OF Y =        2.7342

  CONSTANT TERM =    103.09738

  VARIABLE NO.          COEFFICIENT          STD ERR OF COEFF
     X=  1                   1.43995                   .13841
     X=  4                   -.61395                   .04864

STEP NO.    3
  VARIABLE ENTERING         X=  2
  F LEVEL =       5.5842
  STANDARD ERROR OF Y =        2.3087

  CONSTANT TERM =     71.64830

  VARIABLE NO.          COEFFICIENT          STD ERR OF COEFF
     X=  1                   1.45193                   .11699
     X=  2                    .41610                   .18561
     X=  4                   -.23654                   .17328
```

STEP NO. 4
VARIABLE REMOVED X= 4
F LEVEL = 1.8632
STANDARD ERROR OF Y = 2.4063

CONSTANT TERM = 52.57734

VARIABLE NO.	COEFFICIENT	STD ERR OF COEFF
X= 1	1.46830	.12130
X= 2	.66225	.04585

PREDICTED VS ACTUAL RESULTS

RUN NO.	ACTUAL	PREDICTED	DEVIATION
1	78.50000	80.07400	-1.57400
2	74.29999	73.25091	1.04908
3	104.29999	105.81473	-1.51473
4	87.59999	89.25847	-1.65847
5	95.89999	97.29251	-1.39251
6	109.19999	105.15248	4.04751
7	102.69999	104.00205	-1.30205
8	72.50000	74.57541	-2.07541
9	93.09999	91.27548	1.82451
10	115.89999	114.53754	1.36245
11	83.79999	80.53567	3.26432
12	113.29999	112.43724	.86275
13	109.39999	112.29343	-2.89343

END OF CALCULATION. PROBLEM NO. 2
END OLYMPIAD

9.4 Special Comments on Error Conditions

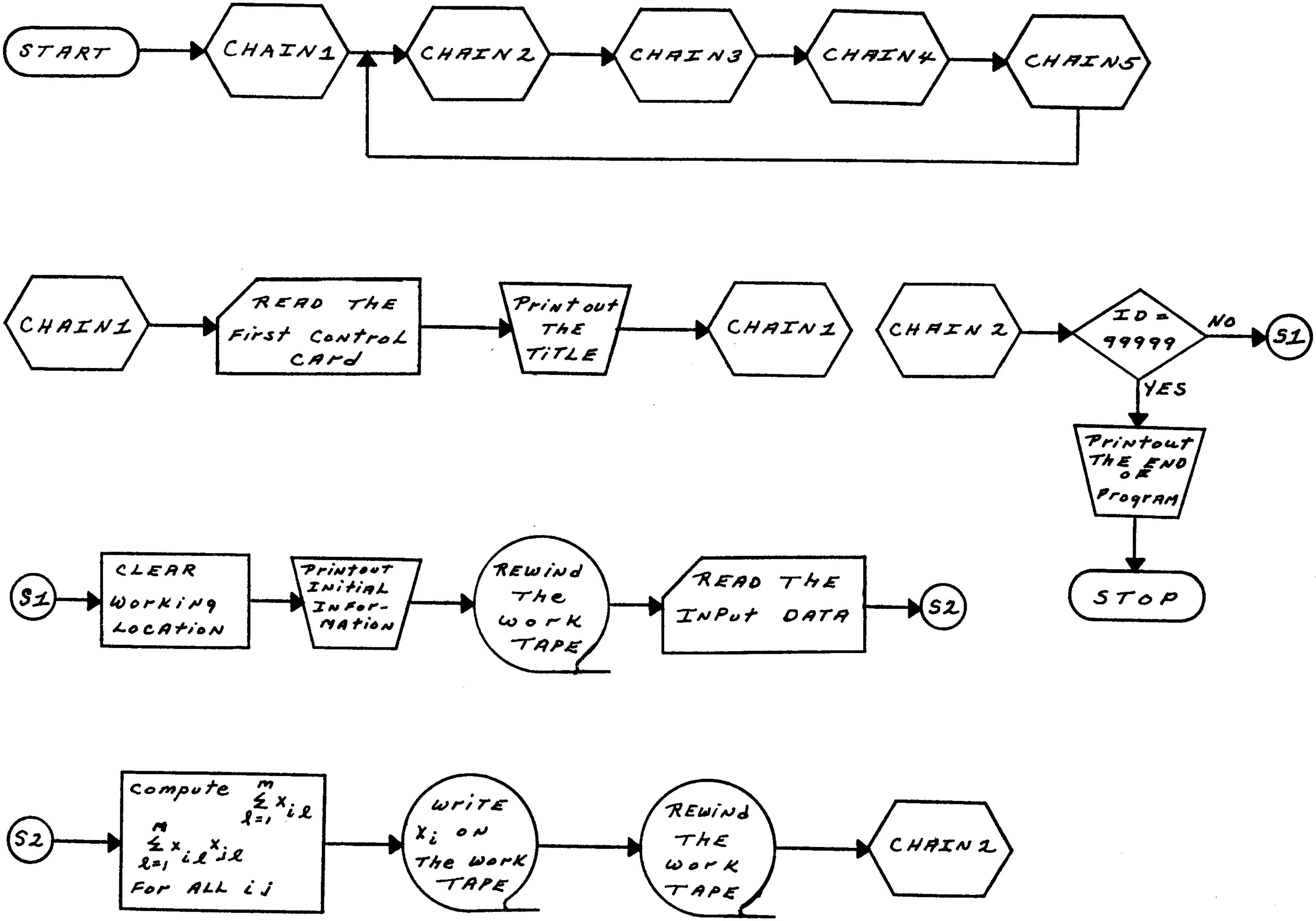
If the following comments appear on output:

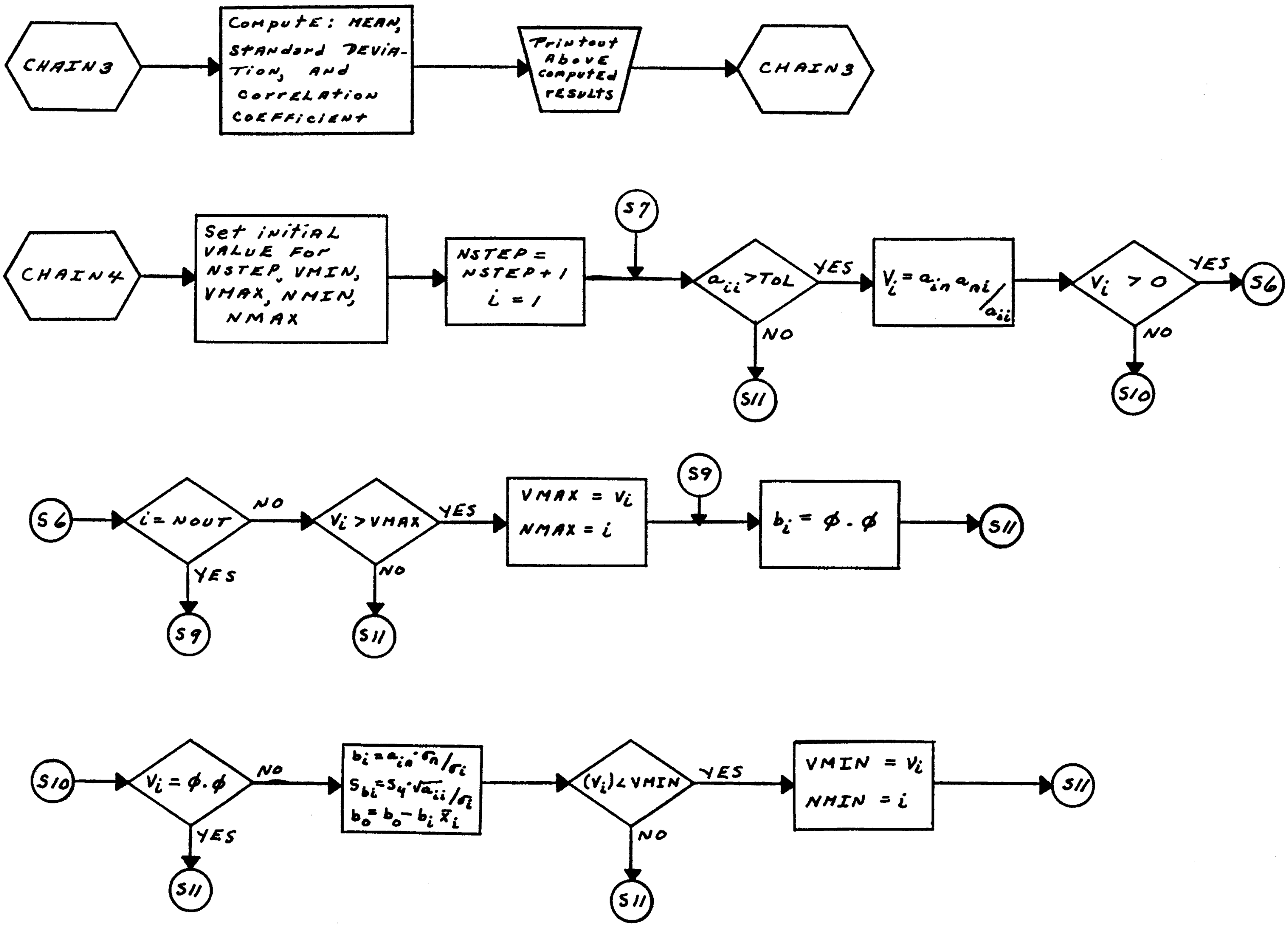
9.5.1 DATA ID DOES NOT AGREE WITH CONTROL CARD.
INID ≠ XXXXX XXXX NOC = XXXXX INNO = XXXXX

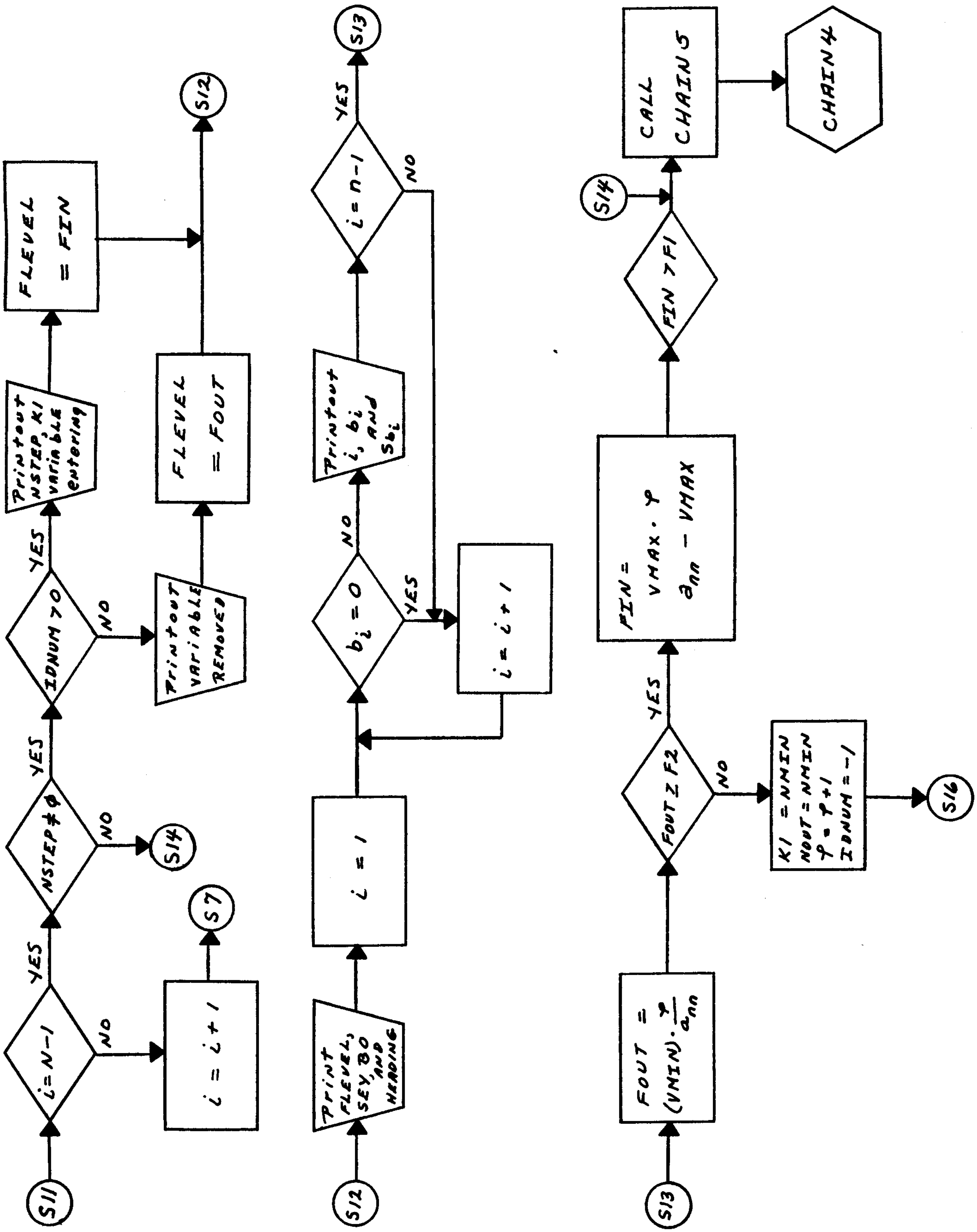
9.5.2 DATA CARD OUT OF LOGICAL SEQUENCE. CHECK.
INID = XXXXX XXXXX NOC = XXXXX INNO = XXXXX

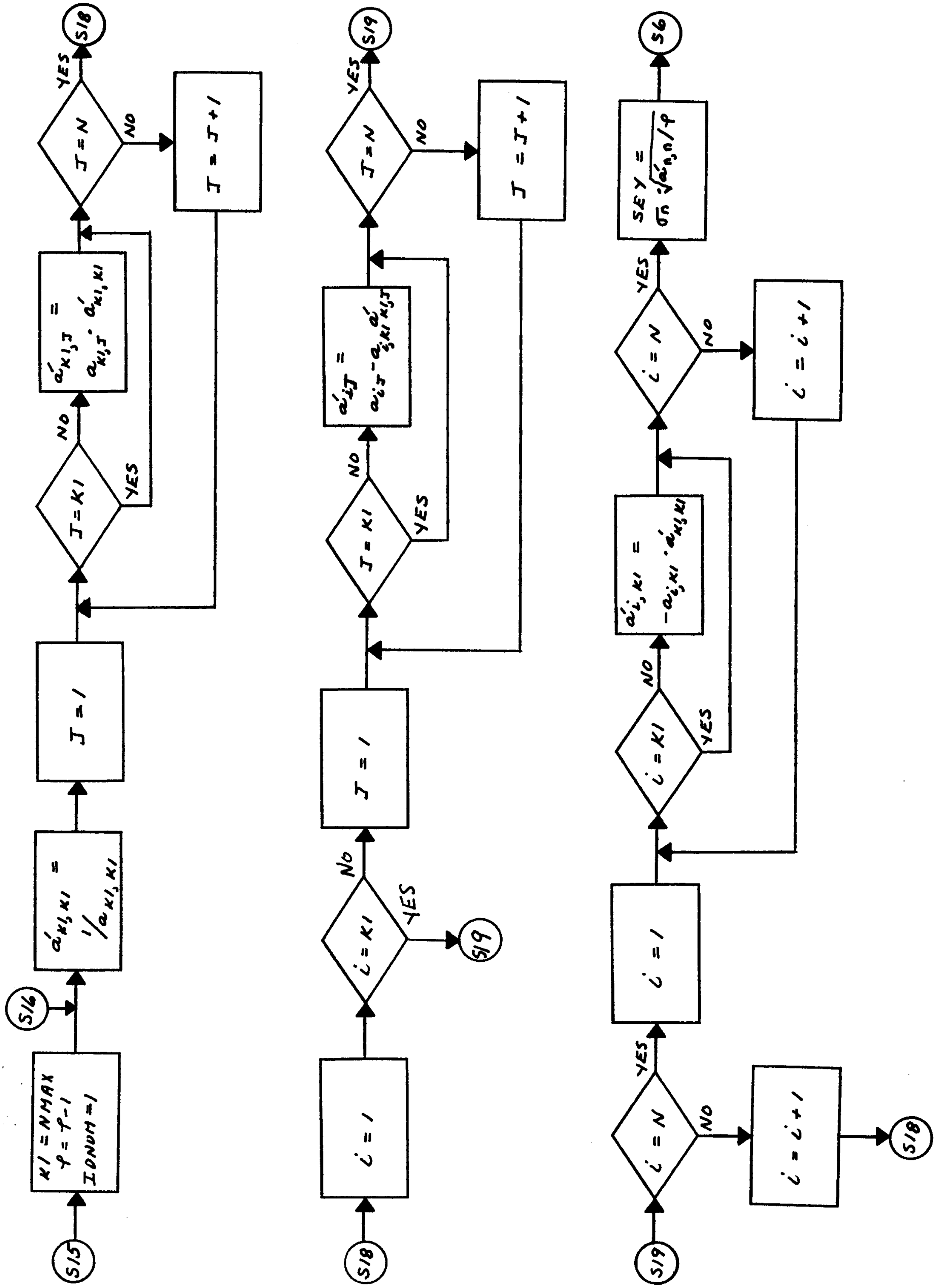
9.5.3 DATA CARD OUT OF NUMERICAL SEQUENCE. INID ≠
XXXX XXXXX NOC = XXXXX INNO = XXXXX

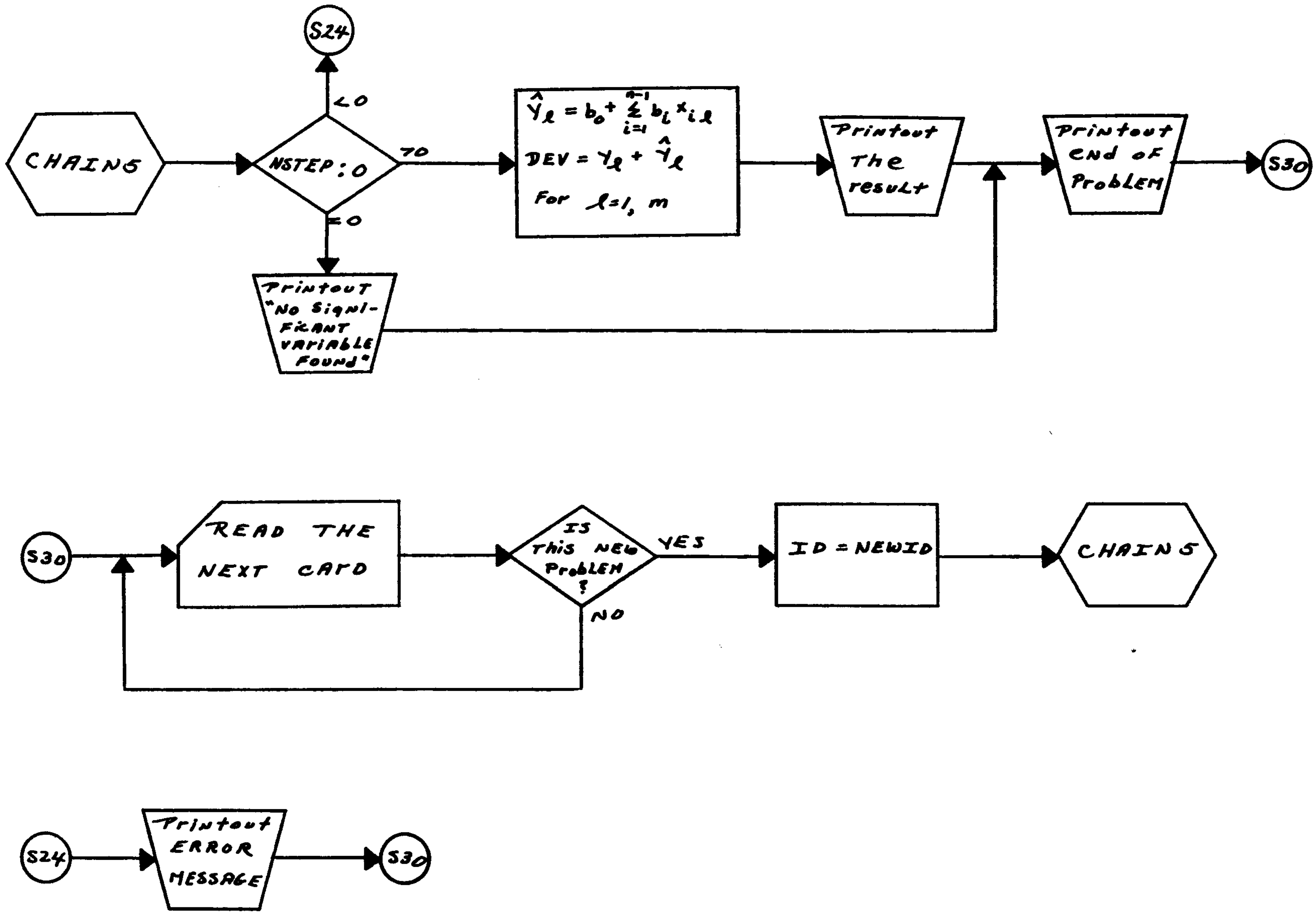
then these errors are caused by the mispunching of the identification number of the test data of a problem or misplacing of data cards. The program will continue, but the user should recheck the input data deck. The number INID, NOC and INNO will help the programmer to pinpoint the mispunched card or misplaced card. After correcting the input data, the program should be re-run.


START
CHAIN1
CHAIN2
CHAIN3
CHAIN4
CHAIN5
CHAIN1
READ THE First Control card
Printout The TITLE
CHAIN1
CHAIN 2
ID = 99999
NO
S1
YES
Printout The END of Program
STOP
S1
CLEAR Working Location
Printout Initial Information
Rewind the work TAPE
READ THE INPut DATA
S2
S2
compute $\sum_{l=1}^{m} x_{il}$ $\sum_{l=1}^{m} x_{il} x_{jl}$ For ALL i j
write x_i on The work TAPE
Rewind The work TAPE
CHAIN2


CHAIN3
Compute: MEAN, Standard Deviation, and Correlation Coefficient
Printout Above computed results
CHAIN3
CHAIN4
Set initial value for NSTEP, VMIN, VMAX, NMIN, NMAX
NSTEP = NSTEP+1
i = 1
S7
a_ii > TOL
YES
NO
S11
V_i = a_in a_ni / a_ii
V_i > 0
YES
S6
NO
S10
S6
i = NOUT
NO
YES
S9
V_i > VMAX
YES
NO
S11
VMAX = V_i
NMAX = i
S9
b_i = ϕ.ϕ
S11
S10
V_i = ϕ.ϕ
NO
YES
S11
b_i = a_in · σ_n / σ_i
S_bi = S_y · √a_ii / σ_i
b_0 = b_0 − b_i x̄_i
(V_i) < VMIN
YES
NO
S11
VMIN = V_i
NMIN = i
S11


S11
i=N-1
YES
NO
i = i + 1
S7
NSTEP ≠ 0
YES
NO
S14
IDNUM > 0
YES
NO
Printout NSTEP, KI variable entering
FLEVEL = FIN
Printout variable REMOVED
FLEVEL = FOUT
S12
S12
Print FLEVEL, SEY, BO AND HEADING
i = 1
b_i = 0
YES
NO
Printout i, b_i AND S_bi
i = n-1
YES
NO
S13
i = i + 1
S13
FOUT = (VMIN)·φ/a_nn
FOUT ≥ F2
YES
NO
KI = NMIN
NOUT = NMIN
φ = φ+1
IDNUM = -1
S16
FIN = VMAX·φ / a_nn - VMAX
FIN > F1
S14
CALL CHAIN5
CHAIN4


S15
KI = NMAX
φ = φ-1
IONUM = 1
S16
a'KI,KI = 1/aKI,KI
J = 1
J = KI
YES
NO
a'KI,J = aKI,J · a'KI,KI
J = N
J = J+1
S18
S18
i = 1
i = KI
S19
J = 1
J = KI
a'i,J = ai,J - ai,KI a'KI,J
J = N
J = J+1
S19
S19
i = N
i = i+1
S18
i = 1
i = KI
a'i,KI = -ai,KI · a'KI,KI
i = N
i = i+1
SEY = σn √(a'n,n/φ)
S6


CHAINS
NSTEP : 0
<0
>0
=0
S24
$\hat{Y}_l = b_0 + \sum_{i=1}^{n-1} b_i x_{il}$
$DEV = Y_l + \hat{Y}_l$
For $l=1, m$
Printout The result
Printout end of Problem
S30
Printout "No Significant variable Found"
S30
READ THE NEXT card
IS This New Problem ?
YES
NO
ID = NEWID
CHAINS
S24
Printout ERROR MESSAGE
S30

SECTION V

RANDOM NUMBER GENERATOR D PROGRAM

I. Introduction

1.1 Purpose

To generate a set of random numbers.

1.2 Method

1.2.1 Symbol

n = number of random numbers to be calculated.

a_j = j^{th} random number.

1.2.2 Description

The routine uses an additive process:

$$a_j = (a_{j-1} + a_{j-16}) \bmod 10^{10}$$

The a_j, $j=1, \ldots, 16$ are specified as input data. These 16 random numbers may be obtained from any table of random numbers (e.g. Quality Control and Industrial Statistics, by A. J. Duncan). After a set of random numbers has been generated by this routine, the last 16 numbers obtained may be used as input to a future run, to generate a new set of random numbers.

II. Usage as a Routine

2.1 Minimum System Components Required

2.1.1 16K FORTRAN

2.1.2 One on-line card reader

2.1.3 One on-line printer (if printed output is desired)

2.1.4 One on-line card punch (if punched output is desired)

2.1.5 One tape: Program Tape

2.2 Input

The input data is read from the card reader. The first card must be a control card, followed by four data cards.

2.2.1 Control Card Format

Columns	Contents	Description or Remark
1-30		Not used.
40	Blank or "1"	Blank for printed output, "1" for punched output.
41-45		Not used.
46-55	N	An integer, right justified, specifying the number of random numbers to be generated.
56-80		Not used.

2.2.2 Data Card Format

Columns	1-25	26-35	36-40	41-50	51-55	56-65	66-70	71-80
1st data card		a_1		a_2		a_3		a_4
2nd data card		a_5		a_6		a_7		a_8
3rd data card		a_9		a_{10}		a_{11}		a_{12}
4th data card		a_{13}		a_{14}		a_{15}		a_{16}

The a_j, $j=1, \ldots, 16$ are 10 decimal digit random numbers. The decimal point is assumed at the extreme left. The other columns are not used.

2.3 Output

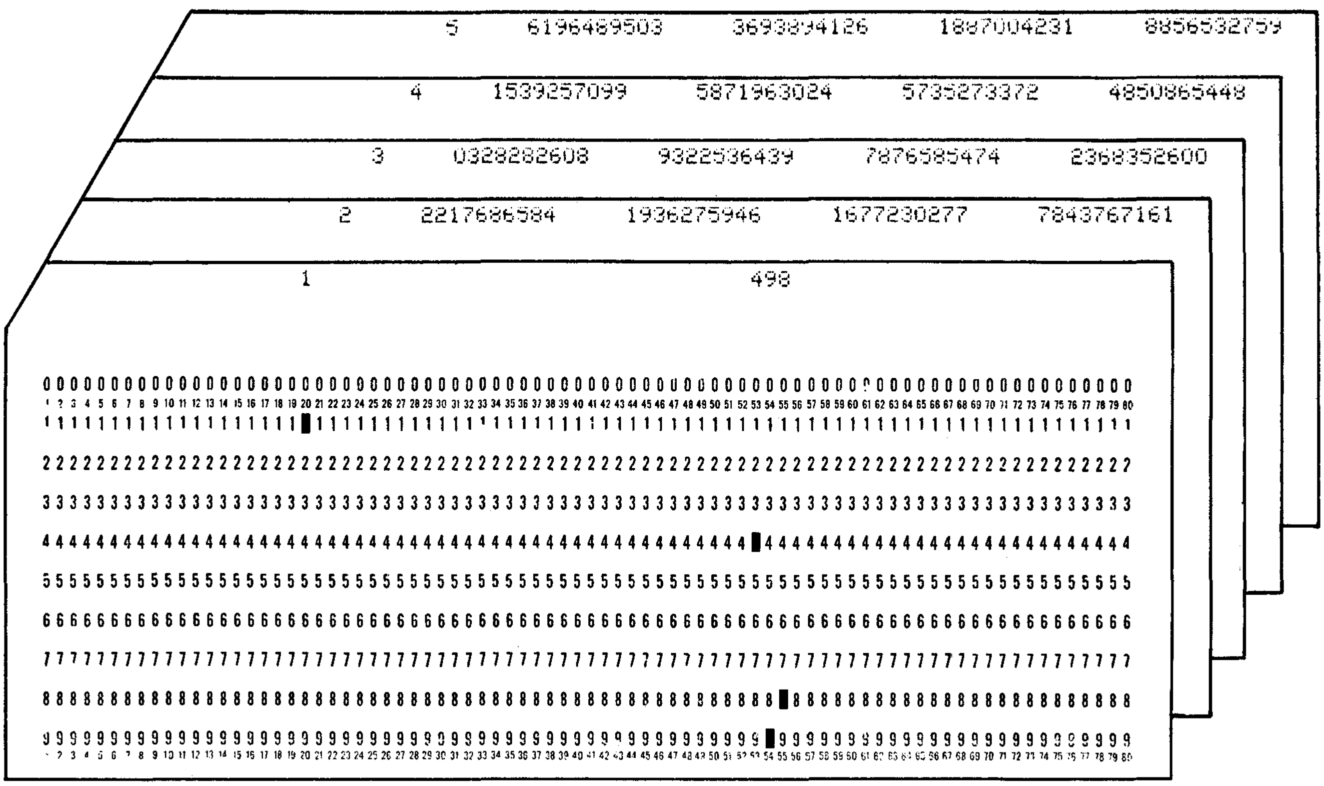
2.3.1 Printed

The generated random numbers will be printed on-line if specified on the control card. The output will include the first 16 random numbers, used as input.

2.3.2 Punched

If punched output is specified, then the new random numbers (excluding the 16 input random numbers) will be punched on-line onto cards. The format of these cards will be identical to the format of the input data cards (See 2.2.2).

2.4 Sample Input Data



The 16 random numbers above were taken from Quality Control and Industrial Statistics, by A. J. Duncan. The card numbers in column 2Ø are inserted merely for convenience.

III. Usage as a Subroutine

3.1 Minimum System Requirements

3.1.1 16K FORTRAN

3.1.2 One tape: Program Tape

3.2 Call Statement

CALL RANDOM (J, N, L, M), where J, N, L, M are a list of actual arguments.

J must be an integer constant, J=3

N is an integer constant, specifying the total number of random numbers to be generated.

L must be an array of integer variables of dimension 8, containing random numbers (a_1, a_2, ..., a_8).

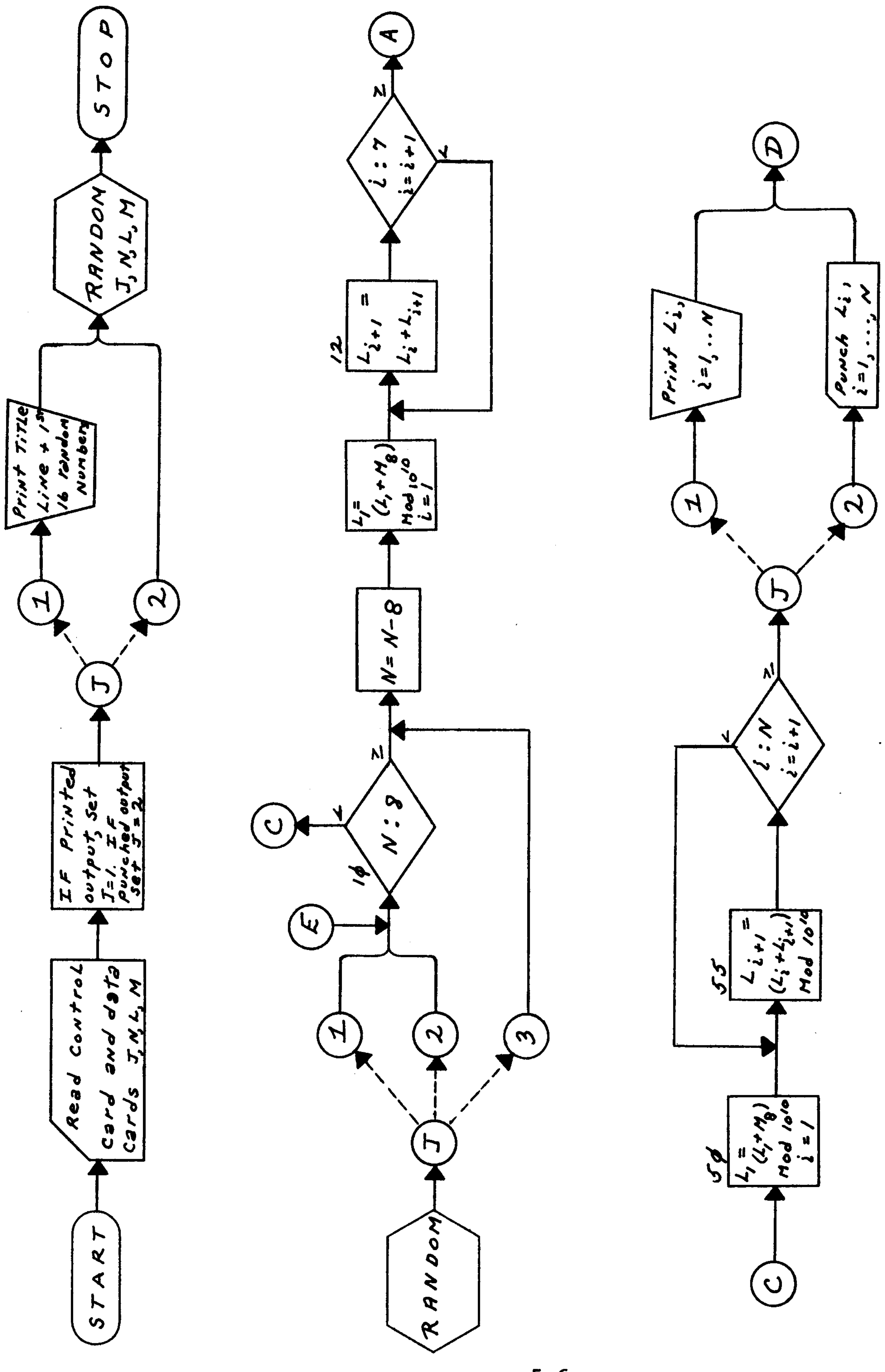
M must be an array of integer variables of dimension 8, containing 8 distinct random numbers (a_9, a_{10}, ..., a_{16}).

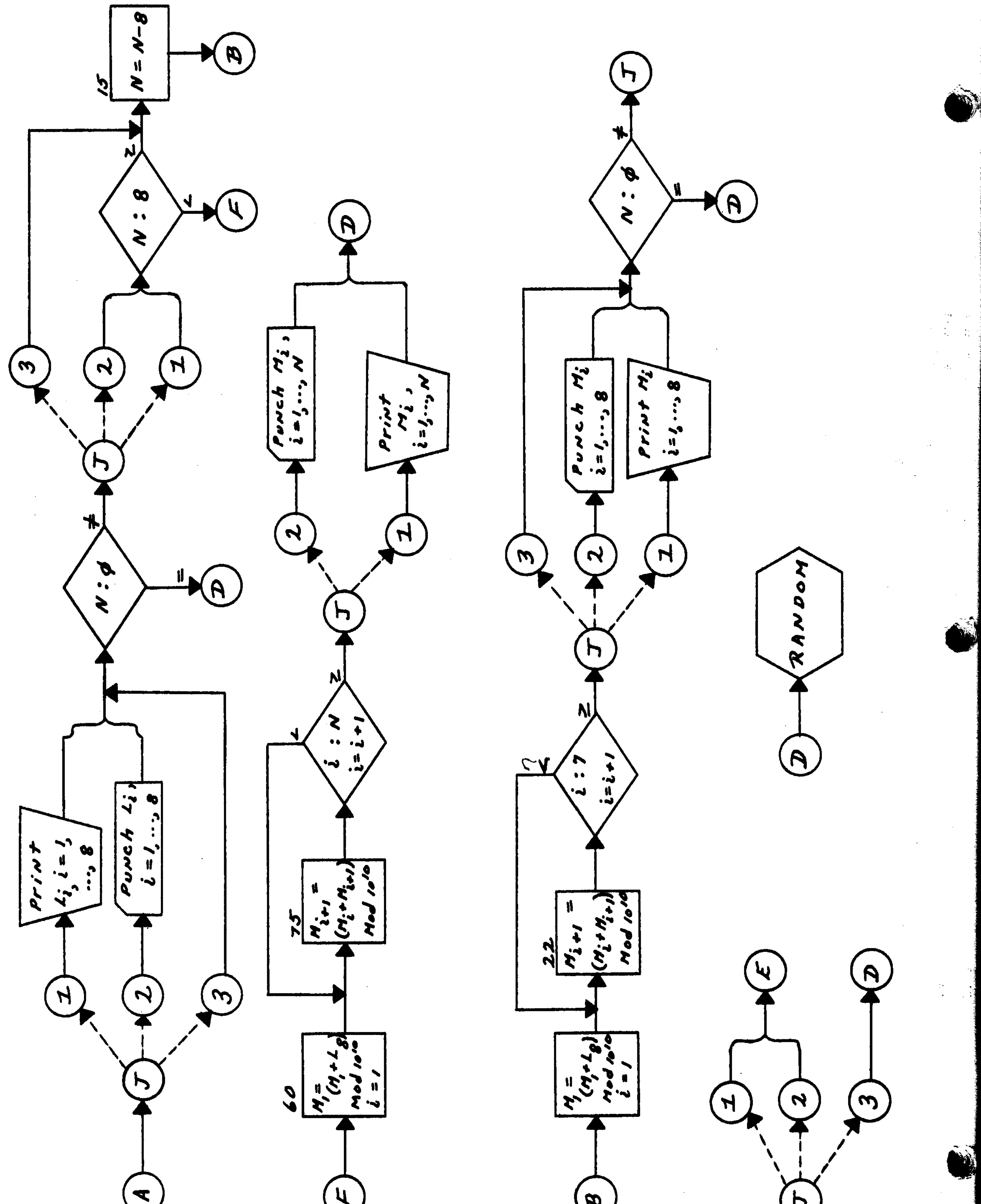
3.3 Output

With each call to the subroutine, RANDOM will deliver 16 random numbers. These will replace the previous 16, stored in L and M. N will be decremented by 16, after each call. Thus, when N $\leq$ Ø, the specified number of random numbers will have been generated.

IV. Disclaimer

This program has been checked but no guarantee is expressed or implied to the user concerning its functioning. Honeywell will cooperate in making functional those programs written by them. However, they assume no responsibility for programs distributed but not written by Honeywell.


START
Read Control card and data cards J,N,L,M
IF Printed output, set J=1. IF punched output set J=2
J
1
2
Print TITLE Line + 1st 16 random Numbers
RANDON J,N,L,M
STOP
RANDOM
J
1
2
3
E
1∅
N:8
C
N=N-8
L1= (L1+M8) Mod 10^10 i=1
12
Li+1 = Li+Li+1
i:7 i=i+1
A
C
5∅
L1 = (L1+M8) Mod 10^10 i=1
55
Li+1 = (Li+Li+1) Mod 10^10
i:N i=i+1
J
1
2
Print Li, i=1,..N
Punch Li, i=1,...,N
D


N = N-8
N : 8
N : ∅
Print L_i, i = 1, ..., 8
Punch L_i, i = 1, ..., 8
Punch M_i, i = 1, ..., N
Print M_i, i = 1, ..., N
M_i = $(M_i + L_8)$ Mod 10^{10} i = 1
M_{i+1} = $(M_i + M_{i+1})$ Mod 10^{10}
i : N i = i+1
i : 7 i = i+1
Punch M_i i = 1, ..., 8
Print M_i i = 1, ..., 8
RANDOM