

AD706936

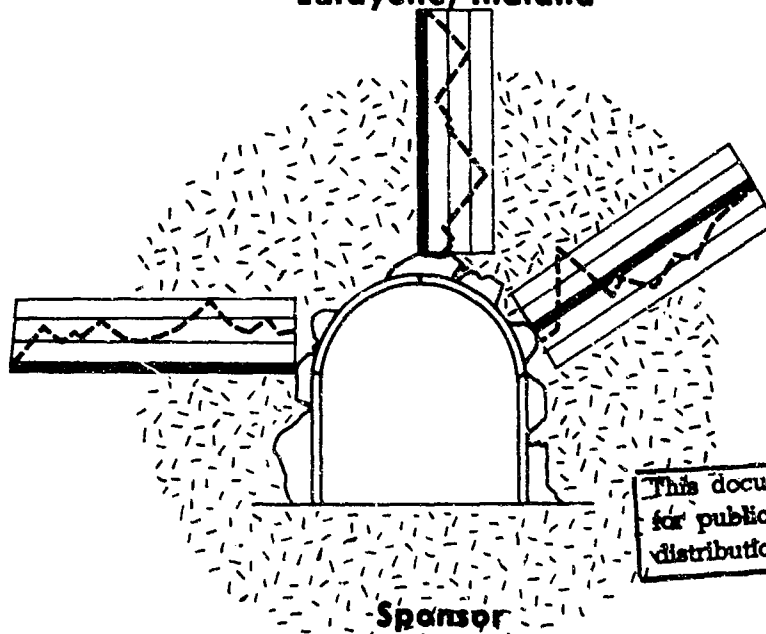
Strain Distribution Around Underground Openings

Technical Report No. 4

STATISTICAL METHODS TO COMPILE AND CORRELATE ROCK PROPERTIES —COMPUTER TECHNIQUES—

Patricia Nahas

Soil and Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana



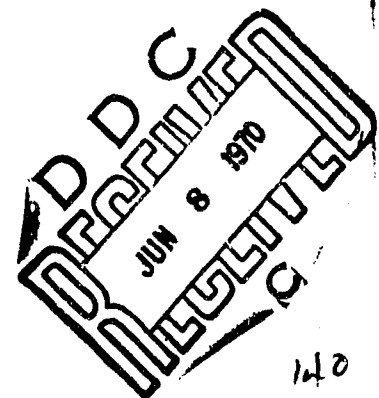
This document has been approved
for public release and sale; its
distribution is unlimited.

Sponsor
ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

May 1970

Prepared For
OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.



Strain Distribution Around Underground Openings

Technical Report No. 4

STATISTICAL METHODS TO COMPILE AND
CORRELATE ROCK PROPERTIES

- COMPUTER TECHNIQUES -

Patricia Nahas
Soil & Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana

Contract No. DACA 73-68-C-0002(P002)

Sponsor

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE

May 1970

Prepared For

OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.

Distribution of this document is unlimited

TABLE OF CONTENTS

	Page
Abstract	
1. Introduction	1
2. Data Storage	1
3. Descriptive Programs	
3.1 Listing of Tape	2
3.2 Count Programs	3
3.3 Ranges and Mean Values	5
4. Computational Programs	
4.1 Least Squares	7
4.2 Histograms and Scattergrams	8
5. Comments and Recommendations	10
Appendix	1A

Abstract

A data tape is necessary for the storage of a systematized collection of physico-mechanical properties of rocks. Specific programs permit the obtaining of descriptive information on the data - ranges, means, and counts. Statistical routines yield histograms, scattergrams, and least squares equations. One objective is to provide information that can form the basis for some degree of uniformity in such research. Experience with the programs yielded certain principles and changes that would improve efficiency. It is recommended that one choose an efficient means of data storage, maintain a back-up data source and precise records, run all descriptive programs first, process as many cases per run as possible, determine common scales where necessary for later interpretation, and label output meaningfully.

1. INTRODUCTION

Several requests have been made for information on how the statistical analyses used in Technical Report No. 2 have been handled, especially with reference to the computer techniques involved. The information in this supplement to Technical Report No. 2 is a brief summary of these techniques. It includes both specific examples of computer programs used to obtain the results presented in Technical Report No. 2 and commentary on variations and possible improvements. These programs are intended as a useful guideline for those doing similar research on their own data at their own facility.

2. DATA STORAGE

The preliminary work involved transferring the data to magnetic tape for efficient storage. First, the data were recorded on the coded sheets described in Appendix A of Technical Report No. 2. The data on these sheets was then keypunched on standard 80-column IBM cards. A scratch (temporary) tape was then created from these cards. The data on this tape was subjected to the standard UTILITY SORT routine for the IBM 7094. This procedure provided a second scratch tape which contained the information sorted according to rock type. The final operation involved the use of a COBOL program to create a magnetic tape containing the original data, sorted on rock type, in such a manner as was desired for print-out. Note that the SORT routines often distinguish between a blank and a zero. Hence, it is to one's advantage to be consistent in this respect when recording the data on the coded sheets. For example, in our case, a three-character field was allowed to accommodate the number identifying different rock types. When this number contained only two digits, say 22, it was sometimes entered as 022 and sometimes as _22. Either way

provide the necessary identification for the corresponding data. But since the SORT routine considers these as two different things, the group of data for rock type #22 on our tape is split, an undesirable situation as part appears in one section of the tape and part in another.

Several copies of this tape have since been made because constant use eventually results in bad spots that produce READ PARITY errors. (One good copy should always be reserved as a back-up tape.) Also, it was convenient to have several tapes available in order to speed up the computer turn-around time. For example, if each of two programs requested either tape #544 or tape #630, it was possible to have both programs running at once, each using the same information but from different tapes mounted on different tape drives. This could not have been done if both programs requested tape #544 only. One of the programs would have had to await the return of tape #544 before being run. (Our original tape was created for the IBM 7094 because that was the computer in use at that time in our Computer Sciences Center. Since then, the CDC 6500 has replaced the 7094 as the principal computer being used. We are now in the process of converting our tape to one internal to the CDC in hopes that computer usage of the tape will be facilitated. Then, because of better turn-around time, it may no longer be necessary to maintain several tapes.)

3. DESCRIPTIVE PROGRAMS

3.1 LISTING OF TAPE (pp. 13-15)

The first step after creating the data tape is to obtain a complete listing of the tape. Only in this way will you be assured that all the data have been successfully transferred to the tape. Such a listing will also provide a useful record for future reference. It should be noted again that one must always maintain a duplicate tape as a back-up tape. Accidents

or repeated usage can result in a tape that can no longer be successfully read by the computer.

After having obtained a listing of the tape, one may want to run programs that will provide for detailed information about the data on the tape. Several of the included programs are of this nature, while others perform certain computations on the data. Each program will now be considered individually. In the Conclusion, I will comment on other similar programs that could be run and on certain approaches or changes that in retrospect would have been more efficient and certainly less time consuming.

3.2 COUNT PROGRAMS

Program # 1 (pp. 16-23) is run to obtain information on the number of non-blank entries present for each of the variables on the tape. Each variable on each of the eight data cards per set^{1/} is read and compared with a constant, B, that is preset to blank. If the variable in question is non-blank, i.e., not equal to B, a counter is increased. (Note that the counters are initialized to zero as a precaution.) Otherwise, the counter maintains its value and the program proceeds to the next test, until all variables have been compared. The program then reads another record from the tape and proceeds as above. This process is repeated until all the data on the tape is exhausted. The output from the program consists of a listing of the variable names and the associated counts or N's. (Note that the variables are tested against a blank value and not zero because zero is a legitimate entry, i.e., a blank indicates no data whereas zero is an acceptable value.)

If such a count is needed only for certain selected variables, then one need read only those particular values from

^{1/} Note that the eight cards that are referred to as a "data set" in Technical Report No. 2 will be referred to as a "record" with respect to the data tape.

the tape. (See program 1a, pp.24-26). Depending upon the manner in which your data is set up on your tape, it may or may not be necessary to provide for reading dummy information in order to skip to the next record. For example, suppose we wish to concern ourselves only with a variable - say A20 - on the A card. Then in order for our DO LOOP to execute properly, one of two alternatives is necessary. We could have the following READ and FORMAT statements:

```
      READ (1,1) A20
1     FORMAT (10X,A4////////)
```

where the seven slashes would cause the computer to skip the next seven cards and begin reading again with the first card of the next record, or we could read dummy variables as follows:

```
      READ (1,1) A20
      READ (1,3) B
      READ (1,3) C
      READ (1,3) D
      READ (1,3) E
      READ (1,3) F
      READ (1,3) G
      READ (1,3) H
1     FORMAT (10X,A4)
3     FORMAT (50X,A5)
```

where the tape unit we are reading from is Tape 1. The latter alternative is selected as safer since the interpretation of the slashes is subject to variation depending on the particular computer.

Program #2 (pp. 27-30) is a slight variation of Program #1. For certain chosen variables, it is necessary to know how many cases there are of non-blank entries for both variables for the same data set. For this program, the required variables for each record are read and then each variable for the particular pair is compared against the blank. If both are non-blank, the

counter is incremented. Otherwise, the program goes on to the next test. After all pairs for a data set are processed, the variables for the next data set are read and compared. This procedure is repeated until all data on the tape is used. The output is a listing of the values of N for the various pairs. Note that here, as previously, the reading and comparisons are performed within the same DO LOOP. Because of the large quantity of data, it is not feasible or even possible to read the data into arrays and then perform the comparisons separately.

One further variation of these programs may be useful. It may be necessary to obtain such information for the variables within a particular rock type rather than over all rock types. Such can be accomplished by reading in the rock type variable along with the data in question and testing its value against the value identifying the rock type of interest. The sequence of instructions would be as follows:

```

DO 100 K = 1,2170
  READ (1,1) IRT,A20,A1
  READ (1,3) BB
  .
  .
  .
  READ (1,3) HH
1  FORMAT (15X, I3, 10X, A4, A3)
3  FORMAT (50X,A5)
  IF (IRT .NE. 223) GO TO 100
  IF {(A20 .NE. B) .AND. (A1 .NE. B)} N = N + 1
100 CONTINUE

```

where IRT is the variable for rock type and 223 is the value of the rock type of interest.

3.3 RANGES AND MEAN VALUES

Program #3 (pp. 31-35) and Program #4 (pp. 36-43) provide further information about the data on the tape. Program

#3 is designed to print out all values within specified ranges for a particular selection of variables and the associated rock types. In addition, the program computes for each variable the maximum and minimum values, the number of values, and the mean over these values.

At this point, it is important to understand that the data on our tape was entered in such a manner that it could be read off only in integer or A field format. The variables could not be read off as real variables. For all homogeneous computations (i.e., computations involving only one variable) it was not necessary to perform any scaling in order to obtain meaningful results. Any scaling could be applied directly to the results. So if the entry 432 for variable B8 really represented 4.32 and we got a mean of 397 for B8, we knew immediately that the actual mean was 3.97. However, if the computation involved more than one variable, it was mandatory that the variables be scaled before the computation began. This fact will be discussed later in conjunction with several of the computational programs. Also discussed in conjunction with the computational programs is the reason for specifying ranges for certain of the variables. (pp. 7, 9 and 10)

Program #4 differs from Program #3 in that the print-out is limited each time to a particular rock type. This permits the comparison, for example, of the ranges of values for a particular variable over the whole tape with the range within a specific rock type. The actual differences in the program are modifications to test for rock type and to reinitialize counters. It is, of course, possible that some variables for a particular rock type would be blank, and such a situation has to be taken into account. It is also possible that the last non-blank record read from the tape would be of a different rock type than the preceding. In that case, it is necessary to provide a way for printing out the last information. By inserting the test on END OF FILE, it is possible to reinitialize all

values, go back and perform the tests, and then print out the desired information. The test on END OF FILE guarantees that this last information will not be lost either through exceeding the DO LOOP limits or through trying to read past the END-OF-FILE on the tape. (Note that the DO LOOP parameter goes from 1 to 2170 previously and from 1 to 2171 in this case to allow for this last possibility.)

The following are also to be noted. The manner in which the DO LOOP parameter is set up requires that the first time through the loop you test IRT(1) against IRT(0). (The IRT array represents the values identifying the different rock types.) To make this test possible, we equivalence the first element of the IRT array, i.e., IRT(1) with the second element of a two-element array called DUMMY. Hence $IRT(1) = DUMMY(2)$. Then, IRT(0) is equivalent to DUMMY(1) which is set equal to the value of the first rock type on the tape. In this way, the first time the program reads the rock type, it reads $IRT(1) = 3$ and when it goes to compare IRT(1) with IRT(0), the computer has a value for IRT(0) and it is 3. From there on, all proceeds as usual.

4. COMPUTATIONAL PROGRAMS

4.1 LEAST SQUARES

Program #5 (pp. 24-26) also reads selected variables from the tape and compares certain pairs with blank. When both variables in the pair for the same data set are non-blank, the values are stored in arrays. However, the values in this case must be scaled so that the Least Squares Equation will make sense. For example, if 429 for B15 represents the real value of 4.29 and 364 for A1 is actually 36.4, this difference in decimals must be maintained for the resulting equation to be meaningful. In addition, the associated rock types are stored in another array, and this information (i.e., the pairs of points and the rock types for which they occur) is part of the print-out. The values in these arrays are the input data for the Least Squares

routine. The routine computes the least squares coefficients A and B of the equation $Y = AX + B$ from this data. Note that the comment cards in the Least Squares program deck adequately explain its source and setup. As many equations as desired can be obtained in a particular run, subject of course to the storage limits and time considerations of a particular computer. It should be noted that the N for a particular pair of variables must be known in advance. This information is needed not only for the dimensioning of the arrays but must also be fed as a parameter to the Least Squares routine. Hence, we see the value of the Count Programs.

4.2 HISTOGRAMS AND SCATTERGRAMS

These Count programs are also necessary for the histogram and scattergram routines which follow. These require the number of data points as one of their input parameters. The programs are set up in such a way that the first part of both reads the selected variables from the tape and makes the comparisons with the blank. For those cases where both variables have non-blank entries for the same data set, the program stores the values in arrays. After the whole tape has been read, these arrays are put on the disk and serve then as input for the parts of the program that produce the scattergrams and histograms. These parts of the routines are taken from: "BIOMEDICAL COMPUTER PROGRAMS, HEALTH SCIENCES COMPUTING FACILITY, DEPARTMENT OF PREVENTIVE MEDICINE AND PUBLIC HEALTH SCHOOL OF MEDICINE, UNIVERSITY OF CALIFORNIA, LOS ANGELES, JANUARY 1, 1964", W. J. Dixon, Editor. Included with the print-out of these programs as they have been adapted to run on our computer are the user's write ups. These provide information on how to set up the data cards, and the various options available. For example, the program that gives the scattergram output could be run just to obtain correlation coefficients.

You will also note that these BMD routines are on a library file at our facility and are loaded as part of the user's program via the control card

LIBCOPY(STATBIN,LGO,BMD2D)

Hence, the program deck for these two routines is set up as follows:

Control Cards

7₈₉

Program Deck

7₈₉

Data Cards for BMD Routine

6₇₈₉

This set up is not immediately obvious from the print-out sheets, especially since the BMD routines are not printed out each time.

Another important point is that these routines accept only real variables as input. That is, the data format card is valid only in Fw.d format. It does not matter that the variables in the first part of the program are written on the disk in A format. The BMD routine will automatically convert them as it reads them. What is necessary though is that the Fw.d format accurately represents the true value of the variables with respect to each other. For example, suppose two variables are written on the tape via the format (A4,A3). Then if the values 1492 and 765 really represent 14.92 and 76.5, they must be read by the format (F4.2,F3.1) and not by (F4.0,F3.0). The former will permit the BMD routine to perform meaningful computations, whereas the latter will not.

1492 - 765 = 727 would be an erroneous and meaningless computation because the real computation is

14.92 - 76.5 = -61.58

The histogram program is also run to obtain a master histogram for each variable on the tape. (For further explanation refer to Section 4.4 in Technical Report No. 2) The program is

a simple variation of the one included. As before, one needs to know the number of non-blank entries for the variable in question. Then it is a simple matter of reading the tape and transferring the values to the disk for input to the histogram. In several cases in our computations, upper and lower limits were set for the variable and each value was tested not only against the blank but also against the desired range. This testing was necessary to eliminate bad entries. These latter had the unfortunate effect of altering the scaling on the histogram in such a way as to make the graph meaningless. For example, suppose the range of the selected variable was 0 to 20, and that a value of 17.4 had been erroneously entered as 174.0. Then instead of being evenly spread, the histogram would be squeezed together on the left, with just one value to the far right, a situation which is of little help for interpretation and comparison. In addition, the histogram program can be run to obtain histograms for variables within a particular rock type. All that is required is the insertion of a test on rock type similar to the one previously described.

5. COMMENTS AND RECOMMENDATIONS

Experience with the preceding programs has led to suggestions which we plan to use in the continuing research on these rock problems. We expect these suggestions to improve the efficiency of the operation and to prevent the reader from repeating some of our mistakes.

RECOMMENDATION NO. 1

All programs giving descriptive information about the data on the tape should be run first. In this way, erroneous or extreme entries will be discovered immediately and the tape can be corrected or the subsequent programs can be written with tests that will screen out such values. This will eliminate the need to rerun programs processed before these errors are discovered.

RECOMMENDATION NO. 2

Investigate the possibility of creating a permanent file containing one's data. In this way, one eliminates the need to mount a tape each time a program is run, and turn-around time should be correspondingly faster. We plan to try this approach with the next group of programs we will be running.

RECOMMENDATION NO. 3

Process more than one histogram or scattergram per run by performing comparisons on several pairs of variables at once and writing the arrays on the disk in order, one after the other. Then include in your deck data cards for each of the successive histograms or scattergrams. (Note the repeat specifications in the write-ups accompanying the BMD routines.) This technique has been tested successfully in connection with another problem.

RECOMMENDATION NO. 4

Keep a back-up tape.

RECOMMENDATION NO. 5

Keep careful and precise records. These may take in part the form of comment cards in programs. But they may also be written notations, etc. Only in this way is one able to go back after a period of time and recall accurately what he did^{2/}. And in addition, only in this way can the information be successfully handed over to another person.

RECOMMENDATION NO. 6

Determine a common scale for each of the base variables in the Comparison Histograms. Use the scale obtained in the Master Histogram for the variable whenever that variable

^{2/} This is especially important in our situation, because Purdue's computer center operation is in part run for student learning in addition to user service. Thus, there is constant change in the system and in available routines. This means that slight changes may be necessary any time a program is rerun after a time lag.

appears in a Comparison Histogram. Note that the SELECT card we use specifies an impossibly small interval so that the BMD program will supply optimum scaling for the values in question. This approach was necessary since we had no advance idea as to the ranges of the variables once the comparisons for the pair had been made. In retrospect, we realize that we could have expanded the program that made the original counts and had it give us maxima and minima also.

RECOMMENDATION NO. 7

Use two label cards for the Comparison Histograms. The first will identify the variable in the particular histogram. The other will indicate the variable of comparison, i.e., the variable against whose values the histogram variable was compared.

COMMENTS

A few additional comments are directed toward the BMD routines. It would be convenient if we could feed these a variable N for the number of data points. If this were possible, we could eliminate the Count programs and simplify the first part of the present scattergram and histogram routines. This task is one to be tackled in the future. It may, however, prove more advantageous to replace the BMD routines with ones of our own, possibly utilizing the Calcomp Plotter for the graphing.

It is hoped that the above will be helpful to all who are involved in similar research. Perhaps some degree of consistency and uniformity can be achieved which will make the research understandable to all involved and which will permit comparisons. It is also hoped that useless and time-consuming duplications of effort will thus be avoided. Much can still be done in the area of programming techniques. What is included here is by no means exhaustive or definitive. But it is set forth with the hope that it is a start toward communication and a help toward further research.

PARTIAL LISTING OF DATA TAPE

O 1314
 M 1314
 A 1424 B 17 1003 0 0
 B 1424
 C 1424
 D 14241013 1 97 1 100+11
 E 1424
 F 1424
 G 1424
 H 1424
 A 849 U 1 511004635 0109 266
 B 849
 C 849
 D 8491012 3242 1 1196+26
 E 849
 F 849
 G 849
 H 849
 A 1303 K 6 91005 0 02 1 261 250 41 17 71
 B 1303
 C 1303
 D 13031013 31 1 208
 E 1303
 F 1303
 G 1303
 H 1303
 A 152 A 1 102007 0 0208 1 1 297 04 75 29 79
 B 1521562 9032112 176 3 124 491
 C 152
 D 1521012 9032521 389
 E 152
 F 152
 G 152
 H 1521102 903252
 A 130 A 1 91007 0 0108054 13 285 09 69 15 43
 B 1301562 9032111 152 4 892 389
 C 130
 D 1301012 9032521 334
 E 130
 F 130
 G 130
 H 1301102 903252
 A 92 B 9 121007 0 0101052 51 254 137 57
 B 921552 33111 +20 342 151
 C 92
 D 921022 33421 24 03 46 06 59 18 1181 02 200+01
 E 92
 F 921 04 235+05 08 276+10 106
 G 92
 H 921112 33421 012
 A 91 B 9 121007 0 0101052 51 254 137 57
 B 911552 33111 +25 340 263
 C 91
 D 911022 33421 99 03 117 09 157 18 1801 02 439+11
 E 91
 F 911 04 531+13 08 582+15 103
 G 91
 H 911112 33421 032
 A 57 3 91007 0 0102 1 61 262 36
 B 571552 33111 +11 378 172
 C 57
 D 571022 33421 97 06 154 13 220 20 2871 03 437+08 333

14

C	455								
D	4551012	9032522	282		2	1005+14			
E	455								
F	4551141	994+20		282 977+19					
G	455								
H	455								
A	259 A	2 92007614	0205055		270	50	49	2	36
B	2591562	9032112		127 20+09 59	268				
C	259								
D	2591012	9032521	173		2	67 +16			
E	259								
F	2591 87 68	+20		173 62 +26					
G	259								
H	2591102	903252						2	21
A	260 A	2 10200760A	0205055		280	04	45	25	8A
B	260								
C	260								
D	2601012	9032521	496		2	87 +15			
E	260								
F	2601248	83 +16		496 80					
G	260								
H	2601102	903252						2	41
A	258 A	2 92007674	0205055		204	180		3	14
B	2581562	9032112		58 -04 091 04A					
C	258								
D	2581012	9032522	25		2	075+26			
E	258								
F	2581 13	028+34		25 028+42					
G	258								
H	2581102	903252						2	03
A	993 P	1 172007	0 02 05						
B	993								
C	993								
D	993								
E	993								
F	993								
G	993								
H	9931112		521 070						
A	817 D	29762007	0 02		2	75			
B	817								
C	817								
D	817								
E	817								
F	817								
G	817								
H	817								
A	818 D	29762007	0 02		3	264	81		
B	818								
C	818								
D	818								
E	818								
F	818								
G	818								
H	818								
A	816 D	29682007634	02		3	294	34	71	14
B	816								
C	816								
D	8161012		32421 282		1	09	+141005		
E	816								
F	8161 26		+17 977		43	+20 994	77	+14 977	
G	816								
H	816								

Program No. 1
COUNT PROGRAM

Number of non-blank entries for each variable on a tape

```

000003      PROGRAM VC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
000003      DIMENSION N(78)
000003      DATA R/JH /
000005      DO 10 K=1,78
000010      N(K) =
000012      DO 100 K=1,2170
000017      READ(1,1) A1,A2,A3,A4,A5,A6,A7,A8,A9
000032      FORMAT(2X,A1,3X,A4,2X,A4,2X,A4,2X,A3,2X,A4,2X,A2,2X,A3)
000037      READ(1,2) B1,B2,B3,B4,B5,B6,B7,B8,B9
000065      FORMAT(2X,A3,A3,A2,A3,A4,A4,14X,A3,2X,A3,2X,A3)
000066      READ(1,3) C1,C2,C3,C4,C5,C6,C7,C8,C9
000113      READ(1,4) D1,D2,D3,D4,D5,D6,D7,D8
000137      FORMAT(21X,A4,5X,A4,3X,A4,3X,A4,4X,A4,3X,A4,2X,A2)
000137      READ(1,5) E1,E2,E3,E4,E5,E6,E7,E8
000163      FORMAT(10X,A4,A3,A4,2X,A2,7X,A4,A3,A4,2X,A2,7X,A4,A3,2X,A2)
000217      READ(1,6) G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12
000253      READ(1,7) H1,H2,H3,H4,H5,H6,H7,H8,H9,H10,H11
000305      FORMAT(21X,A4,6X,A4,3X,A4,3X,A4,3X,A4,3X,A4,3X,A3)
000311      IF(A1 .E. R) N(1) = N(1) + 1
000315      IF(A2 .E. R) N(2) = N(2) + 1
000321      IF(A3 .E. R) N(3) = N(3) + 1
000325      IF(A4 .E. R) N(4) = N(4) + 1
000331      IF(A5 .E. R) N(5) = N(5) + 1
000333      IF(A6 .E. R) N(6) = N(6) + 1
000335      IF(A7 .E. R) N(7) = N(7) + 1
000341      IF(A8 .E. R) N(8) = N(8) + 1
000345      IF(A9 .E. R) N(9) = N(9) + 1
000351      IF(A1 .E. R) N(10) = N(10) + 1
000355      IF(A2 .E. R) N(11) = N(11) + 1
000361      IF(A3 .E. R) N(12) = N(12) + 1
000365      IF(A4 .E. R) N(13) = N(13) + 1
000371      IF(A5 .E. R) N(14) = N(14) + 1
000375      IF(A6 .E. R) N(15) = N(15) + 1
000401      IF(A7 .E. R) N(16) = N(16) + 1
000405      IF(A8 .E. R) N(17) = N(17) + 1
000411      IF(A9 .E. R) N(18) = N(18) + 1
000415      IF(C1 .E. R) N(19) = N(19) + 1
000421      IF(C2 .E. R) N(20) = N(20) + 1
000425      IF(C3 .E. R) N(21) = N(21) + 1
000431      IF(C4 .E. R) N(22) = N(22) + 1
000435      IF(C5 .E. R) N(23) = N(23) + 1
000441      IF(C6 .E. R) N(24) = N(24) + 1
000445      IF(C7 .E. R) N(25) = N(25) + 1
000451      IF(C8 .E. R) N(26) = N(26) + 1
000455      IF(C9 .E. R) N(27) = N(27) + 1
000461      IF(D1 .E. R) N(28) = N(28) + 1
000465      IF(D2 .E. R) N(29) = N(29) + 1
000471      IF(D3 .E. R) N(30) = N(30) + 1
000475      IF(D4 .E. R) N(31) = N(31) + 1

```

```

000501      IF(D5 .E. R) N(32) = N(32) + 1
000505      IF(D6 .E. R) N(33) = N(33) + 1
000511      IF(D7 .E. R) N(34) = N(34) + 1
000515      IF(D8 .E. R) N(35) = N(35) + 1
000521      IF(E1 .E. R) N(36) = N(36) + 1
000525      IF(E2 .E. R) N(37) = N(37) + 1
000531      IF(E3 .E. R) N(38) = N(38) + 1
000535      IF(E4 .E. R) N(39) = N(39) + 1
000541      IF(E5 .E. R) N(40) = N(40) + 1
000545      IF(E6 .E. R) N(41) = N(41) + 1
000551      IF(E7 .E. R) N(42) = N(42) + 1
000555      IF(E8 .E. R) N(43) = N(43) + 1
000561      IF(F1 .E. R) N(44) = N(44) + 1
000565      IF(F2 .E. R) N(45) = N(45) + 1
000571      IF(F3 .E. R) N(46) = N(46) + 1
000575      IF(F4 .E. R) N(47) = N(47) + 1
000601      IF(F5 .E. R) N(48) = N(48) + 1
000605      IF(F6 .E. R) N(49) = N(49) + 1
000611      IF(F7 .E. R) N(50) = N(50) + 1
000615      IF(F8 .E. R) N(51) = N(51) + 1
000621      IF(F9 .E. R) N(52) = N(52) + 1
000625      IF(F10 .E. R) N(53) = N(53) + 1
000631      IF(F11 .E. R) N(54) = N(54) + 1
000635      IF(F12 .E. R) N(55) = N(55) + 1
000641      IF(G1 .E. R) N(56) = N(56) + 1
000645      IF(G2 .E. R) N(57) = N(57) + 1
000651      IF(G3 .E. R) N(58) = N(58) + 1
000655      IF(G4 .E. R) N(59) = N(59) + 1
000661      IF(G5 .E. R) N(60) = N(60) + 1
000665      IF(G6 .E. R) N(61) = N(61) + 1
000671      IF(G7 .E. R) N(62) = N(62) + 1
000675      IF(G8 .E. R) N(63) = N(63) + 1
000701      IF(G9 .E. R) N(64) = N(64) + 1
000705      IF(G10 .E. R) N(65) = N(65) + 1
000711      IF(G11 .E. R) N(66) = N(66) + 1
000715      IF(G12 .E. R) N(67) = N(67) + 1
000721      IF(H1 .E. R) N(68) = N(68) + 1
000725      IF(H2 .E. R) N(69) = N(69) + 1
000731      IF(H3 .E. R) N(70) = N(70) + 1
000735      IF(H4 .E. R) N(71) = N(71) + 1
000741      IF(H5 .E. R) N(72) = N(72) + 1
000745      IF(H6 .E. R) N(73) = N(73) + 1
000751      IF(H7 .E. R) N(74) = N(74) + 1
000755      IF(H8 .E. R) N(75) = N(75) + 1
000761      IF(H9 .E. R) N(76) = N(76) + 1
000765      IF(H10 .E. R) N(77) = N(77) + 1
000771      IF(H11 .E. R) N(78) = N(78) + 1
000775      100 CONTINUE

```

```

000777      WRITE (6,900)
001003      900 FORMAT(1H1)
001003      WRITE (6,1000)
001007      1000 FORMAT(1H ,10A,13H VARIABLE NAME,42X,10H)
001007      WRITE (6,901) N(1)
001015      901 FORMAT(1H0,10A,12H PERMEABILITY,42X,14)
001015      WRITE (6,902) N(2)
001023      902 FORMAT(1H0,10A,21H RIE SPECIFIC GRAVITY,30X,14)
001023      WRITE (6,903) N(3)
001031      903 FORMAT(1H0,10A,25H APPARENT SPECIFIC GRAVITY,35X,14)
001031      WRITE (6,904) N(4)
001037      904 FORMAT(1H0,10A,36H UNSPECIFIED TYPE OF SPECIFIC GRAVITY,24X,14)
001037      WRITE (6,905) N(5)
001045      905 FORMAT(1H0,10A,44H POROSITY,52X,14)
001045      WRITE (6,906) N(6)
001053      906 FORMAT(1H0,10A,10H ABSORPTION,50X,14)
001053      WRITE (6,907) N(7)
001061      907 FORMAT(1H0,10A,21H SCLEPSCOPE HARDNESS,34X,14)
001061      WRITE (6,908) N(8)
001067      908 FORMAT(1H0,10A,17H ABRASIVE HARDNESS,43X,14)
001067      WRITE (6,909) N(9)
001075      909 FORMAT(1H0,10A,16H IMPACT TOUGHNESS,44X,14)
001075      WRITE (6,910) N(10)
001103      910 FORMAT(1H0,10A,30H PROPAGATION VEL-LONG WAVES/LAB,30X,14)
001103      WRITE (6,911) N(11)
001111      911 FORMAT(1H0,10A,36H PROPAGATION VEL-TRANSVERSE WAVES/LAB,24X,14)
001111      WRITE (6,912) N(12)
001117      912 FORMAT(1H0,10A,29H SPECIFIC DAMPING CAPACITY/LAB,31X,14)
001117      WRITE (6,913) N(13)
001125      913 FORMAT(1H0,10A,28H POISSON S RATIO DYNAMIC/LAB,32X,14)
001125      WRITE (6,914) N(14)
001133      914 FORMAT(1H0,10A,28H LONG S MODULUS DYNAMIC/LAB,32X,14)
001133      WRITE (6,915) N(15)
001141      915 FORMAT(1H0,10A,47H MODULUS OF RIGIDITY, SHEAR MODULUS, DYNAMIC/LAB,
113X,14)
001141      WRITE (6,916) N(16)
001147      916 FORMAT(1H0,10A,21H SHEAR STRENGTH/STATIC,39X,14)
001147      WRITE (6,917) N(17)
001155      917 FORMAT(1H0,10A,28H SHEAR STRENGTH AT SS1/STATIC,32X,14)
001155      WRITE (6,918) N(18)
001163      918 FORMAT(1H0,10A,28H SHEAR STRENGTH AT SS2/STATIC,32X,14)
001163      WRITE (6,919) N(19)
001171      919 FORMAT(1H0,10A,32H PROPAGATION VEL-LONG WAVES/FIELD,28X,14)
001171      WRITE (6,920) N(20)
001177      920 FORMAT(1H0,10A,36H PROPAGATION VEL-TRANSVERSE WAVES/FIELD,22X,14)
001177      WRITE (6,921) N(21)
001205      921 FORMAT(1H0,10A,31H SPECIFIC DAMPING CAPACITY/FIELD,29X,14)
001205      WRITE (6,922) N(22)
001213      922 FORMAT(1H0,10A,30H POISSON S RATIO DYNAMIC/FIELD,30X,14)
001213      WRITE (6,923) N(23)
001221      923 FORMAT(1H0,10A,30H LONG S MODULUS DYNAMIC/FIELD,30X,14)
001221      WRITE (6,924) N(24)
001227      924 FORMAT(1H0,10A,49H MODULUS OF RIGIDITY, SHEAR MODULUS, DYNAMIC/FIEL
10,113X,14)
001227      WRITE (6,925) N(25)
001235      925 FORMAT(1H0,10A,27H SHEAR STRENGTH/STATIC/FIELD,33X,14)
001235      WRITE (6,926) N(26)
001243      926 FORMAT(1H0,10A,34H SHEAR STRENGTH AT SS1/STATIC/FIELD,26X,14)
001243      WRITE (6,927) N(27)
001251      927 FORMAT(1H0,10A,34H SHEAR STRENGTH AT SS2/STATIC/FIELD,26X,14)
001251      WRITE (6,928) N(28)
001257      928 FORMAT(1H0,10A,36H COMPRESSIVE STRENGTH, UNCONFINED/LAB,24X,14)

```

```

001257      WRITE(6,929) N(29)
001265 929 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS1/LAR,19X,14
      )
001265      WRITE(6,930) N(30)
001273 930 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS2/LAR,19X,14
      )
001273      WRITE(6,931) N(31)
001301 931 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS3/LAR,19X,14
      )
001301      WRITE(6,932) N(32)
001307 932 FORMAT(1H0,10A,33HMODULUS OF DEFORMATION AT US1/LAR,27X,14)
001307      WRITE(6,933) N(33)
001315 933 FORMAT(1H0,10A,26HPOISSON S RATIO AT US1/LAR,34X,14)
001315      WRITE(6,934) N(34)
001323 934 FORMAT(1H0,10A,26HPOISSON S RATIO AT US1/LAR,34X,14)
001331 935 FORMAT(1H0,10A,24HSET FOR UPPER STRESS/LAR,36X,14)
001331      WRITE(6,935) N(35)
001331 936 FORMAT(1H0,10A,35HCOMPRESSIVE STRENGTH, UNCONFINED/FIELD,22X,14)
001337      WRITE(6,936) N(36)
001337 937 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS1/FIELD,17X,
      )
001345      WRITE(6,937) N(37)
001345 938 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS2/FIELD,17X,
      )
001353      WRITE(6,938) N(38)
001353 939 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS3/FIELD,17X,
      )
001361      WRITE(6,939) N(39)
001361 940 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US1/FIELD,25X,14)
001367      WRITE(6,940) N(40)
001367 941 FORMAT(1H0,10A,28HPOISSON S RATIO AT US1/FIELD,32X,14)
001375      WRITE(6,941) N(41)
001375 942 FORMAT(1H0,10A,28HYOUNG S MODULUS AT US1/FIELD,32X,14)
001403      WRITE(6,942) N(42)
001403 943 FORMAT(1H0,10A,26HSET FOR UPPER STRESS/FIELD,34X,14)
001411      WRITE(6,943) N(43)
001411 944 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US2,31X,14)
001417      WRITE(6,944) N(44)
001417 945 FORMAT(1H0,10A,22HPOISSON S RATIO AT US2,38X,14)
001425      WRITE(6,945) N(45)
001425 946 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US2,38X,14)
001433      WRITE(6,946) N(46)
001433 947 FORMAT(1H0,10A,11HSET FOR US2,49X,14)
001441      WRITE(6,947) N(47)
001441 948 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US3,31X,14)
001447      WRITE(6,948) N(48)
001447 949 FORMAT(1H0,10A,22HPOISSON S RATIO AT US3,38X,14)
001455      WRITE(6,949) N(49)
001455 950 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US3,38X,14)
001463      WRITE(6,950) N(50)
001463 951 FORMAT(1H0,10A,11HSET FOR US3,49X,14)
001471      WRITE(6,951) N(51)
001471 952 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US4,31X,14)
001477      WRITE(6,952) N(52)
001477 953 FORMAT(1H0,10A,22HPOISSON S RATIO AT US4,38X,14)
001505      WRITE(6,953) N(53)
001505 954 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US4,38X,14)
001513      WRITE(6,954) N(54)
001513 955 FORMAT(1H0,10A,11HSET FOR US4,49X,14)
001521      WRITE(6,955) N(55)
001521 956 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US2/FIELD,25X,14)
001527

```

```

001527      WRITE(6,957) N(57)
001535 957 FORMAT(1H0,10A,28HPOISSON S RATIO AT US2/FIELD,32X,I4)
001535      WRITE(6,958) N(58)
001543 958 FORMAT(1H0,10A,28HYOUNG S MODULUS AT US2/FIELD,32X,I4)
001543      WRITE(6,959) N(59)
001551 959 FORMAT(1H0,10A,17HSET FOR US2/FIELD,43X,I4)
001551      WRITE(6,960) N(60)
001557 960 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US2/FIELD,25X,I4)
001557      WRITE(6,961) N(61)
001565 961 FORMAT(1H0,10A,28HPOISSON S RATIO AT US3/FIELD,32X,I4)
001565      WRITE(6,962) N(62)
001573 962 FORMAT(1H0,10A,28HYOUNG S MODULUS AT US3/FIELD,32X,I4)
001573      WRITE(6,963) N(63)
001601 963 FORMAT(1H0,10A,17HSET FOR US3/FIELD,43X,I4)
001601      WRITE(6,964) N(64)
001607 964 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US4/FIELD,25X,I4)
001607      WRITE(6,965) N(65)
001615 965 FORMAT(1H0,10A,28HPOISSON S RATIO AT US4/FIELD,32X,I4)
001615      WRITE(6,966) N(66)
001623 966 FORMAT(1H0,10A,28HYOUNG S MODULUS AT US4/FIELD,32X,I4)
001623      WRITE(6,967) N(67)
001631 967 FORMAT(1H0,10A,17HSET FOR US4/FIELD,43X,I4)
001631      WRITE(6,968) N(68)
001637 968 FORMAT(1H0,10A,16HTENSILE STRENGTH,44X,I4)
001637      WRITE(6,969) N(69)
001645 969 FORMAT(1H0,10A,41HMODULUS OF (TENSILE) DEFORMABILITY AT US1,19X,I4
1)
001645      WRITE(6,970) N(70)
001653 970 FORMAT(1H0,10A,32HPOISSON S RATIO AT US1 (TENSILE),28X,I4)
001653      WRITE(6,971) N(71)
001661 971 FORMAT(1H0,10A,32HYOUNG S MODULUS AT US1 (TENSILE),28X,I4)
001661      WRITE(6,972) N(72)
001667 972 FORMAT(1H0,10A,41HMODULUS OF (TENSILE) DEFORMABILITY AT US2,19X,I4
1)
001667      WRITE(6,973) N(73)
001675 973 FORMAT(1H0,10A,32HPOISSON S RATIO AT US2 (TENSILE),28X,I4)
001675      WRITE(6,974) N(74)
001703 974 FORMAT(1H0,10A,32HYOUNG S MODULUS AT US2 (TENSILE),28X,I4)
001703      WRITE(6,975) N(75)
001711 975 FORMAT(1H0,10A,41HMODULUS OF (TENSILE) DEFORMABILITY AT US3,19X,I4
1)
001711      WRITE(6,976) N(76)
001717 976 FORMAT(1H0,10A,32HPOISSON S RATIO AT US3 (TENSILE),28X,I4)
001717      WRITE(6,977) N(77)
001725 977 FORMAT(1H0,10A,32HYOUNG S MODULUS AT US3 (TENSILE),28X,I4)
001725      WRITE(6,978) N(78)
001733 978 FORMAT(1H0,10A,18HMODULUS OF RUPTURE,42X,I4)
001733      STOP
001735      END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS

006345

UNUSED COMPILER SPACE

012500

LOAD MAP FILE - LGO

13.27.01. 12/19/69. PAGE 1

FMA LOAD 100 LMA LOAD 13452 FMA LOADER 57303 FMA TABLES 56604 UNUSED STORAGE 43132

PROGRAM	ADDRESS	FILE	COMMON	ADDRESS	LENGTH
VC	100	LGO			
INPUTC	6445	SYSTEM			
SYSTEM	7566	SYSTEM	SCOPE2	7566	
OUTPUTC	10637	SYSTEM			
FATAL78	12177	SYSTEM			
SLOS	12315	SYSTEM			
GETRA	13633	SYSTEM			

/BLANK/ 0 0

ENTRY ADDRESS REFERENCES (RELATIVE)

VC	1 1	VC	13	15	17	21	23	25	27	31
INPUTC	6447		33	35	36	41	43	45	47	51
			53	55	57	61	63	64	67	71
			73	75	77	101	103	105	107	111
			112	115	117	121	123	125	127	131
			133	135	136	141	143	145	147	151
			153	155	157	161	162	165	167	171
			173	175	177	201	203	205	207	211
			213	215	216	221	223	225	227	231
			233	235	237	241	243	245	247	251
			252	255	257	261	263	265	267	271
			273	275	277	301	303	304		

20

KRAKEN	6551	VC	2	727						
QBNTRY	7567	VC	37	727						
SYSTEM	7774	OUTPUTC	14	1144						
SYSTEMC	7740									
SYSTEMP	7767									
END	7663	VC	1736							
STOP	7713	VC	1734							
EXIT	7705									
ABNOMRL	7723									
OUTPUTC	10641	INPUTC	40	730						
		VC	15	1145						
			1001	1007	1005	1006	1011	1013	1014	1017
			1021	1022	1025	1027	1030	1033	1035	1036
			1041	1043	1044	1047	1051	1052	1055	1057
			1060	1063	1065	1066	1071	1073	1074	1077
			1101	1102	1105	1107	1110	1113	1115	1116
			1121	1123	1124	1127	1131	1132	1135	1137
			1140	1143	1145	1146	1151	1153	1154	1157
			1161	1162	1165	1167	1170	1173	1175	1176
			1201	1203	1204	1207	1211	1212	1215	1217
			1220	1223	1225	1226	1231	1233	1234	1237
			1241	1242	1245	1247	1250	1253	1255	1256
			1261	1263	1264	1267	1271	1272	1275	1277
			1300	1303	1305	1306	1311	1313	1314	1317
			1321	1322	1325	1327	1330	1333	1335	1336
			1341	1343	1344	1347	1351	1352	1355	1357
			1360	1363	1365	1366	1371	1373	1374	1377

VARIABLE NAME

PERMEABILITY	646	COMPRESSIVE STRENGTH, TRIAXIAL AT LS3/LAB	64
TRUE SPECIFIC GRAVITY	587	MODULUS OF DEFORMATION AT US1/LAB	610
APPARENT SPECIFIC GRAVITY	1332	POISSON S RATIO AT US1/LAB	431
UNSPECIFIED TYPE OF SPECIFIC GRAVITY	72	YOUNG S MODULUS AT US1/LAB	392
POROSITY	879	SET FOR UPPER STRESS/LAB	0
ABSORPTION	301	COMPRESSIVE STRENGTH, UNCONFINED/FIELD	0
SCLENIOSCOPE HARDNESS	321	COMPRESSIVE STRENGTH, TRIAXIAL AT LS1/FIELD	37
ABRASIVE HARDNESS	118	COMPRESSIVE STRENGTH, TRIAXIAL AT LS2/FIELD	28
IMPACT TOUGHNESS	192	COMPRESSIVE STRENGTH, TRIAXIAL AT LS3/FIELD	0
PROPAGATION VEL-LONG WAVES/LAB	433	MODULUS OF DEFORMATION AT US1/FIELD	57
PROPAGATION VEL-TRANSVERSE WAVES/LAB	182	POISSON S RATIO AT US1/FIELD	20
SPECIFIC DAMPING CAPACITY/LAB	203	YOUNG S MODULUS AT US1/FIELD	20
POISSON S RATIO, DYNAMIC/LAB	392	SET FOR UPPER STRESS/FIELD	2
YOUNG S MODULUS, DYNAMIC/LAB	521	MODULUS OF DEFORMATION AT US2	273
MODULUS OF ELASTICITY, SHEAR MODULUS, DYNAMIC/LAB	434	POISSON S RATIO AT US2	279
SHEAR STRENGTH/STATIC	39	YOUNG S MODULUS AT US2	159
SHEAR STRENGTH AT SS1/STATIC	4	SET FOR US2	4
SHEAR STRENGTH AT SS2/STATIC	4	MODULUS OF DEFORMATION AT US3	279
PROPAGATION VEL-LONG WAVES/FIELD	82	POISSON S RATIO AT US3	273
PROPAGATION VEL-TRANSVERSE WAVES/FIELD	31	YOUNG S MODULUS AT US3	159
SPECIFIC DAMPING CAPACITY/FIELD	27	SET FOR US3	18
POISSON S RATIO, DYNAMIC/FIELD	29	MODULUS OF DEFORMATION AT US4	188
YOUNG S MODULUS, DYNAMIC/FIELD	61	POISSON S RATIO AT US4	185
MODULUS OF ELASTICITY, SHEAR MODULUS, DYNAMIC/FIELD	21	YOUNG S MODULUS AT US4	156
SHEAR STRENGTH/STATIC/FIELD	11	SET FOR US4	133
SHEAR STRENGTH AT SS1/STATIC/FIELD	0	MODULUS OF DEFORMATION AT US2/FIELD	81
SHEAR STRENGTH AT SS2/STATIC/FIELD	0	POISSON S RATIO AT US2/FIELD	113
COMPRESSIVE STRENGTH, UNCONFINED/LAB	935	YOUNG S MODULUS AT US2/FIELD	112
COMPRESSIVE STRENGTH, TRIAXIAL AT LS1/LAB	195	SET FOR US2/FIELD	46
COMPRESSIVE STRENGTH, TRIAXIAL AT LS2/LAB	181	MODULUS OF DEFORMATION AT US30/FIELD	1

NOT REPRODUCIBLE

POISSON'S RATIO AT US3/FIELD	0
YOUNG'S MODULUS AT US3/FIELD	0
SET FOR US3/FIELD	1
MODULUS OF DEFORMATION AT US4/FIELD	1
POISSON'S RATIO AT US4/FIELD	1
YOUNG'S MODULUS AT US4/FIELD	1
SET FOR US4/FIELD	0
TENSILE STRENGTH	918
MODULUS OF (TENSILE) DEFORMABILITY AT US1	195
POISSON'S RATIO AT US1 (TENSILE)	314
YOUNG'S MODULUS AT US1 (TENSILE)	297
MODULUS OF (TENSILE) DEFORMABILITY AT US2	466
POISSON'S RATIO AT US2 (TENSILE)	38
YOUNG'S MODULUS AT US2 (TENSILE)	324
MODULUS OF (TENSILE) DEFORMABILITY AT US3	62
POISSON'S RATIO AT US3 (TENSILE)	22
YOUNG'S MODULUS AT US3 (TENSILE)	118
MODULUS OF RUPTURE	291

MM5155. 12/19/69. PURDUE MACE 11/27/69.

09.16.42.MM515/ 3512,NAHAS,1100,CM60000,L20000
 09.16.42.(T2)*P10.
 09.16.42.MAP(UN)
 09.16.42.RUN(S)
 09.16.48.CTIME 004.129 SEC. RUN MOD LEVEL 4H
 09.16.49.REQUEST(TAPE1,556,PI,X,C=84,MT,READ)
 13.26.59. MT53 ASSIGNED - 556
 13.27.00.RFWIND(TAPE1)
 13.27.00.LGO(LC=20000)
 13.27.02.CX 4.566 SEC.
 13.27.02.PX 4.244 SEC.
 13.27.02.NL 13600
 13.38.05.STOP
 13.38.05.CP 66.574 SEC.
 13.38.05.PP 352.148 SEC.
 13.38.05.LINES = 0702 OCTAL
 13.38.05.CM 3.093 MIN-SEC.

```

000003      PROGRAM COUNT(INPUT,OUTPUT,TAPE1,TAPE2=OUTPUT)
000003      DIMENSION N(27),R(8),RR(8)
000003      DATA H/1H /
000003      DO 1 K=1,27
000005      1  N(K) = 0
000010      DO 1000 I=1,2170
000012      READ(1,2) A20,A19,A13,A17,A1,A6
000031      2  FORMAT(36X,A4,2X,A4,8X,A3,8X,A3,2X,A2,2X,A2)
000031      READ(1,3) B15,B16,RR,H25
000045      3  FORMAT(28X,A3,A3,2X,A3,A4)
000045      READ(1,4) C14
000053      4  FORMAT(28X,A3)
000053      READ(1,5) D22,D23,D24,U2,U9,D26,D18
000075      5  FORMAT(21X,A4,12X,A4,3X,A4,6X,A4,A3,A4,2X,A2)
000075      READ(1,6) (R(I),I=1,8)
000107      6  FORMAT(8A10)
000107      READ(1,7) F3,F10,F27,F4,F11,F5,F12
000131      7  FORMAT(10X,A4,A3,A4,11X,A4,A3,15X,A4,A3)
000131      READ(1,6) (RP(I),I=1,8)
000143      READ(1,8) H21,H7
000153      8  FORMAT(21X,A4,48X,A3)
000153      10 IF (A1,NE,B) N( 1) = N( 1) + 1
000157      11 IF (D2,NE,B) N( 2) = N( 2) + 1
000163      12 IF (F3,NE,B) N( 3) = N( 3) + 1
000167      13 IF (F4,NE,B) N( 4) = N( 4) + 1
000173      14 IF (F5,NE,B) N( 5) = N( 5) + 1
000177      15 IF (A6,NE,B) N( 6) = N( 6) + 1
000203      16 IF (H7,NE,B) N( 7) = N( 7) + 1
000207      17 IF (RR,NE,B) N( 8) = N( 8) + 1
000213      18 IF (D9,NE,B) N( 9) = N( 9) + 1
000217      19 IF (F10,NE,B) N(10) = N(10) + 1
000223      20 IF (F11,NE,B) N(11) = N(11) + 1
000227      21 IF (F12,NE,B) N(12) = N(12) + 1
000233      22 IF (A13,NE,B) N(13) = N(13) + 1
000237      23 IF (C14,NE,B) N(14) = N(14) + 1
000243      24 IF (R15,NE,B) N(15) = N(15) + 1
000247      25 IF (R16,NE,B) N(16) = N(16) + 1
000253      26 IF (A17,NE,B) N(17) = N(17) + 1
000257      27 IF (D18,NE,B) N(18) = N(18) + 1
000263      28 IF (A19,NE,B) N(19) = N(19) + 1
000267      29 IF (A20,NE,B) N(20) = N(20) + 1
000273      30 IF (H21,NE,B) N(21) = N(21) + 1
000277      31 IF (D22,NE,B) N(22) = N(22) + 1
000303      32 IF (D23,NE,B) N(23) = N(23) + 1
000307      33 IF (D24,NE,B) N(24) = N(24) + 1
000313      34 IF (R25,NE,B) N(25) = N(25) + 1
000317      35 IF (D26,NE,B) N(26) = N(26) + 1
000323      36 IF (F27,NE,B) N(27) = N(27) + 1
000327      1000 CONTINUE
000331      WRITE(6,909)
000335      909  FORMAT(1H)
000335      WRITE(6,999) ((K,N(K)),K=1,27)
000350      999  FORMAT(1H ,2HN(,12,4H) = ,14)
000350      STOP
000352      END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS
003436

UNUSED COMPILER SPACE
020700

Program No. 1A
COUNT PROGRAM

Number of non-blank entries for selected
variables on a tape

[illegible]

240

N(1) =	115
N(2) =	584
N(3) =	266
N(4) =	272
N(5) =	186
N(6) =	184
N(7) =	266
N(8) =	385
N(9) =	417
N(10) =	272
N(11) =	266
N(12) =	193
N(13) =	867
N(14) =	88
N(15) =	423
N(16) =	178
N(17) =	310
N(18) =	0
N(19) =	1322
N(20) =	578
N(21) =	994
N(22) =	877
N(23) =	175
N(24) =	60
N(25) =	490
N(26) =	389
N(27) =	159

MM5165A. 12/19/69.PURDUE MACE 11/27/69.

09.28.43.MM516/ 3512.NAHAS.T180.CM60000.TP1.P1
 09.28.43.0.
 09.28.43.MAP(ON)
 09.28.43.RUN(S)
 09.28.46.CTIME 000.625 SEC. RUN MOD LEVEL 4R
 09.28.47.REQUEST(TAPE1,556.HY,X.C=84,MT,READ)
 14.06.01. MT51 ASSIGNED - 556
 14.06.01.REWIND(TAPE1)
 14.06.01.LGO.
 14.06.04.CX .847 SEC.
 14.06.04.PX 3.584 SEC.
 14.06.04.NL 11100
 14.26.21.STOP
 14.26.21.CP 37.450 SEC.
 14.26.21.PP 324.078 SEC.
 14.26.21.LINFS = 0222 OCTAL
 14.26.21.CM 2.564 MWD-SFC.

Program No. 2

COUNT PROGRAM

Number of non-blank entries for both
variables in the same data set for
selected pairs of variables

PROGRAM COUNT(INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT,TAPE1)

```

000003 DIMENSION N(136)
000004 DATA N(1) /
000005 DO 1 K=1,136
000006 1 N(K) = 0
000007 DO 100 K=1,2173
000008 100 READ(1,2) A1,A2,A3
000009 100 FORMAT(32X,A1,15X,A4,7X,A4)
000010 100 READ(1,3) B4,B5,B6
000011 100 FORMAT(34X,A2,7X,A4,14X,A3)
000012 100 READ(1,4) C7
000013 100 FORMAT(39X,A4)
000014 100 READ(1,5) D8,D9
000015 100 FORMAT(30X,A4,20X,A4)
000016 100 READ(1,6) E10,E11
000017 100 FORMAT(54X,A4,3X,A4)
000018 100 READ(1,7) F12,F13,F14
000019 100 FORMAT(39X,A4,14X,A4,2X,A2)
000020 100 READ(1,8) R
000021 100 FORMAT(20X,A4)
000022 100 READ(1,9) M15,M16,M17
000023 100 FORMAT(31X,A4,10X,A4,10X,A4)
000024 100 IF( A1,NE,B ) AND( A2,NE,B )
000025 100 IF( A1,NE,B ) AND( A3,NE,B )
000026 100 IF( A1,NE,B ) AND( A4,NE,B )
000027 100 IF( A1,NE,B ) AND( A5,NE,B )
000028 100 IF( A1,NE,B ) AND( A6,NE,B )
000029 100 IF( A1,NE,B ) AND( A7,NE,B )
000030 100 IF( A1,NE,B ) AND( A8,NE,B )
000031 100 IF( A1,NE,B ) AND( A9,NE,B )
000032 100 IF( A1,NE,B ) AND( A10,NE,B )
000033 100 IF( A1,NE,B ) AND( A11,NE,B )
000034 100 IF( A1,NE,B ) AND( A12,NE,B )
000035 100 IF( A1,NE,B ) AND( A13,NE,B )
000036 100 IF( A1,NE,B ) AND( A14,NE,B )
000037 100 IF( A1,NE,B ) AND( A15,NE,B )
000038 100 IF( A1,NE,B ) AND( A16,NE,B )
000039 100 IF( A1,NE,B ) AND( A17,NE,B )
000040 100 IF( A1,NE,B ) AND( A18,NE,B )
000041 100 IF( A1,NE,B ) AND( A19,NE,B )
000042 100 IF( A1,NE,B ) AND( A20,NE,B )
000043 100 IF( A1,NE,B ) AND( A21,NE,B )
000044 100 IF( A1,NE,B ) AND( A22,NE,B )
000045 100 IF( A1,NE,B ) AND( A23,NE,B )
000046 100 IF( A1,NE,B ) AND( A24,NE,B )
000047 100 IF( A1,NE,B ) AND( A25,NE,B )
000048 100 IF( A1,NE,B ) AND( A26,NE,B )
000049 100 IF( A1,NE,B ) AND( A27,NE,B )
000050 100 IF( A1,NE,B ) AND( A28,NE,B )
000051 100 IF( A1,NE,B ) AND( A29,NE,B )
000052 100 IF( A1,NE,B ) AND( A30,NE,B )
000053 100 IF( A1,NE,B ) AND( A31,NE,B )
000054 100 IF( A1,NE,B ) AND( A32,NE,B )
000055 100 IF( A1,NE,B ) AND( A33,NE,B )
000056 100 IF( A1,NE,B ) AND( A34,NE,B )
000057 100 IF( A1,NE,B ) AND( A35,NE,B )
000058 100 IF( A1,NE,B ) AND( A36,NE,B )
000059 100 IF( A1,NE,B ) AND( A37,NE,B )
000060 100 IF( A1,NE,B ) AND( A38,NE,B )
000061 100 IF( A1,NE,B ) AND( A39,NE,B )
000062 100 IF( A1,NE,B ) AND( A40,NE,B )
000063 100 IF( A1,NE,B ) AND( A41,NE,B )
000064 100 IF( A1,NE,B ) AND( A42,NE,B )

```

```

001710 IF((D8.NE.B).AND.(H17.NE.B)) N(100) = N(100) + 1
001721 IF((D9.NE.B).AND.(E10.NE.B)) N(101) = N(101) + 1
001732 IF((D9.NE.B).AND.(E11.NE.B)) N(102) = N(102) + 1
001743 IF((D9.NE.B).AND.(F12.NE.B)) N(103) = N(103) + 1
001754 IF((D9.NE.B).AND.(F13.NE.B)) N(104) = N(104) + 1
001765 IF((D9.NE.B).AND.(F14.NE.B)) N(105) = N(105) + 1
001776 IF((D9.NE.B).AND.(H15.NE.B)) N(106) = N(106) + 1
002007 IF((D9.NE.B).AND.(H16.NE.B)) N(107) = N(107) + 1
002020 IF((D9.NE.B).AND.(H17.NE.B)) N(108) = N(108) + 1
002031 IF((F10.NE.B).AND.(E11.NE.B)) N(109) = N(109) + 1
002042 IF((E10.NE.B).AND.(F12.NE.B)) N(110) = N(110) + 1
002053 IF((E10.NE.B).AND.(F13.NE.B)) N(111) = N(111) + 1
002064 IF((E10.NE.B).AND.(F14.NE.B)) N(112) = N(112) + 1
002075 IF((E10.NE.B).AND.(H15.NE.B)) N(113) = N(113) + 1
002106 IF((E10.NE.B).AND.(H16.NE.B)) N(114) = N(114) + 1
002117 IF((E10.NE.B).AND.(H17.NE.B)) N(115) = N(115) + 1
002130 IF((E11.NE.B).AND.(F12.NE.B)) N(116) = N(116) + 1
002141 IF((E11.NE.B).AND.(F13.NE.B)) N(117) = N(117) + 1
002152 IF((E11.NE.B).AND.(F14.NE.B)) N(118) = N(118) + 1
002163 IF((E11.NE.B).AND.(H15.NE.B)) N(119) = N(119) + 1
002174 IF((E11.NE.B).AND.(H16.NE.B)) N(120) = N(120) + 1
002205 IF((E11.NE.B).AND.(H17.NE.B)) N(121) = N(121) + 1
002216 IF((F12.NE.B).AND.(F13.NE.B)) N(122) = N(122) + 1
002227 IF((F12.NE.B).AND.(F14.NE.B)) N(123) = N(123) + 1
002240 IF((F12.NE.B).AND.(H15.NE.B)) N(124) = N(124) + 1
002251 IF((F12.NE.B).AND.(H16.NE.B)) N(125) = N(125) + 1
002262 IF((F12.NE.B).AND.(H17.NE.B)) N(126) = N(126) + 1
002273 IF((F13.NE.B).AND.(F14.NE.B)) N(127) = N(127) + 1
002304 IF((F13.NE.B).AND.(H15.NE.B)) N(128) = N(128) + 1
002315 IF((F13.NE.B).AND.(H16.NE.B)) N(129) = N(129) + 1
002326 IF((F13.NE.B).AND.(H17.NE.B)) N(130) = N(130) + 1
002337 IF((F14.NE.B).AND.(H15.NE.B)) N(131) = N(131) + 1
002347 IF((F14.NE.B).AND.(H16.NE.B)) N(132) = N(132) + 1
002357 IF((F14.NE.B).AND.(H17.NE.B)) N(133) = N(133) + 1
002357 IF((H15.NE.B).AND.(H16.NE.B)) N(134) = N(134) + 1
002377 IF((H15.NE.B).AND.(H17.NE.B)) N(135) = N(135) + 1
002407 IF((H16.NE.B).AND.(H17.NE.B)) N(136) = N(136) + 1
002417 100 CONTINUE
002421 WRITE(6,900)
002425 900 FORMAT(1H1)
002425 WRITE(6,910) ((K,N(K)),K=1,136)
002440 910 FORMAT(1H 2HN(,12,4H) = ,14)
002440 STOP
002442 END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS
007110

UNUSED COMPILER SPACE

011700

PWA LOAD 100 LWA LOAD 14215 PWA LOADER 57303 PWA TABLES 57026 UNUSED STORAGE 42411

PROGRAM ADDRESS FILE COMMON ADDRESS LENGTH

COUNT 100 LGO
 INPUTC 7210 SYSTEM
 SYSTEM 10331 SYSTEM
 OUTPUTC 11402 SYSTEM
 FATAL7A 12742 SYSTEM
 SIO4 13000 SYSTEM
 GETRA 14176 SYSTEM

SCOPE2 10331

/BLANK/ 0 0

ENTRY ADDRESS REFERENCES (RELATIVE)

ENTRY	ADDRESS	COUNT	13	15	17	21	22	25	27	31
COUNT	101									
INPUTC	7212									
KRAKER	7314									
ORNTY	10332		2							
SYSTEM	10537		37	727						
SYSTEMC	10503		14	1144						
SYSTEMD	10532									
FWO	10426									
STOP	10456									
EXIT	10450									
ABACRM	10466									
OUTPUTC	11404									
KORER	11543									
FATAL7A	12743									
CARD-F	13046									
RKSPRU	13347									
FIZRAK	13357									
PGSFIL	13405									
ROPRU	13475									
DAT	13516									
CIO1	13323									
OPEN	13062									
SIO	13201									
ADVIN	13416									
MMWNS	13230									
POSFI	13426									
F178A	13440									
DAT	13746									
GETRA	14176									

SCOPE2 10331

/BLANK/ 0 0

LOAD MAP FILE - LGO

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

NO NONE

N(1) =	28	N(35) =	0	N(69) =	21	N(103) =	150
N(2) =	137	N(36) =	0	N(70) =	21	N(104) =	147
N(3) =	189	N(37) =	20	N(71) =	0	N(105) =	135
N(4) =	343	N(38) =	42	N(72) =	0	N(106) =	22
N(5) =	0	N(39) =	0	N(73) =	12	N(107) =	22
N(6) =	2	N(40) =	0	N(74) =	0	N(108) =	22
N(7) =	148	N(41) =	19	N(75) =	0	N(109) =	41
N(8) =	268	N(42) =	18	N(76) =	0	N(110) =	0
N(9) =	0	N(43) =	18	N(77) =	0	N(111) =	0
N(10) =	0	N(44) =	0	N(78) =	0	N(112) =	0
N(11) =	109	N(45) =	0	N(79) =	0	N(113) =	0
N(12) =	108	N(46) =	255	N(80) =	0	N(114) =	0
N(13) =	25	N(47) =	0	N(81) =	0	N(115) =	0
N(14) =	21	N(48) =	1	N(82) =	0	N(116) =	0
N(15) =	21	N(49) =	0	N(83) =	18	N(117) =	0
N(16) =	21	N(50) =	74	N(84) =	3	N(118) =	0
N(17) =	0	N(51) =	0	N(85) =	1	N(119) =	0
N(18) =	0	N(52) =	0	N(86) =	0	N(120) =	0
N(19) =	33	N(53) =	1	N(87) =	0	N(121) =	0
N(20) =	0	N(54) =	1	N(88) =	0	N(122) =	156
N(21) =	0	N(55) =	0	N(89) =	0	N(123) =	134
N(22) =	90	N(56) =	21	N(90) =	0	N(124) =	0
N(23) =	191	N(57) =	21	N(91) =	0	N(125) =	0
N(24) =	0	N(58) =	21	N(92) =	193	N(126) =	0
N(25) =	0	N(59) =	0	N(93) =	0	N(127) =	132
N(26) =	100	N(60) =	7	N(94) =	0	N(128) =	0
N(27) =	99	N(61) =	51	N(95) =	93	N(129) =	0
N(28) =	90	N(62) =	149	N(96) =	91	N(130) =	0
N(29) =	0	N(63) =	0	N(97) =	79	N(131) =	0
N(30) =	0	N(64) =	0	N(98) =	0	N(132) =	0
N(31) =	0	N(65) =	39	N(99) =	0	N(133) =	0
N(32) =	23	N(66) =	37	N(100) =	0	N(134) =	25
N(33) =	2	N(67) =	31	N(101) =	2	N(135) =	25
N(34) =	16	N(68) =	21	N(102) =	2	N(136) =	25

MM51140. 12/19/64.PURDUE MACF 11/27/69.

09.29.40.MM511/ 3512.NAHAS.T240.CM60000.TP1.P1
 09.29.40.0.
 09.29.40.MAP(ON)
 09.29.40.RUN(S)
 09.29.46.CTIME 003.639 SEC. RUN MOD LEVEL 48
 09.29.47.REQUEST(TAPE1,556,MY,X,C=RA,MT,READ)
 12.59.59. MT53 ASSIGNED - 556
 12.59.59.REWIND(TAPE1)
 13.00.01.LGN.
 13.00.04.CX 3.851 SEC.
 13.00.04.PX 4.230 SEC.
 13.00.04.NL 14400
 13.10.57.Stop
 13.10.57.CP 37.776 SEC.
 13.10.58.PP 313.462 SEC.
 13.10.58.LINFS = 0554 OCTAL
 13.10.58.CM 3.784 MWD-SFC.

Program No. 3

**Program to print out all values within
specified ranges (and associated rock
types) for specific selection of variables**


```

000342      WRITE(6,51)
000346      51 FORMAT(1H1,40A,12H0PERMEABILITY//)
000346      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000363      60 FORMAT(10I6,10A,13,1H1)
000363      WRITE(6,70)
000367      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000367      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000373      WRITE(6,52) (MEAN,MAX,MIN)
000407      61 FORMAT(10I6,10A,12H0PERMEABILITY//)

```

```

000407      WRITE(6,52)
000413      52 FORMAT(1H1,40A,12H0PERMEABILITY//)
000413      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000430      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000434      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000440      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000454      WRITE(6,53)
000460      53 FORMAT(1H1,40A,12H0PERMEABILITY//)
000460      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000475      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000501      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000505      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000521      WRITE(6,54)
000525      54 FORMAT(1H1,40A,12H0PERMEABILITY//)
000525      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000542      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000546      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000552      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000566      WRITE(6,55)
000572      55 FORMAT(1H1,40A,12H0PERMEABILITY//)
000572      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000587      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000613      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000617      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000633      WRITE(6,56)
000637      56 FORMAT(1H1,40A,12H0PERMEABILITY//)
000637      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000654      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000660      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000664      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000700      WRITE(6,57)
000704      57 FORMAT(1H1,40A,12H0PERMEABILITY//)
000704      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000721      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000725      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000731      WRITE(6,52) (MEAN,MAX,MIN)

```

```

000745      WRITE(6,58)
000751      58 FORMAT(1H1,40A,12H0PERMEABILITY//)
000751      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
000766      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000772      CALL MMV (N1,PERM,MEAN,MAX,MIN)
000776      WRITE(6,52) (MEAN,MAX,MIN)

```

```

001012      WRITE(6,59)
001016      59 FORMAT(1H1,40A,12H0PERMEABILITY//)
001016      WRITE(6,60) (PERM(I),RT1(I), I=1,N1)
001033      70 FORMAT(10A44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
001037      CALL MMV (N1,PERM,MEAN,MAX,MIN)
001043      WRITE(6,52) (MEAN,MAX,MIN)
001057      END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS

036440
UNUSED COMPILER SPACE
024200

```

000010      SUBROUTINE MMV (N1,40A,12H0PERMEABILITY//)
000010      DIMENSION X(2000)
000010      REAL MEAN,MAX,MIN
000010      SUM=0.
000010      MAX=0.
000010      MIN=10000.0
000012      DO 100 I=1,N
000014      SUM=SUM+X(I)
000016      IF (X(I).GE.MAX) MAX=X(I)
000023      IF (X(I).LE.MIN) MIN=X(I)
000032      MEAN=SUM/FL0AT(N)
000034      RETURN
000034      END

```

SUBPROGRAM LENGTH
000051

UNUSED COMPILER SPACE
027500

ASSUMPTION

49	(3)	0	(3)	17	(5)	57	(7)	100	(7)	1	(7)	2	(7)	14	(11)	1	(13)
4	(13)	6	(14)	0	(21)	4	(21)	1	(21)	1	(22)	1	(22)	6	(22)	3	(22)
1	(22)	4	(22)	1	(22)	1	(22)	2	(22)	2	(22)	3	(22)	4	(22)	16	(22)
4	(22)	5	(22)	4	(22)	3	(22)	2	(22)	1	(22)	2	(22)	2	(22)	3	(22)
6	(22)	4	(22)	4	(22)	3	(22)	4	(22)	2	(23)	2	(24)	2	(24)	3	(40)
0	(40)	2	(45)	0	(45)	2	(45)	3	(45)	3	(45)	3	(45)	4	(45)	7	(45)
17	(48)	4	(52)	2	(54)	105	(208)	4	(208)	3	(208)	7	(208)	23	(208)	17	(211)
13	(12)	478	(214)	10	(215)	5	(215)	18	(214)	50	(214)	46	(214)	46	(214)	1	(220)
60	(223)	71	(223)	200	(223)	200	(223)	200	(223)	12	(223)	12	(223)	10	(223)	21	(223)
4	(223)	4	(223)	14	(223)	14	(223)	24	(223)	81	(223)	64	(223)	111	(223)	75	(223)
78	(223)	00	(223)	111	(223)	129	(223)	69	(223)	108	(223)	77	(223)	109	(223)	34	(223)
63	(223)	84	(223)	5	(223)	3	(223)	1	(223)	5	(223)	4	(223)	2	(223)	3	(223)
7	(223)	60	(223)	17	(223)	1	(223)	1	(223)	12	(223)	3	(223)	4	(223)	39	(223)
6	(223)	46	(223)	61	(223)	1	(223)	89	(236)	89	(236)	42	(236)	23	(236)	30	(236)
18	(236)	125	(236)	110	(236)	94	(236)	127	(236)	119	(236)	147	(236)	98	(236)	111	(236)
122	(236)	107	(236)	142	(236)	106	(236)	115	(236)	103	(236)	114	(236)	120	(236)	108	(236)
120	(236)	111	(236)	130	(236)	125	(236)	127	(236)	127	(236)	104	(236)	120	(236)	110	(236)
110	(236)	110	(236)	110	(236)	110	(236)	120	(236)	120	(236)	120	(236)	120	(236)	132	(236)
126	(236)	126	(236)	126	(236)	126	(236)	97	(236)	97	(236)	127	(236)	127	(236)	104	(236)
133	(236)	141	(236)	121	(236)	142	(236)	126	(236)	126	(236)	133	(236)	131	(236)	135	(236)
131	(236)	127	(236)	121	(236)	102	(236)	137	(236)	132	(236)	123	(236)	130	(236)	130	(236)
135	(236)	114	(236)	112	(236)	118	(236)	117	(236)	130	(236)	125	(236)	131	(236)	35	(236)
114	(236)	90	(236)	93	(236)	127	(236)	95	(236)	89	(236)	114	(236)	95	(236)	139	(236)
149	(236)	128	(236)	110	(236)	117	(236)	109	(236)	100	(236)	119	(236)	119	(236)	155	(236)
87	(236)	63	(236)	104	(236)	121	(236)	135	(236)	141	(236)	101	(236)	125	(236)	128	(236)
117	(236)	114	(236)	134	(236)	111	(236)	46	(236)	17	(236)	8	(236)	8	(236)	41	(236)
79	(236)	42	(236)	6	(236)	52	(236)	23	(236)	69	(236)	36	(236)	37	(236)	27	(236)
58	(236)	47	(236)	52	(236)	47	(236)	78	(236)	34	(236)	36	(236)	37	(236)	50	(236)
39	(236)	85	(236)	23	(236)	25	(236)	21	(236)	94	(236)	7	(236)	138	(236)	29	(236)
51	(236)	88	(236)	109	(236)	8	(236)	16	(236)	8	(236)	11	(236)	7	(236)	22	(236)
1	(400)	4	(414)	1	(404)	1	(404)	5	(404)	1	(410)	3	(410)	2	(410)	1	(410)
1	(410)	4	(410)	2	(410)	3	(410)	1	(410)	2	(413)	1	(413)	1	(413)	17	(413)
13	(415)	1	(415)	10	(415)	3	(415)	7	(417)	0	(417)	1	(420)	2	(420)	1546	(438)
8	(448)	1	(448)	17	(443)												

(NUMBERS IN PARENTHESES REFER TO HOOK TYPES)

N = 333
 MEAN = 56
 MAX = 1546
 MTN = 0

NOT REPRODUCIBLE

SCLEKOSCOPIC HARDNESS

74 (3)	90 (3)	71 (5)	77 (7)	69 (7)	84 (7)	77 (7)	53 (7)	71 (7)	49 (7)
85 (7)	71 (7)	76 (7)	77 (7)	69 (7)	100 (11)	92 (13)	76 (13)	70 (13)	41 (13)
80 (13)	84 (13)	84 (13)	64 (13)	63 (13)	96 (13)	72 (13)	90 (14)	95 (14)	84 (14)
82 (14)	83 (14)	77 (14)	63 (14)	63 (14)	71 (14)	70 (14)	66 (14)	92 (21)	40 (21)
83 (21)	66 (21)	80 (21)	76 (21)	50 (22)	88 (22)	95 (22)	95 (22)	95 (22)	98 (22)
100 (22)	44 (22)	45 (22)	40 (22)	85 (22)	89 (22)	85 (22)	37 (22)	59 (22)	53 (22)
92 (22)	94 (22)	104 (22)	97 (22)	99 (22)	99 (22)	95 (22)	91 (22)	95 (22)	97 (22)
85 (22)	100 (22)	51 (22)	97 (22)	75 (22)	100 (22)	80 (22)	83 (22)	99 (23)	65 (34)
87 (39)	66 (45)	79 (45)	93 (45)	64 (45)	88 (45)	75 (45)	70 (47)	40 (47)	10 (47)
34 (47)	49 (47)	58 (47)	69 (48)	64 (48)	64 (48)	69 (48)	77 (48)	77 (48)	17 (48)
86 (48)	74 (50)	40 (52)	82 (52)	70 (52)	83 (59)	80 (59)	54 (64)	57 (65)	58 (65)
57 (65)	51 (65)	73 (65)	73 (65)	98 (211)	24 (201)	7 (201)	101 (208)	84 (208)	104 (208)
27 (208)	67 (211)	40 (211)	74 (211)	98 (211)	30 (212)	17 (212)	74 (218)	56 (218)	69 (218)
71 (218)	42 (218)	51 (218)	53 (218)	58 (218)	39 (218)	36 (218)	52 (218)	27 (223)	27 (223)
58 (223)	52 (223)	61 (223)	66 (223)	54 (223)	93 (223)	56 (223)	64 (223)	52 (223)	61 (223)
58 (223)	52 (223)	52 (223)	13 (223)	30 (223)	36 (223)	55 (223)	46 (223)	10 (223)	13 (223)
16 (223)	16 (223)	5 (223)	52 (223)	48 (223)	51 (223)	33 (223)	48 (223)	54 (223)	59 (223)
45 (223)	48 (223)	54 (223)	54 (223)	59 (223)	65 (223)	48 (223)	41 (223)	54 (223)	52 (223)
46 (223)	41 (223)	50 (223)	46 (223)	34 (223)	53 (223)	52 (223)	52 (223)	3 (223)	38 (223)
52 (223)	14 (223)	54 (223)	13 (223)	42 (223)	42 (223)	52 (223)	52 (223)	14 (223)	54 (223)
13 (223)	58 (223)	50 (223)	57 (223)	64 (223)	44 (223)	65 (223)	58 (223)	43 (223)	37 (223)
28 (223)	45 (223)	40 (223)	48 (223)	47 (223)	60 (223)	64 (223)	36 (223)	61 (223)	40 (223)
14 (223)	23 (223)	42 (223)	47 (223)	56 (223)	47 (223)	59 (223)	47 (223)	57 (223)	44 (223)
46 (223)	51 (223)	49 (223)	55 (223)	61 (223)	23 (223)	59 (223)	31 (223)	20 (223)	31 (223)
20 (223)	54 (223)	50 (223)	53 (223)	63 (223)	62 (223)	50 (223)	23 (223)	33 (223)	50 (223)
55 (223)	44 (223)	45 (223)	29 (223)	36 (223)	42 (223)	54 (223)	32 (223)	21 (223)	21 (223)
25 (223)	21 (223)	25 (223)	21 (223)	25 (223)	55 (223)	42 (223)	54 (223)	26 (223)	42 (223)
24 (223)	34 (223)	61 (223)	26 (223)	34 (223)	26 (223)	28 (223)	31 (223)	51 (223)	43 (223)
39 (223)	57 (223)	73 (223)	61 (223)	42 (223)	64 (223)	22 (223)	51 (223)	50 (223)	28 (223)
14 (223)	45 (223)	65 (223)	34 (223)	71 (223)	58 (223)	51 (223)	62 (223)	36 (223)	38 (223)
42 (223)	37 (223)	31 (223)	48 (223)	42 (223)	44 (223)	50 (223)	13 (223)	53 (223)	71 (223)
73 (223)	49 (223)	41 (223)	20 (223)	19 (223)	29 (223)	12 (223)	95 (223)	34 (223)	95 (223)
92 (400)	84 (400)	46 (400)	93 (400)	64 (400)	64 (400)	78 (400)	81 (400)	97 (400)	99 (400)
97 (404)	99 (404)	75 (404)	81 (404)	85 (404)	89 (404)	75 (404)	71 (404)	92 (404)	74 (404)
74 (404)	60 (404)	87 (404)	99 (404)	71 (404)	96 (404)	81 (404)	31 (404)	71 (404)	40 (404)
37 (407)	40 (407)	40 (407)	78 (407)	64 (407)	60 (407)	56 (407)	56 (407)	49 (407)	46 (407)
43 (410)	39 (410)	40 (410)	46 (410)	38 (410)	66 (410)	55 (410)	42 (410)	43 (410)	52 (410)
41 (410)	49 (410)	40 (410)	81 (410)	55 (410)	83 (410)	73 (410)	73 (410)	77 (410)	34 (410)
74 (413)	71 (413)	97 (413)	94 (413)	78 (413)	74 (413)	82 (413)	71 (413)	41 (413)	79 (413)
56 (417)	46 (417)	75 (417)	59 (417)	59 (417)	72 (417)	65 (417)	78 (417)	47 (417)	68 (417)
56 (423)	42 (423)	31 (423)	42 (423)	46 (423)	85 (423)	84 (423)	85 (423)	22 (423)	6 (423)
6 (607)	85 (622)	63 (622)	96 (622)	74 (622)	51 (622)	49 (622)	50 (622)	53 (622)	58 (622)
52 (622)	57 (622)	44 (622)	50 (622)	72 (622)	60 (622)	53 (622)	51 (622)	99 (622)	18 (622)
41 (648)	42 (648)	46 (648)	46 (648)	46 (648)	46 (648)	46 (648)	46 (648)	46 (648)	46 (648)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 433
 MEAN = 59
 MAX = 104
 MIN = 6

MH33060. 03/17/70. PURDUE MACE 02/07/70.

11.13.22.MH330/ 15073. PERLOFF, TIR0, CM65000, L30
11.13.22.00. TP1. P10.
11.13.22. RUY(S)
11.47.19. CTIME 001.395 SEC. RUN MOD LEVEL 4B
11.47.20. REQUEST (TAPE1, 556, HY, A, C=H4, MT, RFAD)
02.08.31. MT51 ASSIGNED - 556
02.08.32. REWIND (TAPE1)
02.08.32. LGO.
02.08.34. CX 1.533 SFC.
02.08.34. PX 5.411 SFC.
02.08.34. NL 43000
02.32.54. END MAIN
02.32.54. CP 47.039 SFC.
02.32.54. PP 340.884 SFC.
02.32.54. LINES = 1521 OCTAL
02.32.54. CM 3.041 MBD-SEC.

Program No. 4

**Program to print out a particular rock
type of all values within specified ranges
for a specific selection of variables**

000174	31	N1 = N1 + 1	
000176		DECODE(90,40,CARD(1)) PERM(N1)	
000205	40	FORMAT(72X,F4.0)	
000205		RT1(N1) = IRT(K)	
000210		GO TO 12	
000210	32	N2 = N2 + 1	
000212		DECODE(90,41,CARD(1)) SGT(N2)	
000221	41	FORMAT(36X,F4.0)	
000221		RT2(N2) = IRT(K)	
000224		GO TO 13	
000224	33	N3 = N3 + 1	
000226		DECODE(90,42,CARD(1)) SGA(N3)	
000235	42	FORMAT(42X,F4.0)	
000235		RT3(N3) = IRT(K)	
000240		GO TO 14	
000240	34	N4 = N4 + 1	
000242		DECODE(90,43,CARD(1)) SGU(N4)	
000251	43	FORMAT(48X,F4.0)	
000251		RT4(N4) = IRT(K)	
000254		GO TO 15	
000254	35	N5 = N5 + 1	
000256		DECODE(90,44,CARD(1)) PORO(N5)	
000265	44	FORMAT(54X,F3.0)	
000265		RT5(N5) = IRT(K)	
000270		GO TO 16	
000270	36	N6 = N6 + 1	
000272		DECODE(90,45,CARD(1)) ABSOP(N6)	
000301	45	FORMAT(50X,F4.0)	
000301		RT6(N6) = IRT(K)	
000304		GO TO 17	
000304	37	N7 = N7 + 1	
000306		DECODE(90,46,CARD(1)) SCLERO(N7)	
000315	46	FORMAT(65X,F3.0)	
000315		RT7(N7) = IRT(K)	
000320		GO TO 18	
000320	38	N8 = N8 + 1	
000322		DECODE(90,47,CARD(1)) ABRAS(N8)	
000341	47	FORMAT(70X,F2.0)	
000341		RT8(N8) = IRT(K)	
000344		GO TO 19	
000344	39	N9 = N9 + 1	
000346		DECODE(90,48,CARD(1)) TOUGH(N9)	
000345	48	FORMAT(74X,F3.0)	
000345		RT9(N9) = IRT(K)	
000350		GO TO 100	
000350	50	IF(N1.EQ.0) GO TO 501	
000351		WRITE(4,51)	
000355	51	FORMAT(1,HL,40X,12HPFRMEABILITY//)	
000355		WRITE(6,60) (PERM(J),RT1(J),I=1,N1)	
000372	60	FORMAT(100(F6.0,2H /,13,1W))	

```

000003      PROGRAM MAIN(INPUT,OUTPUT,TAPFS=INPUT,TAPF6=OUTPUT,TAPF1)
000003      DATA 01,02,03,04/4400,47,2026,44/
000003      DATA X14 /
000003      DIMENSION CAPD(4)
000003      DIMENSION IRT(2171),DUMMY(2), PERM(1000),SGT(700),SGA(1600),SGU(30
10) ,PRMT(300),AHSNP(500),SCLEPO(500),APRAS(200),TOUGH(300),PT1(100
20),RT2(700),RT3(1600),RT4(300),RT5(1300),RT6(500),RT7(500),RT8(200
3),RT9(300)
000003      INTEGER RT1,RT2,RT3,RT4,RT5,RT6,RT7,RT8,RT9,DUMMY
000003      INTEGER AP,AS
000003      REAL MEAN,MAX,MIN
000003      EQUIVALENCE (IRT(1),DUMMY(2))
000003      DUMMY(1) = 3
000003      N1 = 0
000003      N2 = 0
000003      N3 = 0
000003      N4 = 0
000003      N5 = 0
000003      N6 = 0
000003      N7 = 0
000003      N8 = 0
000003      N9 = 0
000003      DO 100 I=1,2171
000003          READ(1,400) CAPD
000003          400 FORMAT(2A10)
000003          DECODE(10,2,CAPD(1)) IRT(K),A1,A2,A3,A4,A5,A6,A7,A8,A9
000003      2   FORMAT(15X,I3,14X,A1,3X,I4,2X,I4,2X,A4,2X,A3,2X,A4,2X,A3,2X,A7,2X,
1A3)
000003          IF(EOF(1)) 99,101
000003      101 READ(1,5) R
000003          READ(1,7) C
000003          READ(1,21) D
000003          READ(1,7) E
000003          READ(1,7) F
000003          READ(1,21) G
000003          READ(1,7) H
000003      1   FORMAT(5X,A5)
000003          IF((RT1).NE.IRT(K-1)) GO TO 50
000003      11 IF(21.NE.X) GO TO 31
000003      12 IF((A2).Y.6400).AND.(A2.GT.97)) GO TO 32
000003      13 IF((A3).I4.2026).AND.(A3.GT.48)) GO TO 33
000003      14 IF(A4.NE.X) GO TO 34
000003      15 IF(A5.NE.X) GO TO 35
000003      16 IF(A6.NE.X) GO TO 36
000003      17 IF(A7.NE.X) GO TO 37
000003      18 IF(A8.NE.X) GO TO 38
000003      19 IF(A9.NE.X) GO TO 39
000003      GO TO 100
000174

```

```

000372      WRITE(6,70)
000373      CALL MMW(N1,PERM,MEAN,MAX,MIN)
000374      WRITE(6,61) N1,MEAN,MAX,MIN
000416      FORMAT(//5X,3H1 =,1A/2X,6HMFAN =,F6.0/3X,5HMIN =,
000416      1F6.0)
000416      501 IF(N2,FO,0) GO TO 502
000416      WRITE(6,52)
000416      52 FORMAT(//40X,21HTRUF SPECIFIC GRAVITY//)
000423      WRITE(6,60) (SGT(I),I=1,N2)
000430      WRITE(6,70)
000440      CALL MMW(N2,SGT,MEAN,MAX,MIN)
000450      WRITE(6,61) N2,MEAN,MAX,MIN
000450      502 IF(N3,FO,0) GO TO 503
000450      WRITE(6,53)
000450      53 FORMAT(//40X,25HAPPARENT SPECIFIC GRAVITY//)
000471      WRITE(6,60) (SGA(I),I=1,N3)
000504      WRITE(6,70)
000512      CALL MMW(N3,SGA,MEAN,MAX,MIN)
000516      WRITE(6,61) N3,MEAN,MAX,MIN
000522      503 IF(N4,FO,0) GO TO 504
000522      WRITE(6,54)
000522      54 FORMAT(//40X,34HUNSPECIFIED TYPE OF SPECIFIC GRAVITY//)
000527      WRITE(6,60) (SGU(I),I=1,N4)
000527      WRITE(6,70)
000554      CALL MMW(N4,SGU,MEAN,MAX,MIN)
000560      WRITE(6,61) N4,MEAN,MAX,MIN
000564      504 IF(N5,FO,0) GO TO 505
000600      WRITE(6,55)
000601      55 FORMAT(//40X,8HPOROSITY//)
000605      WRITE(6,60) (PORO(I),I=1,N5)
000622      WRITE(6,70)
000622      CALL MMW(N5,PORO,MEAN,MAX,MIN)
000622      WRITE(6,61) N5,MEAN,MAX,MIN
000632      505 IF(N6,FO,0) GO TO 506
000632      WRITE(6,56)
000632      56 FORMAT(//40X,10HABSORPTION//)
000632      WRITE(6,60) (ABSOP(I),I=1,N6)
000632      WRITE(6,70)
000632      CALL MMW(N6,ABSOP,MEAN,MAX,MIN)
000632      WRITE(6,61) N6,MEAN,MAX,MIN
000632      506 IF(N7,FO,0) GO TO 507
000632      WRITE(6,57)
000632      57 FORMAT(//40X,21HSCLEROSCOPIC HARDNESS//)
000632      WRITE(6,60) (SCLERO(I),I=1,N7)
000632      WRITE(6,70)
000632      CALL MMW(N7,SCLERO,MEAN,MAX,MIN)
000632      WRITE(6,61) N7,MEAN,MAX,MIN
000632      507 IF(N8,FO,0) GO TO 508
000632      WRITE(6,58)
000632      58 FORMAT(//40X,17HABRASIVE HARDNESS//)
000632      WRITE(6,60) (ABRAS(I),I=1,N8)
000632      WRITE(6,70)
000632      CALL MMW(N8,ABRAS,MEAN,MAX,MIN)
000632      WRITE(6,61) N8,MEAN,MAX,MIN
000632      508 IF(N9,FO,0) GO TO 509
000632      WRITE(6,59)
000632      59 FORMAT(//40X,16HIMPACT TOUGHNESS//)
000632      WRITE(6,60) (TOUGH(I),I=1,N9)
000632      WRITE(6,70)
000632      CALL MMW(N9,TOUGH,MEAN,MAX,MIN)
000632      WRITE(6,61) N9,MEAN,MAX,MIN
001062

```

```

001074      60 DO 81 J=1,N1
001100      PERM(J) = 0.0
001100      81 RT1(J) = 1.0
001104      DO 82 J=1,N2
001104      SGT(J) = 0.0
001107      82 RT2(J) = 0.0
001112      DO 83 J=1,N3
001112      SGA(J) = 0.0
001114      83 RT3(J) = 0.0
001115      DO 84 J=1,N4
001120      SGU(J) = 0.0
001122      84 RT4(J) = 0.0
001123      DO 85 J=1,N5
001123      PORO(J) = 0.0
001124      85 RT5(J) = 0.0
001134      DO 86 J=1,N6
001136      ARSOP(J) = 0.0
001137      86 RT6(J) = 0.0
001142      DO 87 J=1,N7
001144      SCLERO(J) = 0.0
001145      87 RT7(J) = 0.0
001150      DO 88 J=1,N8
001152      ABRAS(J) = 0.0
001153      88 RT8(J) = 0.0
001156      DO 89 J=1,N9
001156      TOUGH(J) = 0.0
001161      89 RT9(J) = 0.0
001164      N1 = 0
001165      N2 = 0
001166      N3 = 0
001167      N4 = 0
001170      N5 = 0
001171      N6 = 0
001172      N7 = 0
001173      N8 = 0
001174      N9 = 0
001174      GO TO 11
001175      100 CONTINUE
001177      99 IF(N1,FO,0) GO TO 202
001200      WRITE(6,51)
001204      WRITE(6,60) (PERM(I),I=1,N1)
001221      WRITE(6,70)
001225      CALL MMW(N1,PERM,MEAN,MAX,MIN)
001231      WRITE(6,61) N1,MEAN,MAX,MIN
001245      202 IF(N2,FO,0) GO TO 203
001246      WRITE(6,52)
001252      WRITE(6,60) (SGT(I),I=1,N2)
001257      WRITE(6,70)
001257      CALL MMW(N2,SGT,MEAN,MAX,MIN)
001277      WRITE(6,61) N2,MEAN,MAX,MIN
001277      203 IF(N3,FO,0) GO TO 204
001314      WRITE(6,53)
001320      WRITE(6,60) (SGA(I),I=1,N3)
001325      WRITE(6,70)
001325      CALL MMW(N3,SGA,MEAN,MAX,MIN)
001341      WRITE(6,61) N3,MEAN,MAX,MIN
001345      204 IF(N4,FO,0) GO TO 205
001346      WRITE(6,54)
001346      WRITE(6,60) (SGU(I),I=1,N4)
001346      WRITE(6,70)
001346      CALL MMW(N4,SGU,MEAN,MAX,MIN)
001407      WRITE(6,61) N4,MEAN,MAX,MIN
001412

```

34

```

001497      205 IF (N5.F0.0) GO TO 206
001498      WRITE (6.65)
001499      WRITE (6.60) (PORO(I),RT5(I),I=1,N5)
001500      WRITE (6.70)
001501      CALL MMM(N5,PORO,MEAN,MAX,MIN)
001502      WRITE (6.61) N5,MEAN,MAX,MIN
001503      206 IF (N6.F0.0) GO TO 207
001504      WRITE (6.66)
001505      WRITE (6.60) (ABSOP(I),RT6(I),I=1,N6)
001506      WRITE (6.70)
001507      CALL MMM(N6,ABSOP,MEAN,MAX,MIN)
001508      WRITE (6.61) N6,MEAN,MAX,MIN
001509      207 IF (N7.F0.0) GO TO 208
001510      WRITE (6.57)
001511      WRITE (6.60) (SCLERO(I),RT7(I),I=1,N7)
001512      WRITE (6.70)
001513      CALL MMM(N7,SCLERO,MEAN,MAX,MIN)
001514      WRITE (6.61) N7,MEAN,MAX,MIN
001515      208 IF (N8.F0.0) GO TO 209
001516      WRITE (6.58)
001517      WRITE (6.60) (ABRAS(I),RT8(I),I=1,N8)
001518      WRITE (6.70)
001519      CALL MMM(N8,ABRAS,MEAN,MAX,MIN)
001520      WRITE (6.61) N8,MEAN,MAX,MIN
001521      209 IF (N9.F0.0) GO TO 210
001522      WRITE (6.59)
001523      WRITE (6.60) (TOUGH(I),RT9(I),I=1,N9)
001524      WRITE (6.70)
001525      CALL MMM(N9,TOUGH,MEAN,MAX,MIN)
001526      WRITE (6.61) N9,MEAN,MAX,MIN
001527      210 STOP
001528      END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS
002606

UNUSED COMPILER SPACE
015100

```

SUBROUTINE MMM(N,X,MEAN,MAX,MIN)
000010      DIMENSION X(2000)
000011      REAL MEAN,MAX,MIN
000012      SUM = 0.0
000013      MAX = 0.0
000014      MIN = 10000.0
000015      DO 100 I=1,N
000016      SUM = SUM + X(I)
000017      IF (X(I).GE.MAX) MAX = X(I)
000018      IF (X(I).LE.MIN) MIN = X(I)
000019      100
000020      MEAN = SUM/FLOAT(N)
000021      RETURN
000022      END

```

SUBPROGRAM LENGTH
000051

UNUSED COMPILER SPACE
022500

LOAD MAP FILE - LGO

FPA LOAD 100 LWA LOAD 47760 FWA LOADER 57277 FWA TABLES 56513 UNUSED STORAGE 4533

PROGRAM	ADDRESS	FILE	COMMON	ADDRESS	LENGTH
MAIN	100	LGO			
MMH	42566	LGO			
IFENDF	42637	SYSTEM			
INPUTC	42671	SYSTEM			
SYSTEM	44012	SYSTEM	SCOPE2	44012	
INPUTS	45063	SYSTEM			
OUTPTC	45145	SYSTEM			
FATAL78	44505	SYSTEM			
SLOS	44623	SYSTEM			
GETRA	47741	SYSTEM			

/BLANK/ 0 0

ENTRY ADDRESS REFERENCES (RELATIVE)

MAIN	101	MAIN	401	447	515	563	631	677	748	1013
MMH	42567	MAIN	1061	230	1276	1344	1412	1460	1526	1574
			1642	1710						
IFENDE	42640	MAIN	56							
INPUTC	42673	MAIN	17	21	22	62	64	65	70	72
			73	76	100	101	104	106	107	112
			114	115	120	122	123	124	130	131
KRAKFR	42775	INPUTS	5	23						
QBNTRY	44013	MAIN	2							
SYSTEM	44220	IFENDF	22							
		INPUTC	37	727						
		INPUTS	26							
		OUTPTC	14	1144						
SYSTEMC	44144	MAIN	1730							
SYSTEMP	44213	MMH	35							
END	44107	MAIN	1726							
STOP	44137									
EXIT	44131									
ABNORML	44147	IFENDF	23							
		INPUTC	40	730						
		INPUTS	27							
		OUTPTC	15	1145						
INPUTS	45065	MAIN	25	27	31	33	35	37	41	43
			45	47	51	53	54	177	201	203
			204	213	215	217	220	227	231	233
			234	243	245	247	250	257	261	263
			264	273	275	277	300	307	311	313
			314	323	325	327	330	337	341	343
			344							
			353	354	357	363	365	371	374	375
OUTPTC	45147	MAIN	404	406	410	412	414	415	421	422
			425	431	433	437	442	443	452	454
			456	460	462	463	467	470	473	477
			501	505	510	511	520	522	524	526
			530	531	535	536	541	545	547	553

ENTRY ADDRESS REFERENCES (RELATIVE)

	556	557	566	570	572	574	576	577
	603	604	607	613	615	621	624	625
	634	636	640	642	644	645	651	652
	661	663	667	667	672	702	702	704
	706	710	712	713	717	720	723	727
	731	735	740	741	750	752	754	758
	760	761	765	766	771	775	777	783
	1006	1007	1016	1020	1022	1024	1026	1027
	1033	1034	1037	1043	1045	1051	1054	1055
	1064	1066	1070	1072	1074	1075	1202	1203
	1206	1212	1214	1220	1223	1224	1233	1235
	1237	1241	1243	1244	1250	1251	1254	1260
	1262	1266	1271	1272	1301	1303	1305	1307
	1311	1312	1316	1317	1322	1326	1330	1334
	1337	1340	1347	1351	1353	1355	1357	1360
	1364	1365	1370	1374	1376	1402	1405	1406
	1415	1417	1421	1423	1425	1426	1432	1433
	1436	1442	1444	1450	1453	1454	1463	1465
	1467	1471	1473	1474	1500	1501	1504	1510
	1512	1516	1521	1522	1531	1533	1535	1537
	1541	1542	1546	1547	1552	1556	1560	1564
	1567	1570	1577	1601	1603	1605	1607	1610
	1614	1615	1620	1624	1626	1632	1634	1636
	1645	1647	1651	1653	1655	1656	1662	1663
	1666	1672	1674	1700	1703	1704	1713	1715
	1717	1721	1723	1724				

KODER	45306							
FATAL7A	44506	SYSTEM	710					
CARD.F	44603	SIO	56					
BKSPRU.	47112							
FIZBAK.	47122	OUTPIC	32					
POSEIL.	47150							
RDPRU.	47240							
DAT.	47261	OUTPIC	56	72	37			
CIOI.	47066							
OPEN.	44623							
SIO.	44744	INPUTC	22					
ADVIN.	47161	SYSTEM	506					
MYMOS.	44773	OUTPIC	71					
POSFI.	47171	SYSTEM	410					
FIZBA.	47203							
DAT..	47511	INPUTC	56	24	47			
		INPUTS	17	33	42			
SEIRA	47741	IFENDE	4			45		
		INPUTC	10					
		OUTPIC	10					

UNSATISFIED FTERNALS REFERENCES (RELATIVE)

** NONE **

PERMEABILITY

2 (45)	3 (45)	2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
2 (45)	3 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
2 (45)	2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
2 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)
2 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 102
MEAN = 1
MAX = 3
MIN = 1

TRUE SPECIFIC GRAVITY

217 (45)	247 (45)	252 (45)	252 (45)	242 (45)	246 (45)	255 (45)	255 (45)	242 (45)	250 (45)
241 (45)	259 (45)	254 (45)	254 (45)	254 (45)	252 (45)	250 (45)	252 (45)	256 (45)	242 (45)
251 (45)	251 (45)	250 (45)	259 (45)	257 (45)	256 (45)	254 (45)	253 (45)	255 (45)	243 (45)
261 (45)	249 (45)	263 (45)	261 (45)	261 (45)	261 (45)	250 (45)	259 (45)	260 (45)	264 (45)
251 (45)	249 (45)	248 (45)	259 (45)	240 (45)	257 (45)	250 (45)	256 (45)	260 (45)	242 (45)
252 (45)	256 (45)	250 (45)	257 (45)	257 (45)	258 (45)	257 (45)	256 (45)	260 (45)	240 (45)
261 (45)	261 (45)	259 (45)	259 (45)	257 (45)	261 (45)	259 (45)	257 (45)	261 (45)	254 (45)
241 (45)	242 (45)	260 (45)	256 (45)	254 (45)	261 (45)	259 (45)	260 (45)	261 (45)	241 (45)
255 (45)	254 (45)	261 (45)	258 (45)	257 (45)	261 (45)	259 (45)	257 (45)	256 (45)	240 (45)
263 (45)	263 (45)	256 (45)	261 (45)	251 (45)	261 (45)	259 (45)	257 (45)	256 (45)	243 (45)
257 (45)	247 (45)	271 (45)	281 (45)	287 (45)	281 (45)	258 (45)	257 (45)	255 (45)	255 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 106
MEAN = 257
MAX = 287
MIN = 217

APPARENT SPECIFIC GRAVITY

270 (45)	194 (45)	135 (45)	211 (45)	229 (45)	209 (45)	224 (45)	212 (45)	217 (45)	229 (45)
201 (45)	232 (45)	239 (45)	225 (45)	226 (45)	222 (45)	230 (45)	225 (45)	231 (45)	240 (45)
227 (45)	231 (45)	234 (45)	229 (45)	215 (45)	226 (45)	224 (45)	226 (45)	224 (45)	242 (45)
244 (45)	239 (45)	239 (45)	239 (45)	209 (45)	243 (45)	228 (45)	230 (45)	235 (45)	223 (45)
248 (45)	176 (45)	184 (45)	243 (45)	224 (45)	217 (45)	230 (45)	230 (45)	228 (45)	223 (45)
229 (45)	232 (45)	226 (45)	236 (45)	218 (45)	234 (45)	197 (45)	197 (45)	221 (45)	221 (45)
221 (45)	231 (45)	231 (45)	234 (45)	234 (45)	234 (45)	237 (45)	243 (45)	214 (45)	225 (45)
244 (45)	244 (45)	238 (45)	219 (45)	248 (45)	243 (45)	252 (45)	244 (45)	249 (45)	245 (45)
249 (45)	245 (45)	240 (45)	226 (45)	232 (45)	219 (45)	235 (45)	245 (45)	246 (45)	245 (45)
255 (45)	255 (45)	255 (45)	239 (45)	237 (45)	237 (45)	237 (45)	245 (45)	233 (45)	223 (45)
227 (45)	228 (45)	284 (45)	315 (45)	288 (45)	254 (45)	247 (45)	238 (45)	254 (45)	243 (45)
242 (45)	265 (45)	269 (45)	279 (45)	284 (45)	276 (45)		246 (45)	254 (45)	243 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 111
 MEAN = 235
 MAX = 315
 MIN = 135

N.B. These refer to Apparent Specific Gravity
 on preceding page.

UNSPECIFIED TYPE OF SPECIFIC GRAVITY

252 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 1
 MEAN = 252
 MAX = 252
 MIN = 252

POROSITY

104 (45)	455 (45)	143 (45)	143 (45)	93 (45)	137 (45)	115 (45)	172 (45)	140 (45)	55 (45)	107 (45)
100 (45)	71 (45)	112 (45)	108 (45)	108 (45)	126 (45)	89 (45)	100 (45)	81 (45)	101 (45)	93 (45)
91 (45)	59 (45)	83 (45)	164 (45)	164 (45)	118 (45)	124 (45)	115 (45)	113 (45)	53 (45)	41 (45)
93 (45)	93 (45)	93 (45)	198 (45)	198 (45)	67 (45)	126 (45)	80 (45)	93 (45)	104 (45)	51 (45)
200 (45)	257 (45)	20 (45)	134 (45)	134 (45)	130 (45)	80 (45)	80 (45)	122 (45)	122 (45)	122 (45)
78 (45)	114 (45)	54 (45)	142 (45)	142 (45)	98 (45)	234 (45)	234 (45)	138 (45)	138 (45)	138 (45)
112 (45)	113 (45)	97 (45)	97 (45)	97 (45)	98 (45)	85 (45)	61 (45)	161 (45)	80 (45)	40 (45)
60 (45)	42 (45)	157 (45)	33 (45)	33 (45)	48 (45)	23 (45)	47 (45)	42 (45)	42 (45)	42 (45)
37 (45)	55 (45)	134 (45)	90 (45)	90 (45)	146 (45)	99 (45)	53 (45)	44 (45)	32 (45)	30 (45)
30 (45)	30 (45)	70 (45)	90 (45)	90 (45)	90 (45)	90 (45)	77 (45)	92 (45)	128 (45)	158 (45)
103 (45)	41 (45)	5 (45)	25 (45)	25 (45)	31 (45)	27 (45)	7 (45)	7 (45)	7 (45)	10 (45)
18 (45)										

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 111
 MEAN = 96
 MAX = 455
 MIN = 7

ABSORPTION

2 (45)	5 (45)	2 (45)	1 (45)	3 (45)	3 (45)	4 (45)	7 (45)
---------	---------	---------	---------	---------	---------	---------	---------

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 9
 MEAN = 4
 MAX = 5
 MIN = 1

SCLEROSCOPIC HARDNESS

66 (45) 79 (45) 93 (45) 69 (45) 88 (45) 75 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 6
MEAN = 78
MAX = 93
MIN = 66

IMPACT TOUGHNESS

62 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 1
MEAN = 62
MAX = 62
MIN = 62

MH32916. 03/17/70, PURDUE MACF 02/07/70.

11.40.16.MH329/ 15073,PEPLOFF,T60,CM60000 11.40
11.40.16.000,TP1,P10.
11.40.16.MAP(ON)
11.40.16.RUN(15)
11.51.12.CTIME 002.683 SEC. RUN MOD LEVEL 4R
11.51.13.REQUEST(TAPE1,556,HY,X,C=94.MI,READ)
23.17.49. M153 ASSIGNED = 556
23.17.49.REWIND(TAPE1)
23.17.50.L60(LC=100000)
23.17.53.CX 3.224 SEC.
23.17.53.PX 5.396 SEC.
23.17.53.VL 50100
00.13.30.STOP
00.13.30.CP 52.597 SEC.
00.13.30.PP 358.094 SEC.
00.13.30.LINE1 = 014565.OCTAL
00.13.30.C 5.277 MOD-SEC.

Program No. 5

LEAST SQUARES ROUTINE

for non-blank entries for both variables
in the same data set for selected pairs
of variables

PROGRAM LEAD, INPUT, OUTPUT, INPES=INPU, IAPES=UUU, IAPU=IAPU

```
00003 DATA X/1H /
00003 DIMENSION ACARD(1),RCARD(8),DCARD(8),FCARD(8),HCARD(8)
00003 DIMENSION IDEN(1),COEFF(20)
00003 DIMENSION B15A1(137),A1815(137),B15D2(79),D2815(79),B15F4(71),F481
15(71),A15D9(110),D9R15(110),B15F10(61),F10R15(61),B15F11(59),F11R1
25(59),B15A17(221),A17B15(221),B15A19(484),A19B15(484),B15A20(222),
3A20R15(222),B15H21(28),H21B15(28),TYPE1(137),TYPE2(79),TYPE3(71),T
4YPE4(110),TYPE5(61),TYPE6(59),TYPE7(221),TYPE8(484),TYPE9(22),TYP
5F10(28),S1(137),S2(79),S3(71),S4(110),S5(61),S6(59),S7(221),S8(484
6),S9(222),S10(24)
00003 INTEGER TYPE,TYPE1,TYPE2,TYPE3,TYPE4,TYPE5,TYPE6,TYPE7,TYPE8,TYPE9
1,TYPE10
00003 DATA J1,J2,J3,J4,J5,J6,J7,J8,J9,J10/10*0/
00003 DO 100 K=1,2170
00003 READ(1,400)ACARD
00003 FORMAT(1A10)
00003 DECODE(R0,1,ACARD(1)) S,TYPE,A20,A19,A17,A1
400 FORMAT(7X,A1,7X,13,19X,A4,2X,A4,19X,A3,2X,A2)
00003 1 READ(1,400) BCARD
00003 DECODE(R0,2,BCARD(1)) B15
2 FORMAT(28X,A3)
00003 2 READ(1,3) C
00003 FORMAT(50X,A5)
00003 3 READ(1,400) DCARD
00003 DECODE(R0,24,DCARD(1)) D2,D9
24 FORMAT(54X,A4,A3)
00003 4 READ(1,3) E
00003 DECODE(R0,26,HCARD(1)) H21
25 FORMAT(14X,A3,15X,A4,A3)
00003 5 READ(1,400) HCARD
00003 DECODE(R0,26,HCARD(1)) H21
26 FORMAT(21X,A4)
00003 10 IF((R15,NE,X).AND.(A1,NE,X)) GO TO 50
00003 11 IF((R15,NE,X).AND.(A17,NE,X)) GO TO 51
00003 12 IF((R15,NE,X).AND.(A19,NE,X)) GO TO 52
00003 13 IF((R15,NE,X).AND.(A20,NE,X)) GO TO 53
00003 14 IF((R15,NE,X).AND.(D2,NE,X)) GO TO 54
00003 15 IF((R15,NE,X).AND.(D9,NE,X)) GO TO 55
00003 16 IF((R15,NE,X).AND.(F10,NE,X)) GO TO 56
00003 17 IF((R15,NE,X).AND.(F4,NE,X)) GO TO 57
00003 18 IF((R15,NE,X).AND.(F11,NE,X)) GO TO 58
00003 19 IF((R15,NE,X).AND.(H21,NE,X)) GO TO 59
00003 GO TO 100
00003 J1 = J1 + 1
00003 DECODE(R0,40,BCARD(1)) B15A1(J1)
40 FORMAT(28X,F3,1)
00003 DEC DE(R0,440,ACARD(1)) A1815(J1)
440 FORMAT(70X,F2,0)
00003 TYPE1(J1) = TYPE
00003 S1(J1) = S
00003 GO TO 11
00003 J2 = J2 + 1
00003 DECODE(R0,40,RCARD(1)) B15A17(J2)
00003 DECODE(R0,441,ACARD(1)) A17H15(J2)
441 FORMAT(45X,F3,0)
00003 TYPE2(J2) = TYPE
00003 S2(J2) = S
00003 GO TO 12
00003 J3 = J3 + 1
```

```
000350 DECODE(R0,40,BCARD(1)) B15A19(J3)
000357 DECODE(R0,442,ACARD(1)) A19B15(J3)
442 FORMAT(42X,F4,2)
000367 TYPE3(J3) = TYPE
000371 S3(J3) = S
000373 GO TO 13
000373 J4 = J4 + 1
53 DECODE(R0,40,RCARD(1)) B15A20(J4)
000375 DECODE(R0,443,ACARD(1)) A20B15(J4)
443 FORMAT(36X,F4,2)
000404 TYPE4(J4) = TYPE
000414 S4(J4) = S
000416 GO TO 14
000420 J5 = J5 + 1
54 DECODE(R0,40,RCARD(1)) B15D2(J5)
000422 DECODE(R0,444,DCARD(1)) D2B15(J5)
444 FORMAT(54X,F4,2)
000441 TYPE5(J5) = TYPE
000443 S5(J5) = S
000445 GO TO 15
000445 J6 = J6 + 1
55 DECODE(R0,40,BCARD(1)) B15D9(J6)
000447 DECODE(R0,445,DCARD(1)) D9B15(J6)
445 FORMAT(58X,F3,2)
000466 TYPE6(J6) = TYPE
000470 S6(J6) = S
000472 GO TO 16
000472 J7 = J7 + 1
56 DECODE(R0,40,BCARD(1)) B15F10(J7)
000474 DECODE(R0,446,FCARD(1)) F10B15(J7)
446 FORMAT(14X,F3,2)
000513 TYPE7(J7) = TYPE
000515 S7(J7) = S
000517 GO TO 17
000517 J8 = J8 + 1
57 DECODE(R0,40,BCARD(1)) B15F4(J8)
000521 DECODE(R0,447,FCARD(1)) F4B15(J8)
447 FORMAT(32X,F4,2)
000530 TYPE8(J8) = TYPE
000540 S8(J8) = S
000542 GO TO 18
000544 J9 = J9 + 1
58 DECODE(R0,40,BCARD(1)) B15F11(J9)
000546 DECODE(R0,448,FCARD(1)) F11B15(J9)
448 FORMAT(36X,F3,2)
000565 TYPE9(J9) = TYPE
000565 S9(J9) = S
000567 GO TO 19
000571 J10 = J10 + 1
59 DECODE(R0,40,BCARD(1)) B15H21(J10)
000573 DECODE(R0,449,HCARD(1)) H21B15(J10)
449 FORMAT(21X,F4,2)
000612 TYPE10(J10) = TYPE
000614 S10(J10) = S
000616 CONTINUE
000620 NORDR = 1
000620 NORDR = NORDR + 1
000623 READ(5,20) IDEN
000631 WRITE(6,4) J1
```

```

000637 WRITE(6,5) (S1(I),TYPE1(I),B15A1(I),A1B15(I),I=1,J1)
000640 CALL LSTSQ(J1,NORDR,B15A1,A1B15,COEFF)
000644 WRITE(6,6) IDEN
000672 WRITE(6,7) (COEFF(I),I=1,NRDR)
000705 READ(5,20) IDEN
000713 WRITE(6,4) J2
000721 WRITE(6,5) (S2(I),TYPE2(I),B15A17(I),A17B15(I),I=1,J2)
000742 CALL LSTSQ(J2,NORDR,B15A17,A17B15,COEFF)
000746 WRITE(6,6) IDEN
000754 WRITE(6,7) (COEFF(I),I=1,NRDR)
000767 READ(5,20) IDEN
000775 WRITE(6,4) J3
001003 WRITE(6,5) (S3(I),TYPE3(I),B15A19(I),A19B15(I),I=1,J3)
001024 CALL LSTSQ(J3,NORDR,B15A19,A19B15,COEFF)
001030 WRITE(6,6) IDEN
001036 WRITE(6,7) (COEFF(I),I=1,NRDR)
001051 READ(5,20) IDEN
001057 WRITE(6,4) J4
001065 WRITE(6,5) (S4(I),TYPE4(I),B15A20(I),A20B15(I),I=1,J4)
001106 CALL LSTSQ(J4,NORDR,B15A20,A20B15,COEFF)
001112 WRITE(6,6) IDEN
001120 WRITE(6,7) (COEFF(I),I=1,NRDR)
001133 READ(5,20) IDEN
001141 WRITE(6,4) J5
001147 WRITE(6,5) (S5(I),TYPE5(I),B15D2(I),D2B15(I),I=1,J5)
001170 CALL LSTSQ(J5,NORDR,B15D2,D2B15,COEFF)
001174 WRITE(6,6) IDEN
0012 WRITE(6,7) (COEFF(I),I=1,NRDR)
001215 READ(5,20) IDEN
001223 WRITE(6,4) J6
001231 WRITE(6,5) (S6(I),TYPE6(I),B15D9(I),D9B15(I),I=1,J6)
001252 CALL LSTSQ(J6,NORDR,B15D9,D9B15,COEFF)
001256 WRITE(6,6) IDEN
001264 WRITE(6,7) (COEFF(I),I=1,NRDR)
001277 READ(5,20) IDEN
001305 WRITE(6,4) J7
001313 WRITE(6,5) (S7(I),TYPE7(I),B15F10(I),F10B15(I),I=1,J7)
001334 CALL LSTSQ(J7,NORDR,B15F10,F10B15,COEFF)
001340 WRITE(6,6) IDEN
001346 WRITE(6,7) (COEFF(I),I=1,NRDR)
001361 READ(5,20) IDEN
001367 WRITE(6,4) J8
001375 WRITE(6,5) (S8(I),TYPE8(I),B15F4(I),F4B15(I),I=1,J8)
001416 CALL LSTSQ(J8,NORDR,B15F4,F4B15,COEFF)
001422 WRITE(6,6) IDEN
001430 WRITE(6,7) (COEFF(I),I=1,NRDR)
001443 READ(5,20) IDEN
001451 WRITE(6,4) J9
001457 WRITE(6,5) (S9(I),TYPE9(I),B15F11(I),F11B15(I),I=1,J9)
001500 CALL LSTSQ(J9,NORDR,B15F11,F11B15,COEFF)
001504 WRITE(6,6) IDEN
001512 WRITE(6,7) (COEFF(I),I=1,NRDR)
001525 READ(5,20) IDEN
001533 WRITE(6,4) J10
001541 WRITE(6,5) (S10(I),TYPE10(I),B15H21(I),H21B15(I),I=1,J10)
001542 CALL LSTSQ(J10,NORDR,B15H21,H21B15,COEFF)
001566 WRITE(6,6) IDEN
001574 WRITE(6,7) (COEFF(I),I=1,NRDR)
001607 20 FORMAT(1A4)
001607 4 FORMAT(1H1,///10X,39HSTRAIGHT LINE LEAST SQUARES FIT THROUGH,14,1X
1,12HPPOINTS (X,Y)///)

```

```

001607 5  FORMAT(6(X,A1,1H-,13.1H(,F5.0,1H, ,F5.0,1H)))
001607 6  FORMAT(1H0,18A4)
001607 7  FORMAT(1H0,5X,3HY =,F8.3,3H + ,F8.3,4H * X)
001607   STOP
001611   END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS

020627

UNUSED COMPILER SPACE

022300

000010	SUBROUTINE LSTSQ(NUM,NORDR,X,Y,RETRN)	LSTS0002
000010	INTEGER R,RR	LSTS0003
000010	REAL MATRX(20,20),YSUM(20),RETRN(20)	
	DIMENSION X(500),Y(500),SX(40)	
		LSTS0005
		LSTS0006
		LSTS0007
	SUBROUTINE LSTSQ	LSTS0008
	DECKS USED	LSTS0009
	LSTSQ	LSTS0010
		LSTS0011
	PURPOSE	LSTS0012
	LSTSQ COMPUTES AN N TH-ORDER POLYNOMIAL LEAST SQUARES	LSTS0013
	FIT THROUGH M POINTS.	LSTS0014
		LSTS0015
	USAGF	LSTS0016
	CALL LSTSQ(NUM,NORDR,X,Y,RETRN)	LSTS0017
		LSTS0018
	DESCRIPTION OF PARAMETERS	LSTS0019
	NUM - NUMBER OF POINTS USED.	LSTS0020
	NORDR - ORDER OF THE DESIRED POLYNOMIAL FIT.	LSTS0021
	X - NUM-VECTOR OF X-VALUES.	LSTS0022
	Y - NUM-VECTOR OF CORRESPONDING Y-VALUES.	LSTS0023
	RETRN - (NORDR+1)-VECTOR OF COEFFICIENTS OF DESIRED	LSTS0024
	POLYNOMIAL ORDERED FROM COEFFICIENT OF CONSTANT	LSTS0025
	TERM TO COEFFICIENT OF HIGHEST POWER TERM.	LSTS0026
		LSTS0027
	REMARKS	LSTS0028
	THE CURRENT DIMENSION STATEMENT IS SUFFICIENT FOR NORDR .LE.	LSTS0029
	19 AND NUM .LE. 200. IF THE USER WISHES TO CHANGE THE	LSTS0030
	DIMENSION STATEMENT TO SUIT HIS NEEDS, THE FOLLOWING FORM	LSTS0031
	SHOULD BE USED--	LSTS0032
		LSTS0033
	TYPE STATEMENT	LSTS0034
	REAL MATRX(NORDR+1,NORDR+1),YSUM(NORDR+1),RETRN(NORDR+1)	LSTS0035
	DIMENSION STATEMENT	LSTS0036
	DIMENSION X(NUM),Y(NUM),SX(2*NORDR)	LSTS0037
		LSTS0038
	METHOD	LSTS0039
	A STANDARD LINEAR LEAST-SQUARES FIT IS APPROXIMATED BY	LSTS0040
	CALCULATING THE COEFFICIENTS OF THE NORMAL EQUATIONS AND	LSTS0041
	THEN SOLVING THE NORMAL EQUATIONS USING THE GAUSS-	LSTS0042
	ELIMINATION PROCESS. ORTHOGONAL COEFFICIENTS ARE NOT USED.	LSTS0043
	FOR REFERENCE--	LSTS0044
	MCCRACKEN, D.D. AND DORN, W.S., #NUMERICAL METHODS AND	LSTS0045
	FORTRAN PROGRAMMING#, JOHN WILEY & SONS(NEW YORK--1964),	LSTS0046
	PP. 231-243, PP.262-275.	LSTS0047
	PROGRAMMER--RICHARD F. PUK	LSTS0048
		LSTS0049
		LSTS0050
		LSTS0051

000010	NRDR = NRDR+1	LSTS0052
000012	C ZERO MATRICES	LSTS0053
000013	DO 10 I=1,NRDR	LSTS0054
000015	YSUM(I)=0.	LSTS0055
000016	DO 10 J=1,NRDR	LSTS0056
000027	10 MATRX(I,J)=0.	LSTS0057
000030	RR = 2*NDRD	LSTS0058
000031	DO 11 I=1,RR	LSTS0059
	11 SX(I)=0.	LSTS0060
000035	C CALCULATE SUMS	LSTS0061
000037	DO 20 K=1,RR	LSTS0062
000040	DO 20 I=1,NUM	LSTS0063
000052	20 SX(K)=SX(K)+X(I)**K	LSTS0064
000054	DO 25 K=1,NDRD	LSTS0065
000055	DO 25 I=1,NUM	LSTS0066
000071	25 YSUM(K+1) = YSUM(K+1)+(X(I)**K)*Y(I)	LSTS0067
000072	DO 26 I=1,NUM	LSTS0068
	26 YSUM(I) = YSUM(I)+Y(I)	LSTS0069
000076	C INSERT IN MATRIX (CHANGE FOR DIFFERENT NORMAL EQUATIONS)	LSTS0070
000077	DO 30 K=1,NRDR	LSTS0071
000100	DO 30 I=1,NRDR	LSTS0072
000102	IF(I+K-2)27,28,27	LSTS0073
000105	27 INSRT = I+K-2	LSTS0074
000112	MATRX(K,I) = SX(INSRT)	LSTS0075
000112	GO TO 30	LSTS0076
000117	28 MATRX(K,I) = NUM	LSTS0077
	30 CONTINUE	LSTS0078
000124	C SOLVE NORMAL EQUATIONS(MATRIX)	LSTS0079
000125	C GAUSS FLIMINATION SFEQUENCE	LSTS0080
000126	LITE=2	LSTS0081
000130	DO 10 J=1,NRDR	LSTS0082
000132	J1=J+1	LSTS0083
000133	IF(J1-NRDR)6,6,5	LSTS0084
000133	5 LITE=1	LSTS0085
000135	6 CONTINUE	LSTS0086
000137	L=J	LSTS0087
000145	DO 12 I=J1,NRDR	LSTS0088
000146	GO TO (12,17),LITE	LSTS0089
000146	17 IF(ABS(MATRX(I,J))-ABS(MATRX(L,J)))12,12,111	LSTS0090
000146	111 L=I	LSTS0091
000146	12 CONTINUE	LSTS0092
000147	IF(L-J)13,14,13	LSTS0093
000171	13 DO 15 K=J,NRDR	LSTS0094
000175	SWIT=MATRX(J,K)	LSTS0095
000203	MATRX(J,K)=MATRX(L,K)	LSTS0096
000211	16 MATRX(L,K)=SWIT	LSTS0097
000213	SWIT=YSUM(J)	LSTS0098
000216	YSUM(J)=YSUM(L)	LSTS0099
000220	YSUM(L)=SWIT	LSTS0100
000220	14 CONTINUE	LSTS0101
000222	DO 10 I=J1,NRDR	LSTS0102
000230	GO TO(110,23),LITE	LSTS0103
000237	23 EM=MATRX(I,J)/MATRX(J,J)	LSTS0104
000241	MATRX(I,J)=0.	LSTS0105
000243	DO 15 K=1,NRDR	LSTS0106
000250	C=EM*MATRX(J,K)	LSTS0107
000255	MATRX(I,K)=MATRX(I,K)-C	LSTS0108
000257	15 CONTINUE	LSTS0109
000271	110 YSUM(I)=YSUM(I)-EM*YSUM(J)	LSTS0110
000277	RETRN(NRDR)=YSUM(NRDR)/MATRX(NRDR,NRDR)	LSTS0111
000301	IN=NRDR+1	LSTS0112
000302	DO 4) NIX=2,NRDR	LSTS0113
000304	MIX=IN-NIX	LSTS0114
000306	MIN=MIX+1	LSTS0115
	SUMA=0.	LSTS0116

INPUTS	24453	FILE - LGD	ADDRESS	LEAST	14	16	20	22	24	26	30	32
					33	44	46	50	51	70	72	74
					76	77	116	120	122	124	126	127
					146	150	152	153	277	301	303	304
					307	311	313	314	324	326	335	336
					334	336	340	341	351	353	355	356
					361	363	365	366	376	400	402	403
					406	410	412	413	423	425	427	430

LOAD MAP ENTRY ADDRESS 10-23-20. 03/15/70. PAGE 2

REFERENCES (RELATIVE)

OUTPUTC	24535	LEAST	433	435	437	440	450	452	454	455
			460	462	464	465	475	477	501	502
			505	507	511	512	522	524	526	527
			532	534	536	537	547	551	553	554
			557	561	563	564	574	576	600	601
			504	606	610	611				
			633	635	636	641	645	647	651	653
			657	666	670	671	674	700	704	715
			717	720	723	727	731	733	735	741
			750	752	753	756	762	766	777	1001
			1002	1005	1011	1013	1015	1017	1023	1032
			1034	1035	1040	1044	1050	1061	1063	1064
			1067	1072	1075	1077	1101	1105	1114	1116
			1117	1122	1126	1132	1143	1145	1146	1151
			1155	1157	1161	1163	1167	1176	1200	1201
			1204	1210	1214	1225	1227	1230	1233	1237
			1241	1243	1245	1251	1260	1262	1263	1266
			1272	1276	1307	1311	1312	1315	1321	1323
			1325	1327	1333	1342	1344	1345	1350	1354
			1360	1371	1373	1374	1377	1403	1405	1407
			1411	1415	1424	1426	1427	1432	1436	1442
			1453	1455	1456	1461	1465	1467	1471	1473
			1477	1506	1510	1511	1514	1520	1524	1535
			1537	1540	1543	1547	1551	1553	1555	1561
			1570	1572	1573	1576	1602	1606		

KODER
RBAIEX
FATAL78
CARD.F
RKSPRU.
FIZBAK.
POSFIL.
ROPRU.
DAT.
CLOI.
OPFN.

57

LSTSQ
SYSTEM
SIOs

32

OUTPUTC

72

OUTPUTC

22

INPUTC

506

SYSTEM

22

OUTPUTC

52

INPUTC

71

OUTPUTC

410

SYSTEM

INPUTC
INPUTS
INPUTC
REFERENCES (RELATIVE)

24
33

47
42

45

UNSATISFIED EXTERNALS

** NONF **

STRAIGHT LINE LEAST SQUARES FIT THROUGH 137 POINTS (X,Y)

A- 7(18, 29)	A- 7(15, 15)	A- 7(13, 9)	A- 7(6, 3)	A- 13(19, 37)	A- 13(16, 17)
A- 13(16, 16)	A- 13(17, 23)	A- 13(17, 19)	A- 13(17, 24)	A- 13(18, 21)	A- 13(16, 9)
A- 14(14, 11)	A- 14(18, 30)	A- 14(18, 30)	A- 14(18, 23)	A- 14(16, 16)	A- 14(16, 11)
A- 21(16, 19)	A- 21(18, 22)	A- 22(8, 17)	A- 22(17, 25)	A- 22(11, 19)	A- 22(12, 24)
A- 22(15, 26)	A- 22(14, 32)	A- 39(16, 26)	A- 47(7, 13)	A- 47(20, 14)	A- 47(8, 21)
A- 47(19, 4)	A- 47(19, 10)	A- 50(11, 6)	A- 50(11, 12)	A- 52(14, 21)	A- 52(16, 22)
A- 59(16, 11)	A- 64(17, 18)	A- 218(18, 14)	A- 218(17, 11)	A- 218(17, 13)	A- 218(17, 14)
A- 218(9, 3)	A- 218(14, 7)	A- 218(11, 6)	A- 218(14, 8)	A- 218(10, 4)	A- 223(12, 31)
A- 223(15, 10)	A- 223(16, 7)	A- 223(16, 8)	A- 223(14, 7)	A- 223(17, 9)	A- 223(8, 10)
A- 223(16, 15)	A- 223(16, 9)	A- 223(16, 10)	A- 223(11, 3)	A- 223(14, 4)	A- 223(14, 4)
A- 223(16, 7)	A- 223(17, 9)	A- 223(18, 8)	A- 223(12, 7)	A- 223(19, 10)	A- 223(13, 8)
A- 223(16, 9)	A- 225(11, 10)	A- 225(8, 10)	A- 225(10, 7)	A- 225(9, 9)	A- 225(15, 12)
A- 225(10, 13)	A- 225(13, 7)	A- 225(11, 12)	A- 225(12, 12)	A- 236(6, 2)	A- 236(6, 1)
A- 236(10, 5)	A- 236(12, 7)	A- 236(10, 6)	A- 236(10, 7)	A- 236(13, 8)	A- 236(15, 10)
A- 236(6, 4)	A- 236(7, 3)	A- 236(10, 6)	A- 237(14, 4)	A- 237(16, 14)	A- 237(15, 7)
A- 236(13, 13)	A- 238(14, 11)	A- 238(16, 18)	A- 257(18, 52)	A- 400(19, 40)	A- 404(9, 12)
A- 404(14, 22)	A- 404(11, 11)	A- 404(13, 17)	A- 404(15, 10)	A- 404(13, 27)	A- 404(15, 19)
A- 404(14, 18)	A- 404(15, 11)	A- 404(15, 12)	A- 404(13, 15)	A- 404(18, 16)	A- 404(16, 17)
A- 407(19, 20)	A- 407(16, 6)	A- 407(17, 14)	A- 407(17, 7)	A- 407(17, 18)	A- 407(19, 38)
A- 410(14, 8)	A- 410(14, 7)	A- 410(19, 7)	A- 412(16, 8)	A- 413(18, 39)	A- 416(17, 6)
A- 417(17, 7)	A- 419(20, 25)	A- 419(15, 12)	A- 422(17, 7)	A- 423(18, 8)	A- 440(17, 40)
A- 440(17, 40)	A- 622(18, 20)	A- 622(19, 11)	A- 622(17, 48)	A- 622(21, 24)	A- 622(14, 5)
A- 622(14, 5)	A- 629(9, 3)	A- 629(9, 4)	A- 629(16, 9)	A- 629(17, 10)	

(R15/X VERSUS A1/Y)

$$Y = -5.121 \cdot X + 1.302 \cdot X$$

MF14330. 03/15/70.PURDUE MACE 02/07/70.

09.40.10.MF143/	15073,PERI.OFF,T60,TP1,CM65000;
09.40.10.P10.	
09.40.10.MAP(ON)	
09.40.10.RUN(S)	
09.40.16.CTIME 002.814 SEC. RUN MOD LEVEL 48	
09.40.17.REQUEST(TAPE1,556,HY,X,C=84,MI,READ)	
10.23.15. M751 ASSIGNED - 556	
10.23.15.REWIND(TAPE1)	
10.23.18.160.	
10.23.20.CX	3.363 SFC.
10.23.20.PX	5.341 SFC.
10.23.20.NL	27600
10.37.16.STOP	
10.37.16.CP	62.454 SFC.
10.37.16.PP	339.266 SEC.
10.37.16.LINES	1376 OCTAL
10.37.16.CM	1.321 MWD-SEC.

HISTOGRAM ROUTINE

for non-blank entries for both variables
in the same data set

PROGRAM CH2(INPUT,OUTPUT,TAPES=INPUT,TAPES=J(20),TAPES=J(20),TAPES=J(20))

000003
000003
000003
000004
000006
000013
000021
000027
000035
000043
000051
000057
000065
000065
000065
000075
000075
000077
000101
000102
000104
000117
000132
000132
000134

DATA 4L/1M /
DIMENSION H(15(49),D(26(49)
N = 0
DO 100 K=1,2170
H(1,1) = A
H(1,1) = H14
H(1,1) = C
H(1,1) = D25
H(1,1) = F
H(1,1) = G
H(1,1) = H
1 FORMAT(2X,A3)
2 FORMAT(61X,A4)
3 FORMAT(50X,A5)
IF((H(1,1).NE.C1).AND.(D(26,NE.H1)) GO TO 10
GO TO 100
N = N + 1
H(1,1) = H15
H(26,1) = D25
100 CONTINUE
WRITE(8,60) (H(15(K),K=1,N)
50 FORMAT(2X,A3)
WRITE(8,50) (D(26(K),K=1,N)
50 FORMAT(2X,A4)
END FILE B
END

PROGRAM LENGTH INCLUDING I/O BUFFERS
004465

UNUSED COMPILER SPACE
027200

51

0000801
(2X,F3.1)
SELECT11000010.001

HISTOGRAM FOR PROPAGATION VELOCITY, LONGITUDINAL WAVE/LAB

0000801
PRGSLM 2JUD001000490010000
(2X,F4.2)
SELECT11000010.001

HISTOGRAM FOR YOUNG/S MODULUS, LAB STATIC (1ST LOADING)

FINISH
6789

N.B. This insert indicates the deck set-up as submitted to the computer. These data cards for the BMD routine are not included in the output listing.

LOG MAP FILE - 150

FOR LOAN 100 1-10 LOAD 11-15 FOR LOAN 16-20 FOR TABLE 21-25

18.22.55. 12/19/60. PAGE 1

JOHN STONE PAGE 2114

Continued on following page

CM2	100	1.50
ENDFIL	4605	SYSTEM
INDFIL	4660	SYSTEM
INDFIL	5001	SYSTEM
INDFIL	7452	SYSTEM
INDFIL	11412	SYSTEM
INDFIL	11530	SYSTEM
INDFIL	11046	SYSTEM

REFERENCES (RELATIVE)

CH2	101	CH2	133	11	15	17	20	23	28
ENDFIL	4506	CH2	7	12	15	17	20	23	28
INALIC	4662	CH2	24	33	34	37	41	42	48
			47	53	55	56	61	63	68

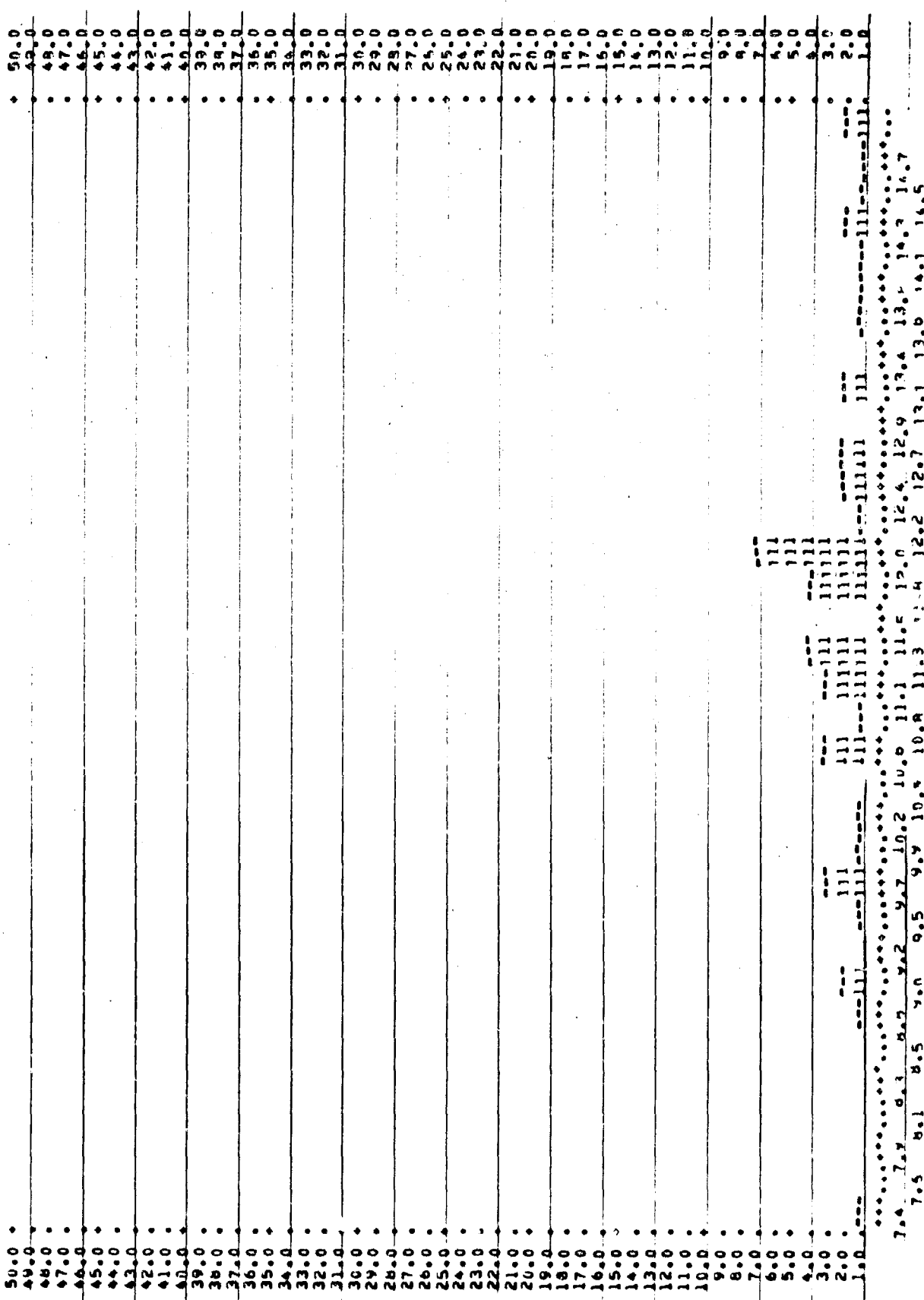
UNSATISFIED STATEMENTS
OF THE

HISTOGRAM FOR PROPAGATION VELOCITY, LONGITUDINAL WAVE/LA8
 THE VALUE GIVEN FOR THE INTERVAL WIDTH IS TOO SMALL.
 A NEW VALUE, .22%, HAS BEEN SUBSTITUTED.

MIN = 7.349994 MAX = 15.200000

7.4 7.9 8.3 8.8 9.2 9.7 10.2 10.6 11.1 11.5 12.0 12.4 12.9 13.4 13.9 14.3 14.7
 7.5 8.1 8.5 9.0 9.5 9.9 10.4 10.8 11.3 11.8 12.2 12.7 13.1 13.6 14.1 14.5

TOP LEFT HAND SCALE IS 50.0



9M0050 GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 04 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCL

PROBLEM CODE . . . 21000
NO. OF VARIABLES . . . 1
NO. OF CASES . . . 49
NO. OF SELECTION CARDS . . . 1
NO. OF VARIABLES READ . . . 0
NO. OF INDEX CARDS . . . 0
NO. OF FORMAT CARDS . . . 1

9M0050 GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 04 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCL

HISTOGRAM OF VARIABLE 1

3.4500	1.6800	2.5000	4.5100	4.3600	3.4700	2.7400	2.5600	3.2700	4.0000
3.5000	3.5500	2.5000	2.7400	2.4000	3.6000	4.1300	4.1000	4.1000	4.5400
4.5400	4.5400	2.4600	2.4000	2.4600	3.6300	3.6300	3.0300	4.9600	4.9400
6.9600	6.8000	0.8000	4.8000	2.5000	6.0400	6.9400	4.9400	5.3800	5.3800
5.3800	5.4400	2.4700	2.4600	1.1400	4.7100	7.7400	1.1400	.7200	

Minister of Agriculture

66667E

MIN 2.3

2. 2
MAY 1964

TOP LEFT HAND SCALE IS 50.0

[illegible]

3	2	1	0	2.1	2.5	2.9	3.4	3.8	4.2	4.7	5.1	5.6	6.0	6.4	6.9	7.3
5	1.0	1.4	1.9	2.3	2.7	3.2	3.6	4.0	4.5	4.9	5.4	5.8	6.2	6.7	7.1	

MM53941. 12/19/69. PURDUE PAGE 11/27/59.

09.29.55. MM539/ 3512, NAHAS, T60, CM65000, TP1, P10
09.29.55..
09.29.55. MAP (ON)
09.29.55. RIV(S)
09.29.57. CTIME 000.208 SEC. RUN MOD LEVEL 4B
09.29.58. REQUEST (TAPF1, 556, NY, X, C, R4, MT, READ)
18.22.52. MISO ASSIGNED = 55A
18.22.52. REWIND (TAPF1)
18.22.53. LG7.
18.22.55. CX 460 SEC.
18.22.55. PX 3.496 SEC.
18.22.55. NL 12000
18.36.15. END CM2
18.36.15. REWIND (TAPF1)
18.36.15. RETURN (TAPF1)
18.36.15. TAPE LIMITED
18.36.15. REWIND (END)
18.36.15. RFL (55000)
18.36.15. CX 20.124 SEC.
18.36.15. PX 341.041 SEC.
18.36.15. NL 65000
18.56.37. REWIND (TAPF4)
18.56.37. LINCOPY (STATSIN, LG0, RMDSN)
18.59.22. LG7.
18.59.27. CX 20.940 SEC.
18.59.27. PX 346.319 SEC.
18.59.27. NL 56300
18.59.30. STOP
18.59.30. CX 21.907 SEC.
18.59.30. PX 347.309 SEC.
18.59.30. LINES = 0430 OCTAL
18.59.30. CM 2.449 120-SEC.

NOT REPRODUCIBLE

BMD PROGRAM TO PRODUCE HISTOGRAM

SUBPROGRAM LENGTH
000052

TPWD2

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

10	-	000017	12	-	000010	14	-	000012	15	-	000016
16	-	000014	18	-	000022	19	-	000020	22	-	000024
24	-	000026	28	-	000027	40	-	000030	49	-	000043

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS

000040

START OF TEMPORARIES

000050

START OF INDIRECTS

000052

UNUSED COMPILE SPACE

022400

	PROGRAM BM050 (INPUT=1,OUTPUT=1,TAPES=INPUT,TAPES=OUTPUT,TAPES)	BM05000001
000000	GENERAL PLOT WITH HISTOGRAM AUGUST 18, 1964	BM05000002
C	HEALTH SCIENCES COMPUTING FACILITY, UCLA	BM05000003
C	A400 CONVERSION BETTY BENSON	BM05000004
000003	DIMENSION FG(999),X(15000),SYM(15),NX(15),HEAD(57),YMT(500),XMA(50	BM05000005
	10),Z(15),NXX(15),XY(51,17)	BM05000006
C		BM05000007
000003	COMMON X,HEAD,XY,SYM,Z	BM05000008
000003	EQUIVALENCE (FG,XMT)	BM05000009
000003	TYPE INTEGER A123,B123,C123,D123,MODE,XY	BM05000010
C		BM05000011
000003	110 FORMAT(70HBM050 GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06	BM05000012
	1 AUG. 18, 1964 /	BM05000013
	10H HEALTH SCIENCES COMPUTING FACILITY,UCLA//)	BM05000014
C		BM05000015
000003	A123=(+6MFINISH)	BM05000016
000005	B123=(+6HPRUHLN)	BM05000017
000006	C123=(+6HCHSVAR)	BM05000018
000010	D123=(+6HSELECT)	BM05000019
000011	5 READ(5,101)TODE,SAME,VV,NP,NG,NAD,NTHAN,MTAPE,NCAHU	BM05000020
000037	204 FORMAT(45HCONTROL CARDS INCORRECTLY ORDERED OR PINCHED)	BM05000021
000037	IF (TODE-A123) 200,201,200	BM05000022
000042	202 WRITE(6,204)	BM05000023
000046	201 STOP	BM05000024
000050	200 IF (TODE-B123) 202,203,202	BM05000025
000052	203 NTAPE=MTAPE	BM05000026
000054	IF (NTAPE.EQ.0) NTAPE=5	BM05000027
000056	306 IF (NV*(NV-501))309,204,202	BM05000028
000062	309 IF ((NP-1)*(NP-1500))205,202,202	BM05000029
000070	205 IF ((VV+NAD)*(NP-1500)) 206,200,202	BM05000030
000075	206 CALL VCHECK(NCAHU)	BM05000031

```

000077 207 *H1IF( 6,110)
000103 *H1IF( 6,210) SAME,NV, *P,NG,NAUD,NHAI,NCARD
000125 71 NTOT=NP*NV-NP
000130 IF((NV+NAUD)*(NV+NAUD-501)/1,202,202
000136 1 NCARD=NCARD)*8
000140 HEAD( 5,102)(FG(1),I=1,NCARD)
000152 NCARD=NTOT+NP
000154 DO 211 I=1,NCARD
000156 211 X(I)=0.0
000162 70 DO 3 I=1,NP
C ***** READ IN THE RAW DATA AND TRANSPUSE THE MATRIX
000164 HEAD( NTAPE+FG)(AMA(J), J=1,NV)
000177 DO 3 J=1,NV
000201 K=NP*J-NP+1
000204 3 X(K)=XMA(J)
000212 IF(NTRAN) 202,22,21
000214 21 CALL TRANS(NP,NV,NTRAN)
000217 IF(NV) 202,202,22
000221 22 NPV=NP
000223 IF (NAUD) 999,999,998
000224 998 NV = NV + NAUD
000226 999 K = 1
000227 DO 63 I=1,NV
000231 XMI(I)=99999999.
000233 XMA(I)=-99999999.
000235 DO 64 J=K,NPV
000237 XMI(I)=AMINI(X(J),XMI(I))
000244 64 XMA(I)=AMAXI(X(J),XMA(I))
000253 K=K+NP
000255 63 NPV=NPV+NP
000260 DO 50 JJ=1,NG
000261 HEAD( 5,104)TODE,NH,NL,NC,NV,FN
000300 IF (TODE=0123) 202,209,202
000302 209 IF(NH*(NH-4)/215,202,202
000306 215 NH=NH*8
000310 HEAD( 5,102)(HEAD(1),I=1,NH)
000322 IF(NC) 202,20,8
000324 * NNC=(NC+6)/1
000330 IF(NNC-2)9,9,202
000332 9 NG2=0
000333 DO 150 I=1,NNC
000335 NG1=NG2+1
000337 NG2=NG2+7
000340 HEAD( 5,105)TODE,(NA(J),SYM(J),J=NG1,NG2)
000357 IF (TODE=C123) 202,150,202
000361 150 CONTINUE
C
C RATTLE COMMENT
C FILL COMPUTER WORD WITH INPUT SYMBOL
C
000364 IF(NC.GE.1) CALL FILL(SYM, NC)
C
000370 XMAA=-99999999.
000372 XMIN=99999999.
000373 IF(NC=1)20,11,12
000376 11 J=XMA(1)
000400 XMAX=XMA(J)
000402 XMIN=XMI(J)
000403 GO TO 14
000404 12 DO 13 I=1,NC
000406 J=XMA(I)
000410 XMAX=AMAXI(XMAX,XMA(J))

```

```

QM05000032
QM05000033
QM05000034
QM05000035
QM05000036
QM05000037
QM05000038
QM05000039
QM05000040
QM05000041
QM05000042
QM05000043
QM05000044
QM05000045
QM05000046
QM05000047
QM05000048
QM05000049
QM05000050
QM05000051
QM05000052
QM05000053
QM05000054
QM05000055
QM05000056
QM05000057
QM05000058
QM05000059
QM05000060
QM05000061
QM05000062
QM05000063
QM05000064
QM05000065
QM05000066
QM05000067
QM05000068
QM05000069
QM05000070
QM05000071
QM05000072
QM05000073
QM05000074
QM05000075
QM05000076
QM05000077
QM05000078
QM05000079
QM05000080
QM05000081
QM05000082
QM05000083
QM05000084
QM05000085
QM05000086
QM05000087
QM05000088
QM05000089
QM05000090
QM05000091
QM05000092
QM05000093

```

000414	13	XM1N= AM1N1(XM1N,XM1(J))		QMD500000
000423	14	NPV=0		QMD5000095
000424	10	DO 65 I=1,NC		QMD5000096
000426	05	NXX(I)=NX(I)*NP-NP		QMD5000097
000434		YYY=NY*NP-NP		QMD5000098
000436		IF(NL)23,23,24		QMD5000099
000440	24	WRITE(6,110)		QMD5000100
000444		IF(9-NC)242,249,249		QMD5000101
000447	242	WRITE(6,108)NY*(NX(I)+1=1,9)		QMD5000102
000463		WRITE(6,112)		QMD5000103
000467		WRITE(6,111)(NX(I),I=10,NC)		QMD5000104
000502		GO TO 250		QMD5000105
000503	249	WRITE(6,108)NY*(NX(I)+1=1,NC)		QMD5000106
000520	250	WRITE(6,112)		QMD5000107
000524		DO 26 I=1,NP		QMD5000108
000526		MY=YYY+I		QMD5000109
000530		Y=X(MY)		QMD5000110
000532		DO 25 J=1,NC		QMD5000111
000533		MX=NXX(J)+I		QMD5000112
000535	25	Z(J)=X(MX)		QMD5000113
000541	26	WRITE(6,106)Y,(Z(K),K=1,NC)		QMD5000114
	C 23	WRITE OUTPUT TAPE 6,110	(ORIGINAL)	QMD5000115
000561		WRITE(6,7777)		QMD5000116
000564	23	WRITE(6,103)(HEAD(I),I=1,NH)		QMD5000117
000577		NNP=FN		QMD5000118
000601		YMAX=XMA(NY)		QMD5000119
000603		YM1N=XMI(NY)		QMD5000120
000605		DO 16 I=1,NP		QMD5000121
000606		MY=YYY+I		QMD5000122
000610		Y=X(MY)		QMD5000123
000612		DO 15 J=1,NC		QMD5000124
000613		MX=NXX(J)+I		QMD5000125
000615	15	Z(J)=X(MX)		QMD5000126
000621	16	CALL PLOT(Y,YM1N,YMAX,XM1N,XMAX,NC,NNP)		QMD5000127
000633		IF(NNP)31,32,32		QMD5000128
000634	31	NC=-1		QMD5000129
000635		GO TO 33		QMD5000130
000636	32	NC=0		QMD5000131
000637	33	CALL PLOT(Y,YM1N,YMAX,XM1N,XMAX,NC,NNP)		QMD5000132
000646		GO TO 50		QMD5000133
000647	20	NYT=NY*NP		QMD5000134
000652		YYY=NYT-NP+1		QMD5000135
000654		IF(NL)29,29,28		QMD5000136
000655	28	WRITE(6,110)		QMD5000137
000661		NNC=(NP+9)/10		QMD5000138
000665		NG2=YYY-1		QMD5000139
000667		WRITE(6,107)NY		QMD5000140
000675		DO 285 I=1,NNC		QMD5000141
000677		NG1=NG2+1		QMD5000142
000701		NG2=NG2+10		QMD5000143
000702		IF(NYT=NG2)283,284,284		QMD5000144
000704	283	NG2=NYT		QMD5000145
000706	284	WRITE(6,125)(X(J),J=NG1,NG2)		QMD5000146
000721	285	CONTINUE		QMD5000147
	C 29	WRITE OUTPUT TAPE 6,110	(ORIGINAL)	QMD5000148
000724		WRITE(6,7777)		QMD5000149
000727	29	WRITE(6,103)(HEAD(I),I=1,NH)		QMD5000150
000742		XMAX=XMA(NY)		QMD5000151
000744		XM1N=XMI(NY)-.0000005		QMD5000152
000746		IF((XMAX-XM1N)/FN-34.)34,34,35		QMD5000153
000753	35	FN=(XMAX-XM1N)/34.		QMD5000154
000756		WRITE(6,109)FN		QMD5000155
000764		GO TO 34		QMD5000156

```

000765 34 CALL HIST(NYY, NYT, XMIN, XMAX, FN, IP)
000771 50 CONTINUE
000774 GO TO 5

C
000774 101 FORMAT(2A6, I3, I5, I3, I4, 3RX, I3, 2I2)
000774 102 FORMAT (8A10)
000774 103 FORMAT (1H0, 20X, HA10)
000774 7777 FORMAT (1H1)
000774 104 FORMAT(A6, 2I1, I2, I3, F11.0)

C
C BATTLE COMMENT
C FORMAT 105 CHANGED FROM (7(A6, I3), 46)
C IN ORDER TO COMPLIMENT ADDED BATTLE SUBROUTINE CALLED FILL.
000774 105 FORMAT(A6, / (I3, R), 5X))
000774 106 FORMAT(1H 10(F10.4, 1X))
000774 107 FORMAT(1H 23H HISTOGRAM OF VARIABLE 13//)
000774 108 FORMAT(14H BASE VARIABLE, 3X, 16H CROSS VARIABLES/6X, 10(I3, RX))
000774 109 FORMAT(1H, 54H THE VALUE GIVEN FOR THE INTERVAL WIDTH IS TOO SMALL
1, /13H A NEW VALUE, F11.4, 22H, HAS BEEN SUBSTITUTED, //)
000774 111 FORMAT(5X, 5(I3, HX))
000774 112 FORMAT(1H )
000774 125 FORMAT(1H 10F11.4)
000774 210 FORMAT(14H PROBLEM CODE 3(2H ) 1X A6, /18H NO. OF VARIABLES 3(2H ),
A13, /14H NO. OF CASES 4(2H, ) 15, /24H NO. OF SELECTION CARDS 13, /24H
A NO. OF VARIABLES ADDED 13, /22H NO. OF TRNGEN CARDS 2H, 13, /22H N
NO. OF FORMAT CARDS 2H, 13, //)

C
000774 END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS
005544

BM050

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	-	000136	5	-	000011	8	-	000324	9	-	000332
10	-	000424	11	-	000376	12	-	000404	14	-	000423
20	-	000647	21	-	000214	22	-	000221	23	-	000564
24	-	000440	28	-	000655	29	-	000727	31	-	000634
32	-	000636	33	-	000637	34	-	000765	35	-	000753
50	-	000771	65	-	000427	70	-	000162	71	-	000125
101	-	001061	102	-	001065	103	-	001067	104	-	001074
105	-	001100	106	-	001104	107	-	001107	108	-	001114
109	-	001123	110	-	001007	111	-	001140	112	-	001143
125	-	001145	150	-	000361	200	-	000050	201	-	000046
202	-	000042	203	-	000052	204	-	001043	205	-	000070
206	-	000075	207	-	000077	209	-	000302	210	-	001150
211	-	000157	215	-	000306	242	-	000447	249	-	000503
250	-	000520	283	-	000704	284	-	000706	304	-	000056
309	-	000062	998	-	000274	999	-	000226	7777	-	001072

BLOCK NAMES AND LENGTHS

- 037122

VARIABLE ASSIGNMENTS

A123	=	004207	B123	=	004210	C123	=	004211	D123	=	004212
F8	=	001216	FN	=	004237	HEAD	=	035230C01	I	=	004226
J	=	004227	JJ	=	004232	K	=	004230	MTAPE	=	004222
MX	=	004250	MY	=	004246	NAID	=	004220	NC	=	004235
NCARD	=	004223	NG	=	004217	NG1	=	004242	NSD	=	004241
NH	=	004233	NL	=	004234	NMC	=	004240	NPD	=	004251
NP	=	004216	NPV	=	004231	NTAPE	=	004224	NTPI	=	004225
NTRAN	=	004221	NV	=	004215	NA	=	003165	NXX	=	004170
NY	=	004236	NY1	=	004254	NY2	=	004245	SAVE	=	004214
SYM	=	037064C01	TOUE	=	004213	X	=	000000C01	XMA	=	003204
XMAX	=	004243	X4I	=	001216	XMIN	=	004244	XY	=	035321C01
Y	=	004247	YMAX	=	004252	YMIN	=	004253	Z	=	037103C01

START OF CONSTANTS

000776

START OF TEMPORARIES

001200

START OF INDIRECTS

001212

UNUSED COMPILER SPACE

017200

```

SUBROUTINE HIST(NYY, NYT, XMIN, XMAX, SYMB, NP)
CHIST SUBROUTINE HIST FOR BMD05D JULY 17, 1964
000011 DIMENSION XY(51,17), X(15000), INT(35), XM(3), n(3), SYM(15),
      1  AZ(15), BONE(3), CLAB(57)
C
000011 COMMON X, CLAB, XY, SYM, Z
000011 TYPE INTEGER XM, BONE, U, XY, W
000011 23 FORMAT(1H F5.1, 1X, A1, 17A6, A1, 1X, F5.1)
000011 101 FORMAT(1H 5X, 16(F4.1, 2X), F4.1/9X, 16(F4.1, 2X)/8X, 17(2H+++.))
000011 102 FORMAT(8X, 17(6H+++.)/6X, 16(F4.1, 2X), F4.1/9X, 16(F4.1, 2X))
000011 4000 FORMAT( 8H MIN = , F12.6, 8UX, 7H MAX = , F12.6)
C
000011 C BATTLE COMMENT
000011 OCTAL MASKS INCREASED FOR 0400 WORD SIZE.
000012 XM(1) = 00000077777000000000
000012 XM(2) = 77777700000000000000
000014 BONE(1) = (+6H111000).AND.XM(2)
000016 BONE(2) = (+6H000111).AND.XM(1)
000020 U(1) = (+6H---000).AND.XM(2)
000022 U(2) = (+6H000---).AND.XM(1)
000024 M=1
000025 WRITE( 6, 4000) XM(1), XMAX
000034 DO 50 I=1, 35
000041 50 INT(I)=0
000044 DO 100 K=1, 17
000046 DO 100 J=1, 50
C
000047 C BATTLE COMMENT
000057 C HOLLERITH BLANKS USED INSTEAD OF 0060 OCTAL BLANKS.
000062 100 XY(J,K) = (+6H )
000062 MINV=XMIN/SYMB
000064 TXMIN=XMIN/SYMB-1.0
000064 CLAB(1)=XMIN
000065 DO 16 I=2, 34
000066 16 CLAB(I)=CLAB(I-1)+SYMB
000073 WRITE( 6, 101) (CLAB(I), I=1, 34, 2), (CLAB(J), J=2, 33, 2)

```

```

000112      DO 1 I=NYX, NYT
000117      K=X(I)/SYMB-TAMIN
000123      INT(K)=INT(K)+1
000127      IF (INT(M)-INT(K)) 8, 1, 1
000133      8 M=K
000135      1 CONTINUE
000140      YMAX=INT(M)
000142      SC=50.0
000144      32 IF (YMAX-SC) 30, 30, 31
000147      31 SC=SC+50.0
000151      GO TO 32

C
000152      30 WRITE( 6, 103) SC
000160      103 FORMAT(23H TOP LEFT HAND SCALE IS 50.1)
000160      SC=50.0/SC
000162      15 DO 6 I=1, 34
000167      XL=INT(I)
000171      L=XL*SC+.5
000175      IF (L) 5, 6, 5
000176      5 MH=(3*I-1)/6+1
000203      IT=MOD(I, 2)
000207      IF (IT) 52, 59, 62
000210      59 IT=2
000211      62 XY(L, MB)=(XY(L, MB).AND.XM(IT)).OR.U(IT)
000221      L=L-1
000222      IF (L) 11, 6, 11
000223      11 DO 10 K=1, L
000225      10 XY(K, MB)=(XY(K, MB).AND.XM(IT)).OR.BONE(IT)
000244      6 CONTINUE
000246      DO 7 K=1, 50
000250      L=51-K
000252      M=L
000253      W=R/SC
000255      I=MOD(K, 5)
000261      IF (I-1) 2, 3, 2
000263      3 W=(+1)H+
000265      GO TO 7
000265      2 W=(+1)H+
000267      7 WRITE( 6, 23) W, W, (XY(L, M), M=1, 17), W, R
000321      WRITE( 6, 102) (CLAB(I), I=1, 34, 2), (CLAB(J), J=2, 33, 2)
000340      RETURN
000341      END

```

```

QM05000218
QM05000219
QM05000220
QM05000221
QM05000222
QM05000223
QM05000224
QM05000225
QM05000226
QM05000227
QM05000228
QM05000229
QM05000230
QM05000231
QM05000232
QM05000233
QM05000234
QM05000235
QM05000236
QM05000237
QM05000238
QM05000239
QM05000240
QM05000241
QM05000242
QM05000243
QM05000244
QM05000245
QM05000246
QM05000247
QM05000248
QM05000249
QM05000250
QM05000251
QM05000252
QM05000253
QM05000254
QM05000255
QM05000256
QM05000257
QM05000258
QM05000259

```

SUBPROGRAM LENGTH
000536

HIST

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	-	000135	2	-	000265	3	-	000263	5	-	000176
6	-	000244	7	-	000267	8	-	000133	10	-	000225
11	-	000223	15	-	000162	16	-	000066	23	-	000344
30	-	000152	31	-	000147	32	-	000144	50	-	000041
59	-	000210	62	-	000211	100	-	000047	101	-	000351
102	-	000360	103	-	000414	4000	-	000367			

BLOCK NAMES AND LENGTHS
- 037122

VARIABLE ASSIGNMENTS

BONE	=	000515	CLAB	=	035230C01	D	=	000512	I	=	000522
INT	=	000444	IT	=	000534	J	=	000524	K	=	000523
L	=	000532	M	=	000521	MB	=	000533	1JAN	=	000525
R	=	000535	SC	=	000530	SYM	=	037064C01	IX-1M	=	000526
W	=	000520	X	=	000000C01	XL	=	000531	AM	=	000507
XY	=	035321C01	YMAX	=	000527	Z	=	037103C01			

START OF CONSTANTS

000343

START OF TEMPORARIES

000426

START OF INDIRECTS

000440

UNUSED COMPILED SPACE

021200

		SUBROUTINE PLOT(X,ZMIN,ZMAX,WMIN,WMAX,NC,NP)	RM05000260
	CPL0TR	SUBROUTINE PLOT FOR RM0000 (MODIFIED) JULY 20, 1964	RM05000261
	C	BATTELLE COMMENT	RM05000262
	C	MASKING VECTOR (XM) REMOVED FROM PLOT, NOW IN FORM3.	RM05000263
000012		DIMENSION V(15000),Y(15),XY(51,1/),CLAB(57),AM(4),SYM(15)	RM05000264
		1 GF(10),FMT(10)	RM05000265
000012		INTEGER BLANKS,GF,FMT,TC,TP,T,XY,W	RM05000266
000012		DIMENSION TP(1),TC(1)	RM05000267
000012		COMMON V,CLAB,XY,SYM,Y	RM05000268
000012		COMMON /PRESET/ NCC	RM05000269
000012		DATA (NCC = 0)	RM05000270
000012	100	FORMAT(1H 0X5(F12.3,AX),F12.3/17X,5(F12.3,AX))	RM05000271
000012	101	FORMAT(1H F12.3,1X,A1,16A6,45,A1,F12.3)	RM05000272
000012	102	FORMAT(1H 13X,A1,16A6,45,A1)	RM05000273
000012	1000	FORMAT(1H 14X,101A1)	RM05000274
000012	1001	FORMAT(15X,20(5H+,...),1H+)	RM05000275
000012		BLANKS=(+6H)	RM05000276
000013		IF(NCC)4B,50,4B	RM05000277
000014	50	KL=0	RM05000278
000015		GF(1)=(+6H1X)	RM05000279
000017		GF(2)=(+6H2X)	RM05000280
000020		GF(3)=(+6H3X)	RM05000281
000022		GF(4)=(+6H4X)	RM05000282
000023		GF(5)=(+6H5X)	RM05000283
000025		GF(6)=(+6H6X)	RM05000284
000026		GF(7)=(+6H7X)	RM05000285
000030		GF(8)=(+6H8X)	RM05000286
000031		GF(9)=(+6H9X)	RM05000287
000033		GF(10)=(+6H10X)	RM05000288
000034		FMT(1)=(+6H117X)	RM05000289
000036		FMT(2)=BLANKS	RM05000290
000037		FMT(3)=BLANKS	RM05000291
000040		FMT(4)=(+6H5(F12.))	RM05000292
000041		FMT(5)=(+6H3,AX)/)	RM05000293
000043		FMT(6)=(+6H7X,)	RM05000294
000044		FMT(8)=(+6H4(F12.))	RM05000295
000046		FMT(9)=(+6H3,AX),)	RM05000296
000047		FMT(10)=(+6H12.3))	RM05000297
000051		TC=(+1H,)	RM05000298
000052		TP=(+1H,)	RM05000299
000054		CALL SCALE(WMIN,WMAX,100,0,JY,YMIN,YMAX,YI,0)	RM05000300

000062		YR=YMAX-YMIN	RM05000301
000064	230	J=JY	RM05000302
000066		IF(J*(J-10))204,201,201	RM05000303
000075	201	IF(KL)220,220,231	RM05000304
000077	231	WRITE(6,1001)	RM05000305
000103		IF(KL)250,250,220	RM05000306
000111	220	CLAB(1)=YMIN	RM05000307
000113		DO 222 I=2,11	RM05000308
000114	222	CLAB(I)=CLAB(I-1)+YIJ	RM05000309
000122		WRITE(6,100)(CLAB(I),I=1,11,2),(CLAB(J),J=2,10,2)	RM05000310
000141		IF(KL)231,231,14	RM05000311
000147	204	IF(J-5)205,221,207	RM05000312
000152	207	J=J-5	RM05000313
000154	205	JYI=5-J	RM05000314
000156	221	CONTINUE	RM05000315
000156		IF(KL)226,226,227	RM05000316
000160	226	FMT(3)=GF(JY)	RM05000317
000163	225	FMT(7)=GF(JY)	RM05000318
000166		TI=JY	RM05000319
000167		TI=TI*YIJ/10.	RM05000320
000171		CLAB(1)=YMIN+TI	RM05000321
000173		DO 223 I=2,10	RM05000322
000175	223	CLAB(I)=CLAB(I-1)+YIJ	RM05000323
000203		WRITE(6,FMT)(CLAB(I),I=2,10,2),(CLAB(J),J=1,9,2)	RM05000324
000223		IF(KL)227,227,14	RM05000325
000231	227	IF(JY-5)208,209,208	RM05000326
000233	209	WRITE(6,1002)	RM05000327
000237		IF(KL)250,250,226	RM05000328
000245	208	J = 5 - JYI	RM05000329
000247		WRITE(6,1000)(TC,I=1,J),(TP,(TC,I=1,4)),K=1,19,TP,	RM05000330
	1	(TC,I=1,JYI)	RM05000331
000306		IF(KL)250,250,226	RM05000332
000314	250	CONTINUE	RM05000333
000314		NCC=1	RM05000334
000315		IC=0	RM05000335
000316		IF(NP)80,11,11	RM05000336
000320	11	DO 1 I=1,51	RM05000337
000322		DO 1 J=1,17	RM05000338
000323	1	XY(I,J)=BLANKS	RM05000339
000333		CALL SCALE(ZMIN,ZMAX,50,X,XMIN,XMAX,XIJ)	RM05000340
000341		XR=XMAX-XMIN	RM05000341
000343	48	IF(NC)52,13,49	RM05000342
000350	49	IF(NP)80,10,10	RM05000343
000352	10	DO 9 N=1,NC	RM05000344
000354		SYMM = SYM(N)	RM05000345
000356		XDIFF=XMAX-X	RM05000346
000360		IF(XDIFF)105,106,106	RM05000347
000361	105	XDIFF=0.0	RM05000348
000362	106	YDIFF=YMAX-Y(N)	RM05000349
000366		IF(YDIFF)107,108,108	RM05000350
000367	107	YDIFF=0.0	RM05000351
000370	108	L=51.-(50.*XDIFF)/XR*.5	RM05000352
000377		K=101.-(100.*YDIFF)/YR*.5	RM05000353
000405		M=MOD(K,6)	RM05000354
000411		K=(K-1)/6+1	RM05000355
000415		IF(M)21,16,21	RM05000356
000416	16	M=6	RM05000357
000417	21	LL=M	RM05000358
000421		M=(M-1)*6	RM05000359
C			RM05000360
C		BATTELLE COMMENT	RM05000361
C		MASK IN PROPER CHARACTER - SUBROUTINE ADDED BY BATTELLE TO	RM05000362
C		REPLACE INDIANA CO-OP ROUTINE CALLED FORM2.	RM05000363

000423		CALL FORM3(XY(L,K), LL, SYMH)	RM05000364
000432	9	CONTINUE	RM05000365
000440		GO TO 15	RM05000366
000440	80	DO 86 I=1,17	RM05000367
000442	86	XY(1,I)=BLANKS	RM05000368
000450		L=1	RM05000369
000451	00	95 N=1,NC	RM05000370
000452		SYMH=SYM(N)	RM05000371
000454		YDIFFR=YMAX-Y(N)	RM05000372
000457		IF(YDIFFR)860,865,865	RM05000373
000450	860	YDIFFR=0.0	RM05000374
000461	865	K=101.-(100.*YDIFFR)/YR+.5	RM05000375
000470		M=MOD(K,6)	RM05000376
000474		IF(M)90,91,90	RM05000377
000475	91	M=6	RM05000378
000476	90	LL=M	RM05000379
000500		K=(K-1)/6+1	RM05000380
000504		M=(M-1)*2	RM05000381
			RM05000382
			RM05000383
			RM05000384
			RM05000385
			RM05000386
			RM05000387
			RM05000388
			RM05000389
			RM05000390
			RM05000391
			RM05000392
			RM05000393
			RM05000394
			RM05000395
			RM05000396
			RM05000397
			RM05000398
			RM05000399
			RM05000400
			RM05000401
			RM05000402
			RM05000403
			RM05000404
			RM05000405
			RM05000406
			RM05000407
			RM05000408
			RM05000409
			RM05000410
			RM05000411
			RM05000412
			RM05000413
			RM05000414
			RM05000415
			RM05000416
			RM05000417
			RM05000418
			RM05000419
			RM05000420
			RM05000421

000506	95	CALL FORM3(XY(L,K), LL, SYMH)
000523		IF(MOD(IC,5))97,96,97
000527	96	W=TP
000531		GO TO 98
000531	97	W=TC
000533	98	WRITE(6,101)X,W,(XY(1,N),N=1,17),W,X
000562		IC=IC+1
000564		GO TO 15
000570	13	M=6-JX
000572		LL=50+M
000574		T=JX
000575		IF(5-JX)131,131,135
000577	131	T=0.0
000600	135	RLAB=XMAX-(T*XIJ)/5.0
000605		W=TC
000606		K=52
000607		DO 31 L=M,LL
000611		K=K-1
000613		I=MOD(L,5)
000617		IF(I-1)2,3,2
000621	3	W=TP
000623		WRITE(6,101)RLAB,W,(XY(K,N),N=1,17),W,RLAB
000646		RLAB=RLAB-XIJ
000650		W=TC
000652		GO TO 31
000656	2	WRITE(6,102)W,(XY(K,N),N=1,17),W
000676	31	CONTINUE
000704	52	KL=1
000705		GO TO 230
000706	14	NCC=0
000707	15	RETURN
000710		END

SUMPROGRAM LENGTH
001164

PLOTR

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	-	000323	2	-	000656	3	-	000621	10	-	000352
11	-	000320	13	-	000570	14	-	000706	15	-	000707
16	-	000416	21	-	000417	31	-	000676	48	-	000343
49	-	000350	50	-	000014	52	-	000704	80	-	000440
86	-	000442	90	-	000476	91	-	000475	95	-	000527
97	-	000531	98	-	000533	100	-	000713	101	-	000721
102	-	000726	105	-	000361	106	-	000362	107	-	000367
108	-	000370	131	-	000577	135	-	000600	201	-	000075
204	-	000147	205	-	000154	207	-	000152	208	-	000245
209	-	000233	220	-	000111	241	-	000156	222	-	000115
223	-	000176	225	-	000163	226	-	000160	227	-	000231
230	-	000064	231	-	000077	250	-	000314	860	-	000460
865	-	000461	1000	-	000732	1001	-	000735			

BLOCK NAMES AND LENGTHS

- 037122 PRESET - 000001

VARIABLE ASSIGNMENTS

BLANKS	-	001126	CLAB	-	035230C01	FMT	-	001114	GF	-	001102
I	-	001142	IC	-	001146	J	-	001141	JX	-	001147
JY	-	001134	JYT	-	001143	K	-	001145	KL	-	001133
L	-	001160	LL	-	001142	M	-	001161	N	-	001154
NCC	-	000000C02	NP	-	000000	RLAB	-	001163	SYM	-	037064C01
SYMB	-	001155	T	-	001131	TC	-	001127	TS	-	001130
TI	-	001144	V	-	000000C01	W	-	001132	XDIFFR	-	001146
XIJ	-	001152	XN	-	001074	XMAX	-	001151	XMTN	-	001150
XR	-	001153	XY	-	035321C01	Y	-	037103C01	YDIFFR	-	001157
YIJ	-	001137	YMAX	-	001136	YMIN	-	001135	YR	-	001140

START OF CONSTANTS

000712

START OF TEMPORARIES

001022

START OF INDIRECTS

001072

UNUSED COMPILER SPACE

017400

```

SUBROUTINE TRANS(N,NJ,NTR)
CTRANS      SUBROUTINE TRANS FOR BMD05D
000006      DIMENSION DATA (15000)
000006      COMMON DATA
000006      TYPE INTEGER C123,TODE
000006      ASNF(X)=ATAN (X/SQRT (1.0-X**2))

C
000023      C123=(+6HTRNGEN)
000024      ON=N+1
000026      MARY=0
000030      WRITE(          6,1403)
000033      WRITE(          6,1406)
000037      IERROR=0
000040      DO 1000 I=1,NTR
000044      READ(          5,900)TODE,NE,NC,NV,CO
000061      IF (TODE-C123) 300,6,300
    
```

JULY 17, 1964

BMD05D00422
 BMD05D00423
 BMD05D00424
 BMD05D00425
 BMD05D00426
 BMD05D00427
 BMD05D00428
 BMD05D00429
 BMD05D00430
 BMD05D00431
 BMD05D00432
 BMD05D00433
 BMD05D00434
 BMD05D00435
 BMD05D00436
 BMD05D00437

000065	303 NJ=-NJ		8M05000438
000066	RETURN		8M05000439
000067	6 WRITE(	6.1402)I,NE,NC,NV,CO	8M05000440
000105	MA=N*NE=N		8M05000441
000111	MB=N*NV=N+1		8M05000442
000115	NC=MB*N-1		8M05000443
000116	IF(NC*(15-NC))1500,1500,2		8M05000444
000122	1500 WRITE(	6.1406)	8M05000445
000126	GO TO 1000		8M05000446
000131	2 IF(NC-11) 4, 3, 3		8M05000447
000134	3 K=CO		8M05000448
000136	MD=N*K-N		8M05000449
000141	4 DO 200 J=MB,MC		8M05000450
000143	MA=MA+1		8M05000451
000145	MD=MD+1		8M05000452
000146	5 CONTINUE		8M05000453
000146	GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140),NC		8M05000454
000170	10 IF(DATA(J))99,32,8		8M05000455
000173	8 DATA(MA)=SQRT (DATA(J))		8M05000456
000201	GO TO 200		8M05000457
000203	20 IF(DATA(J))99,11,12		8M05000458
000206	11 DATA(MA)=1.0		8M05000459
000211	GO TO 200		8M05000460
000211	12 DATA(MA)=SQRT (DATA(J))+SQRT (DATA(J)+1.0)		8M05000461
000226	GO TO 200		8M05000462
000230	30 IF(DATA(J))99,99,14		8M05000463
000233	14 DATA(MA)=ALOG(DATA(J))+0.43+2944819		8M05000464
000241	GO TO 200		8M05000465
000244	40 DATA(MA)=EXP (DATA(J))		8M05000466
000252	GO TO 200		8M05000467
000254	50 IF(DATA(J))99,32,17		8M05000468
000257	17 IF(DATA(J)-1.0)18,19,99		8M05000469
000263	19 DATA(MA)=3.1415926536/2.0		8M05000470
000266	GO TO 200		8M05000471
000266	18 A=SQRT (DATA(J))		8M05000472
000272	DATA(MA)=ASNFA)		8M05000473
000300	GO TO 200		8M05000474
000300	60 A=DATA(J)/ON		8M05000475
000303	B=A+1.0/ON		8M05000476
000305	IF(A)99,23,24		8M05000477
000307	23 IF(B)99,26,27		8M05000478
000311	26 DATA(MA)=0.0		8M05000479
000313	GO TO 200		8M05000480
000314	27 DATA(MA)=ASNFA(SQRT (B))		8M05000481
000325	GO TO 200		8M05000482
000325	24 IF(B)99,28,29		8M05000483
000327	28 DATA(MA)=ASNFA(SQRT (A))		8M05000484
000340	GO TO 200		8M05000485
000340	29 A=SQRT (A)		8M05000486
000342	B=SQRT (B)		8M05000487
000345	DATA(MA)=ASNFA(A)+ASNFB)		8M05000488
000357	GO TO 200		8M05000489
000357	70 IF(DATA(J))31,99,31		8M05000490
000361	31 DATA(MA)=1.0/DATA(J)		8M05000491
000365	GO TO 200		8M05000492
000366	80 DATA(MA)=DATA(J)+CO		8M05000493
000373	GO TO 200		8M05000494
000373	90 DATA(MA)=DATA(J)*CO		8M05000495
000377	GO TO 200		8M05000496
000400	100 IF(DATA(J))33,32,33		8M05000497
000402	32 DATA(MA)=0.0		8M05000498
000404	GO TO 200		8M05000499

```

000405      33 DATA(MA)=DATA(J)**CO
000413      60 TO 200
000414      110 DATA(MA)=DATA(J)+DATA(MD)
000422      60 TO 200
000422      120 DATA(MA)=DATA(J)-DATA(MD)
000430      60 TO 200
000430      130 DATA(MA)=DATA(J)*DATA(MD)
000435      60 TO 200
000436      140 IF(DATA(MD))157,99,157
000440      157 DATA(MA)=DATA(J)/DATA(MD)
000445      60 TO 200
000446      99 IF(MARY)43,44,44
000450      44 MARY=-999
000451      IERROR=-999
000452      WRITE(          6,1404)I
000460      43 WRITE(          6,1405)J
000466      200 CONTINUE
000473      MARY=0
000474      1000 CONTINUE
000477      IF(IERROR)42,1111,1111
000500      42 WRITE(          6,1401)
000504      1111 RETURN
C
000505      900 FORMAT(A6,I3,I2,I3,F6.0)
C
000505      1400 FORMAT(48HOCARD NEW TRANS ORIG. ORIG. VAR(B)/45H NO.
000505      1401 FORMAT(78H VALUES OF VARIABLES OF WHICH AN ERROR WAS FOUND DURING
000505      1TRANS-GENERATION WILL /77H STILL BE INCLUDED IN THE GRAPHS. HOWEVER
000505      2R. THESE GRAPHS MAY BE MEANINGLESS /54H SINCE SOME VALUES WILL B
000505      3E TRANSFORMED AND OTHERS NOT.)
000505      1402 FORMAT(2H I2,I8,2I9,4X,F10.5)
000505      1403 FORMAT(1H06X,23HTRANS GENERATOR CARD(S))
000505      1404 FORMAT(55H0THE INSTRUCTIONS INDICATED ON TRANS GENERATOR CARD NO.1
000505      12,1X,3HRE-/60H SULTED IN THE VIOLATION OF A RESTRICTION FOR THIS T
000505      2TRANSFOR-/59H MATION. THE VIOLATION OCCURRED FOR THE ITEMS LISTED B
000505      3ELOW.)
000505      1405 FORMAT(10H ITEM NO. 15)
000505      1406 FORMAT(107H0TRANSGENERATION CODE ON CARD LISTED ABOVE IS INCORRECT
000505      X. PROGRAM WILL PROCEED WITHOUT THIS TRANSGENERATION.)
C
000505      END

SUBPROGRAM LENGTH
000732
TRANS

FUNCTION ASSIGNMENTS
ASNF = 000010

STATEMENT ASSIGNMENTS
2 - 000131 3 - 000134 4 - 000141 5 - 000146
6 - 000067 8 - 000173 10 - 000170 11 - 000206
12 - 000211 14 - 000233 17 - 000257 18 - 000266
19 - 000263 20 - 000203 23 - 000307 24 - 000325
26 - 000311 27 - 000314 28 - 000327 29 - 000340
30 - 000230 31 - 000361 32 - 000402 33 - 000405
40 - 000244 42 - 000500 43 - 000460 44 - 000450
50 - 000254 60 - 000300 70 - 000357 80 - 000366
90 - 000373 99 - 000446 100 - 000400 110 - 000414
120 - 000422 130 - 000430 140 - 000436 157 - 000440
200 - 000466 300 - 000065 900 - 000521 1000 - 000474
1111 - 000504 1400 - 000524 1401 - 000540 1402 - 000570
1403 - 000574 1404 - 000601 1405 - 000626 1406 - 000631
1500 - 000122

```

BLOCK NAMES AND LENGTHS
- 035230

VARIABLE ASSIGNMENTS

A	=	000730	B	=	000731	CO	=	000721	C1P3	=	000710
DATA	=	000900C01	I	=	000715	ERROR	=	000714	J	=	000727
K	=	000725	MA	=	000722	MARY	=	000713	MB	=	000723
MC	=	000724	MD	=	000726	NC	=	000717	NE	=	000716
NV	=	000720	ON	=	000712	TUDE	=	000711			

START OF CONSTANTS
000507

START OF TEMPORARIES
000646

START OF INDIRECTS
000702

UNUSED COMPILER SPACE
020400

		SUBROUTINE VFCHCK(NVF)		
000003	CVFCHCK	SUBROUTINE TO CHECK FOR PROPER NUMBER OF VARIABLE FORMAT CARDS		BM05000542
000006	10	IF(NVF)10,10,20		BM05000543
000010		WRITE(	6,4000)	BM05000544
000012	50	NVF=1		BM05000545
		RETURN		BM05000546
000013	20	IF(NVF=10)50,50,10		BM05000547
000016	4000	FORMAT(1M023X7)NUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF		BM05000548
		KIED, ASSUMED TO BE 1.)		BM05000549
000016	END			BM05000550

SUBPROGRAM LENGTH
000036
VFCHCK

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10 - 000004 20 - 000013 50 - 000012 4000 - 000022

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS
000020

START OF TEMPORARIES
000034

START OF INDIRECTS
000036

UNUSED COMPILER SPACE
022600

```

SUBROUTINE SCALE(YMIN,YMAX,YINT,JY,TYMIN,TYMAX,YIJ)
SUBROUTINE SCALE FOR SUB PLOT R      AUGUST 18, 1964
CSCALE
DIMENSION C(10)
000012      C(1)= 1.0
000012      C(2)=1.5
000013      C(3)=2.0
000014      C(4)=3.0
000016      C(5)=4.0
000017      C(6)=5.0
000021      C(7)=7.5
000022      C(8)=10.0
000024      TEST=.5*(2**(-24))
000025      50  YR=YMAX-YMIN
000033      TT=YR/YINT
000034      J=ALOG(TT)*0.43+29.4819
000036      E=10.0**J
000042      TT=TT/E
000046      I=0
000047      IF(TT-1.0)205,201,201
000050      205  TT=TT*10.0
000056      E=E/10.0
000060      201  I=I+1
000062      IF(8-I)1,2,2
000064      1  E=E*10.0
000066      I=1
000070      2  IF(TT-C(I))233,202,201
000071      233  YIJ=C(I)*E
000075      GO TO 203
000101      202  Y=TYMIN/C(I)
000102      J=Y
000105      T=J
000107      IF(0.0001-ABS (T-Y))204,233,233
000110      204  YIJ=C(I+1)*E
000115      203  X=((YMAX+YMIN)/YIJ-YINT)/2.0+.00001
000120      K=X
000125      IF(K)235,240,240
000127      235  Y=K
000131      IF(X-Y)236,240,236
000132      K=K-1
000135      236  TYMIN=K
000137      240  TYMIN=YIJ*TYMIN
000140      TYMAX=TYMIN+YINT*YIJ
000142      IF(YMAX-TYMAX-TEST)10,10,401
000144      10  TT=YINT/10.
000147      JY=TT+.000001
000150      YIJ=YINT*(YIJ/10.0)
000153      J=TYMIN/YIJ
000155      IF (K)242,241,241
000157      242  J=J-1
000161      241  J=J*JY+JY-K
000163      JY=J
000167      RETURN
000170      END

```

SUBPROGRAM LENGTH
000244
SCALE

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	-	000066	2	-	000071	10	-	000147	50	-	000033
201	-	000062	202	-	000102	203	-	000120	204	-	000115
205	-	000056	233	-	000075	235	-	000131	236	-	000135
240	-	000137	241	-	000163	242	-	000161			

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

C	=	000220	E	=	000236	I	=	000237	J	=	000235
K	=	000243	T	=	000241	TEST	=	000232	TT	=	000234
X	=	000242	Y	=	000240	YIJ	=	000000	YR	=	000233

START OF CONSTANTS

000172

START OF TEMPORARIES

000211

START OF INDIRECTS

000217

UNUSED COMPILER SPACE

021700

SUBROUTINE FILL(SYM, NC)

```

C
C
C   BATTELLE SUBROUTINE
C
C   THIS ROUTINE FILLS THE COMPUTER WORD WITH THE INPUT SYMBOL SYM(N)
C   ADDED TO INDIANA CO-OP VERSION IN ORDER TO COMPLEMENT ADDED
C   BATTELLE SUBROUTINE CALLED FORM3.
C
C   THIS VERSION IS FOR A X-BYTE WORD, WHERE X = 10.
C
000005   DIMENSION SYM(15)
C
000005   DO 10 N=1,NC
000006     10  ENCODE(10,1000,SYM(N)) (SYM(N), I=1,10)
000031     RETURN
C
000031     1000 FORMAT(10H1)
000031     END

```

RMD5000607
 RMD5000608
 RMD5000609
 RMD5000610
 RMD5000611
 RMD5000612
 RMD5000613
 RMD5000614
 RMD5000615
 RMD5000616
 RMD5000617
 RMD5000618
 RMD5000619
 RMD5000620
 RMD5000621
 RMD5000622
 RMD5000623
 RMD5000624

SUBPROGRAM LENGTH

000043

FILL

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1000 = 000034

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

I	=	000042	N	=	000041
---	---	--------	---	---	--------

START OF CONSTANTS

000033

START OF TEMPORARIES

000036

START OF INDIRECTS

000041

UNUSED COMPILER SPACE

022500

```

C      SUBROUTINE FORM3(GRAPH, LOC, SYMBOL)
C
C      BATTELLE SUBROUTINE
C
C      THIS SUBROUTINE PERFORMS ALL THE MASKING OPERATIONS FOR PLOT3.
C
C      GRAPH = THE COMPUTER WORD AT THE COORDINATES TO BE LOADED.
C      LOC = THE BYTE LOCATION WITHIN THE COMPUTER WORD FOR LOADING.
C      SYMBOL = THE INPUT CHARACTER TO BE LOADED. MUST FILL THE
C      SIX LEFT BCD CHARACTERS OF THE COMPUTER WORD.
C
C      THIS ROUTINE AND THE ENTIRE PROGRAM TREATS THE PLOTTING WORD
C      AS IF IT WERE ONLY A ALPHANUMERIC CHARACTERS LONG. LEFT-JUSTIFIED.
C
C      THIS ROUTINE LOADS THE INPUT SYMBOL AT THE LOCATION LOC WITHIN
C      THE COMPUTER WORD GRAPH.
C      IF AN ITEM ALREADY EXISTS AT A LOCATION TO BE LOADED,
C      THE ROUTINE WILL LOAD AN NUMBER OR LETTER INDICATING THE NUMBER
C      OF ITEMS AT THAT LOCATION. REFER TO TABLE FOR MULTIPLE-ITEM
C      ALPHANUMERICS. THE NUMBER OF ITEMS, N, IS EQUAL TO THE
C      CHARACTER STORED AT TABLE(N-1).
C      NOT ALL CALLING PROGRAMS USE THIS MULTIPLE-ITEM FEATURE, BUT
C      IT HAS BEEN INCLUDED FOR THE SAKE OF STANDARDIZATION.
C
000006      DIMENSION TABLE(18), MASK(8)
000006      INTEGER TABLE, GRAPH, SYMBOL, BLANKS
000006      DATA(BLANKS=0H
C
000006      DATA(TABLE=6H222222,6H333333,6H444444,6H555555,6H666666,6H777777,
1          6H888888,6H999999,6HAAAAAA,6HBBBBBB,6HCCCCCC,6HDDDDDD,
2          6HEEEEE,6HFFFFFF,6HGGGGGG,6HHHHHHH,6HIIIIII,6H////////)
C
000006      DATA (MASK = 77000000000000000000, 00770000000000000000,
1          00007700000000000000, 00000077000000000000,
2          00000000770000000000, 00000000007700000000)
C
C      MASK OUT LOCATION UNDER INVESTIGATION.
000006      ITEM = GRAPH.AND.MASK(LOC)
C
C      TEST TO SEE IF THERE ARE ALREADY 19 OR MORE ITEMS AT LOCATION
000010      LOGIC = TABLE(18).AND.MASK(LOC)
000012      IF(ITEM.EQ.LOGIC) RETURN
C
C      TEST TO SEE IF BLANK (IF NO PREVIOUS ITEMS AT LOCATION).
000015      LOGIC = BLANKS.AND.MASK(LOC)
000020      IF(ITEM.NE.LOGIC) GO TO 10
C
C      IF BLANK AT LOCATION, MASK IN SYMBOL.
000022      GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(SYMBOL.AND.MASK(LOC))
000027      RETURN
C
C      SCAN THRU MULTIPLE-ITEM TABLE FOR MATCH.
000030      DO 20 I=1,17
000032      LOGIC = TABLE(I).AND.MASK(LOC)
000035      IF(ITEM.NE.LOGIC) GO TO 20
C
C      IF MULTIPLE-SYMBOL IS FOUND, MASK IN NEXT SYMBOL.
000040      GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(TABLE(I+1).AND.MASK(LOC))
000045      RETURN
C
000046      20 CONTINUE

```

```

RMD5000625
RMD5000626
RMD5000627
RMD5000628
RMD5000629
RMD5000630
RMD5000631
RMD5000632
RMD5000633
RMD5000634
RMD5000635
RMD5000636
RMD5000637
RMD5000638
RMD5000639
RMD5000640
RMD5000641
RMD5000642
RMD5000643
RMD5000644
RMD5000645
RMD5000646
RMD5000647
RMD5000648
RMD5000649
RMD5000650
RMD5000651
RMD5000652
RMD5000653
RMD5000654
RMD5000655
RMD5000656
RMD5000657
RMD5000658
RMD5000659
RMD5000660
RMD5000661
RMD5000662
RMD5000663
RMD5000664
RMD5000665
RMD5000666
RMD5000667
RMD5000668
RMD5000669
RMD5000670
RMD5000671
RMD5000672
RMD5000673
RMD5000674
RMD5000675
RMD5000676
RMD5000677
RMD5000678
RMD5000679
RMD5000680
RMD5000681
RMD5000682
RMD5000683
RMD5000684
RMD5000685
RMD5000686

```

```

      C      IF NO MATCH IS FOUND, THEN MASK IN A 2.
000050      GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(TABLE(1).AND.MASK(LOC))
000056      RETURN
000056      END

```

```

AMD5000687
AMD5000688
AMD5000689
AMD5000690

```

```

SUBPROGRAM LENGTH
000131

```

FORM3

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

```

10      - 000030      20      - 000046

```

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

```

BLANKS - 000125      I      - 000130      ITEM - 000126      LOCIC - 000127
MASK   - 000117      TABLE - 000075

```

START OF CONSTANTS

```

000060

```

START OF TEMPORARIES

```

000061

```

START OF INDIRECTS

```

000071

```

UNUSED COMPILER SPACE

```

022300

```

SCATTERGRAM ROUTINE

**for non-blank entries for both variables
in the same data set**

REFERENCES (RELATIVE)

ENTRY ADDRESS

SCAT	101	124	17	21	22	25
ENFIL	4/52	7	30	41	43	44
INPUTC	5144	27	47	60	63	65
		64				

KRANK
QNTARY
SYSTEM

SCAT	5150	2				
ENFIL	6166	61				
INPUTC	4373	37				
		14				

SYSTEMC
SYSTEMAD
END
STDP
EATL
AMVORNL

SCAT	4337	125				
ENFIL	4366	62				
INPUTC	4262	40				
SCAT	4312	15				
	4304	110				
	4322					

OUTPTC
KUTER
FATALTH
CARUF
AKSPRJ
FIZMAS
POSFIL

ENFIL	7240	62				
INPUTC	7377	40				
SCAT	40577	15				
	40674	110				
	11203					
	11213					
	11241					

ADJHJ
DAT
CITI
DREN

ENFIL	11331	45				
INPUTC	11352	32				
ENFIL	11157	56				
INPUTC	10716	37				
		11				
		22				
		506				
		22				
		52				
		71				
		30				
		410				

SID
ADVIN
MVAS

ENFIL	11035	72				
INPUTC	11252	37				
ENFIL	11064	15				

LOAD MAP FILE - 140

ENTRY ADDRESS

REFERENCES (RELATIVE)

POSFL	11262	24				
FIZMA	11274					
DAT	11502	55				
BETMA	12032	5				
		10				
		10				

UNSATISFIED CATEMVALS

REFERENCES (RELATIVE)

00 ONE 00

LOAD MAP FILE - 140

FMA LOAD 100 FMA LOAD 00632 FMA LOADER 75303 FMA TABLES 74072 DIVISE STORAGE 5200

PROGRAM ADDRESS FILE COMMON ADDRESS LENGTH

BM22D 100 L40
 COOL 11170 L40
 CONV 11600 L60
 PLJ1 11712 L40
 TRNGEN 12072 L60
 VFCHCK 13230 L60
 SCALE 14025 L60
 AC3UEK 14053 L40
 ALNLD3 14322 SYSTEM
 ATAN 14334 SYSTEM
 ENDFIL 14423 SYSTEM
 EXP 14517 SYSTEM
 INDUCT 14612 SYSTEM
 SYSTIE4 14671 SYSTEM
 SYSTIE4 16012 SYSTEM
 INDUCT 17063 SYSTEM
 INDUCT 17063 SYSTEM
 OUTPT8 17470 SYSTEM
 OUTPT8 17725 SYSTEM
 OUTPTS 21265 SYSTEM
 REALEX 21351 SYSTEM
 REALIN 21412 SYSTEM
 SURT 21505 SYSTEM
 THALER 21550 SYSTEM
 R-AMEX 21601 SYSTEM
 FATAL/H 21660 SYSTEM
 SIDS 21774 SYSTEM
 GETHA 23114 SYSTEM

SCOPE2 16012

BLANK/ 23122 43477

REFERENCES (RELATIVE)

ENTRY	ADDRESS	REFERENCES (RELATIVE)	SCOPE2	BLANK/	SCOPE2	BLANK/	SCOPE2	BLANK/
BM22D	101	562	573	647	640	674	505	
COOL	11200	517	1247	1314				
CONV	11691	1205						
PATY	11713	1323						
PLJ1	12073	511	621	631	641	722	732	
TRNGEN	13237	202						
VFCHCK	14026	45	51					
SCALE	14065	551	577	600	711	1057		
AC3UEK	14323	74	154					
ALD3	14340	74						
ALD310	14335	164						
ATAN	14424	33						
ENDFIL	14520	17						
FAD	14613	50						
		1125						
		175						
		22						

[illegible]

LOG MAP
Entry ADDRESS

FILE - 150
REFERENCES (RELATIVE)

22.23.14. 12/10/60. PAGE 3

OUTPUT	1727	44020	153	154	207	211	213	216	224
			228	229	255	247	201	203	245
			267	270	304	317	320	303	305
			353	355	361	363	304	1077	1141
			1102	1107	1133	1134	1137	1143	1147
			1152	1163	1171	1175	1202	1203	1213
			1214	1239	1242	1244	1205	1212	1273
			1227	1330	1374	1400	1401	1606	1608
		COUL	224	224	1374	1400	1401	1606	1608
		PALTY	40	44	55	57	64	72	
		PLUT	245	247	253	256	260	262	264
			314	322	330	333	344	336	342
			670	672	680	681	684	686	688
			422	424	429	432	433	435	437
		TRNGEN	5	7					
		VFCCHK	5	7					
		OUTPUTS	42	47					
		CONV	107	111	115	117	121	123	125
		PLUT	127	131	133	137	140		
		SCALE	134	139					
		44020	40						
		COUL	273						
		PLUT	165	516					
		44020	1224						
		TRNGEN	25	132	151	216	204	201	274
		COUL	274						
		TRNGEN	65						
		SYSTEM	335						
		SIUS	710						
			56						
		ENUFIL	45						
		OUTPUTC	32						
		OUTPUTC	55	72					
		ENUFIL	44	17					
		INPUTM	37	15					
		OUTPUTB	70	103	25	202	232	207	177
			74	116	40	135	64	175	
			221						
		RESIM	37	24					
		ENUFIL	11						
		INPUTC	22						
		SYSTEM	505						
		INPUTB	155	134					
		OUTPUTB	75	131					
		OUTPUTC	22						
		INPUTC	52						
		INPUTB	34						
		OUTPUTB	30						
		OUTPUTC	71						
		ENUFIL	30						
		SYSTEM	410						
		RESIM	34						
		INPUTB	54						
		REFERENCES (RELATIVE)							
			43						
		OUTPUTB	24						
		ENUFIL	124						
		OUTPUTB							

22.23.14. 12/10/60. PAGE 4

Continued on following page

NOT REPRODUCIBLE

FILE NO. 22750
 DATE 2/6/64
 RET 2 23114

INPUTS 54
 INPUTS 17
 ENUTL 5
 INPUTS 10
 INPUTS 121
 OUTPUTS 52
 OUTPUTS 10
 REMAINING 5
 REFERENCES (RELATIVE)

UNCLASSIFIED EXTENSIVE
 00 04 86

BMU020 CORRELATION WITH TRANSGENERATION - VERSION OF NOV. 13, 1964
 HEALTH SCIENCES COMMISSION FACILITY, JCL

PROBLEM CODE JH00
 NUMBER OF VARIABLES 2
 NUMBER OF CASES 111

REMAINING SAMPLE SIZE 111
 SUMS

7180.0000 275.5100

MEANS

64.6847 2.5722

CROSS PRODUCT DEVIATIONS

COL. COL. COL.
 1 75687.9540 1.540457
 2 174.0457 4.0641

STANDARD DEVIATIONS

26.2311 1.921

VARIANCE-COVARIANCE MATRIX

COL. COL. COL.
 1 688.0724 1.5422
 2 1.5422 .0370

CORRELATION MATRIX

COL. COL. COL.
 1 1.0000 .3134
 2 .3134 1.0000

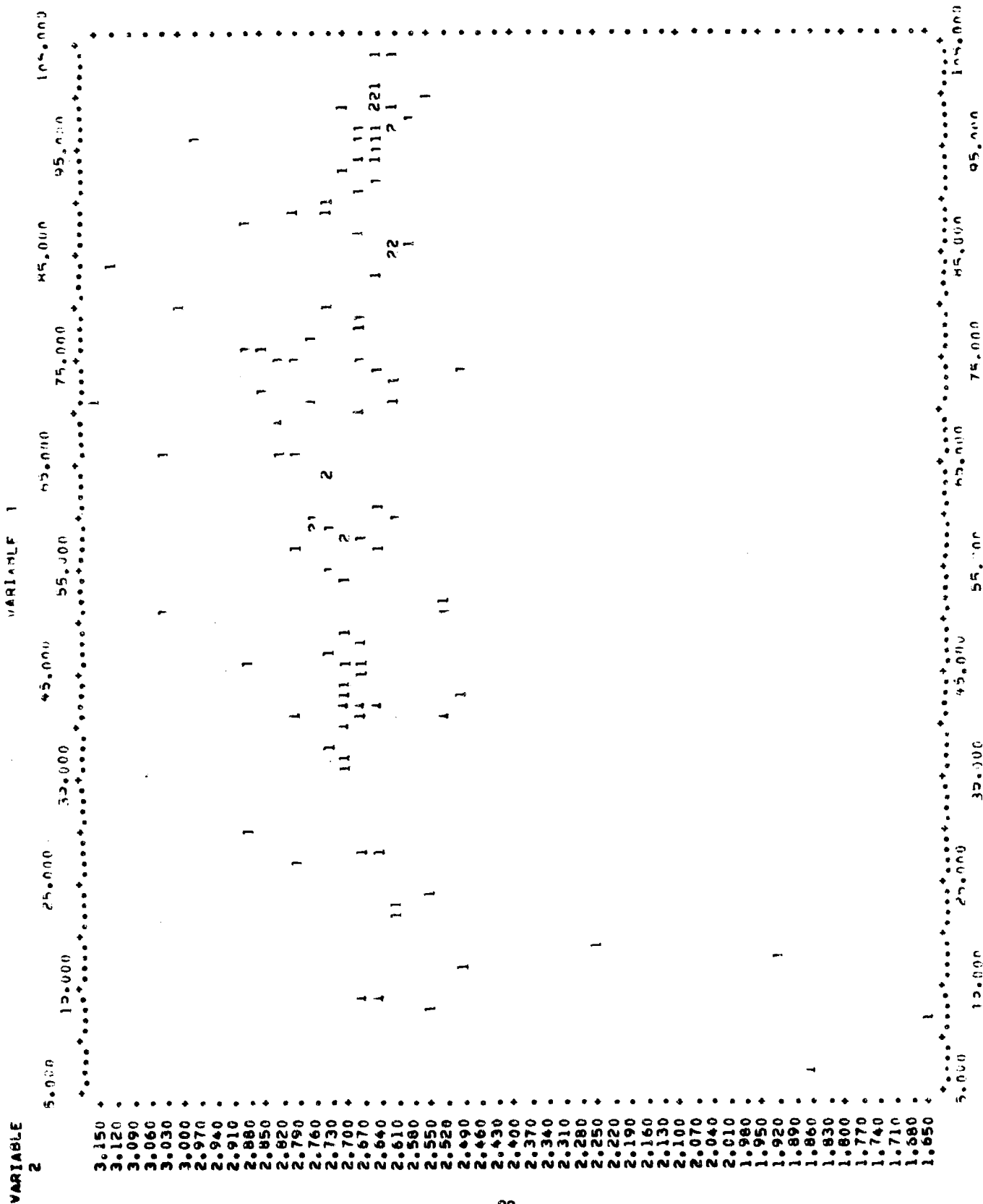
NOT REPRODUCIBLE

MM53550. 12/19/54. OAKS 16 MAPS 11/27/54.

```

22.05.24.MH327 3512.744.5.150.CW/60000.FP1.010
22.05.24..
22.05.24.MAP(0V)
22.05.24.RUN(S)
22.05.28.CTIME 000.213 SEC. 0 IN MOD LEVEL 03
22.05.29.REQUEST(TAPE1.555.HY.A.CER4.HT.WC40)
22.16.09. MT50 ASSIGNED - 555
22.16.10.REWIND(TAPE1)
22.16.10.LGO.
22.16.12.CX 4.489 SEC.
22.16.12.PX 4.422 SEC.
22.16.12.NL 12200
22.23.05.END SCAT
22.23.05.REWIND(TAPE1)
22.23.05.RETURN(TAPE1)
22.23.05. TAPE LIMIT=0
22.23.05.REWIND(LGO)
22.23.05.RFL(75000)
22.23.05.CX 20.561 SEC.
22.23.05.PX 344.050 SEC.
22.23.05.NL 16000
22.23.08.REWIND(TAPE1)
22.23.08.LTRCOPY(STATIN,LG0004020)
22.23.11.LGO.
22.23.17.CX 22.142 SEC.
22.23.17.PX 344.914 SEC.
22.23.17.NL 57000
22.23.23.STOP
22.23.23.CP 25.054 SEC.
22.23.23.PP 351.265 SEC.
22.23.23.LINES = 0573 OCTAL
22.23.23.CM 2.551 MW0=SEC.

```



MM53850. 12/14/69. P. 1000 IF MARCH 11/27/69.

22.05.24.MM5385/ 3512.544.5.160.0M/6000.1P1.010
22.05.24..
22.05.24.MAP(ON)
22.05.24.RUN(S)
22.05.28.CTIME 000.213 SEC. DIN MOD LEVEL 43
22.05.29.REQUEST(TAPF1.556.0Y.1.0C00.0T.0E00)
22.16.09. MISO ASSIGNED - 556
22.16.10.REWIND(TAPF1)
22.16.10.LGO.
22.16.12.CX .480 SEC.
22.16.12.PX 4.422 SEC.
22.16.12.NL 12200
22.23.05.END SCAT
22.23.05.REWIND(TAPF1)
22.23.05.RETURN(TAPF1)
22.23.05. TAPE LIMIT=0
22.23.05.REWIND(LGO)
22.23.05.RFL(75000)
22.23.05.CX 20.661 SEC.
22.23.05.PX 344.050 SEC.
22.23.05.NL 76000
22.23.08.REWIND(TAPF1)
22.23.08.LIACOPY(STATIN,LGO,MOD20)
22.23.11.LGO.
22.23.17.CX 22.142 SEC.
22.23.17.PX 349.914 SEC.
22.23.17.NL 67000
22.23.23.STOP
22.23.23.CP 25.054 SEC.
22.23.23.PP 351.285 SEC.
22.23.23.LINES = 0573 OCTAL
22.23.23.CM 2.501 MOD-SEC.

BMD PROGRAM TO PRODUCE SCATTERGRAM

EDITSYM CONTROL CARDS

*COPY*BM020,8M050

```

PROGRAM BM020 (INPUT=1,OUTPUT=1,TAPE=4,TAPE5=INPUT,TAPE6=OUTPUT,
ITAPER)
CH00020 CORRELATION WITH TRANSGENERATION NOVEMBER 13, 1964
C 0400 CONVERSION REPLY BENSON
C HEALTH SCIENCES COMPUTING FACILITY, UCLA
000003 TYPE INTEGER WK,ASK,4123,H123,C123,D123,A1,100E,CONF,REL,OP,FMT,PI
IS
000003 TYPE INTEGER PC7,P01,P02,P03
000003 DIMENSION AMAX(150),AMIN(150),YY(15),SYM(120),NS(120),SX(150)
1 ,SA2(150),SXY(135,135),DATA(150),FMT(120)
000003 DIMENSION NEW(150),JUMP(150),NA(150),RN(150)
000003 DIMENSION CON(36),NSUB(36),REL(36),OP(36),INDEX(36),WK(8)
000003 DIMENSION IDATA(14)
000003 COMMON SXY
000003 COMMON IDATA
000003 DIMENSION C(255),Q(150)

C
000003 100 FORMAT(54HBM020 CORRELATION WITH TRANSGENERATION - VERSION OF
A15NOV. 13, 1964 /
140H HEALTH SCIENCES COMPUTING FACILITY,UCLA//
214H PROBLEM CODE AA./
321H NUMBER OF VARIABLES 13//
417H NUMBER OF CASES 15,/)

C
000003 PC7=(+2HNO)
000005 WK(1)=(+6HGT )
000006 WK(2)=(+6HGE )
000010 WK(3)=(+6HMT )
000011 WK(4)=(+6HLE )
000013 WK(5)=(+6HEW )
000014 WK(6)=(+6HNE )
000016 WK(7)=(+6HOM )
000017 WK(8)=(+6HAM )
000021 ASK=(+6H** )
000022 C123=(+6HTRNGEN)
000024 D123=(+6HPLUTSL)
000025 H123=(+6HPRHMLM)
000027 A123=(+6HFINISH)
000030 A1=(+6MV )
000032 NTAPE=5
000033 IT1=4

C
000034 49H HEAD( 5,102)T0DE,CODE, NVAR,NSAM,NSCL,NADD,NROUL,PU1,PW2,
IPQ3,NTG,MTAPE,KVR
GO TO 496
000072 499 OFCODE(80,102,TDATA)T0DE,CONF,NVAR,NSAM,NSCL,NADD,NROUL,PU1,PW2,PQ
000073 13,NIG,MTAPE,KVR
496 REWIND IT1
000133 IF (T0DE-A123)/00,70,700
000135 700 IF (T0DE-D123)/010,703,9010
000137 703 IF (MTAPE-EP,0) 741,742
000141 741 MTAPE=5 GO TO 773
000145 742 IF (MTAPE-NE,4) GO TO 773
000147 PHINT 5
000151 5 FORMAT (////////// * YOU HAVE ASSIGNED LOGICAL NUMBER 4 TO YOUR DATA
000155 *TAPE. CHOOSE SOME OTHER NUMBER,//////////)

```

NOT REPRODUCIBLE

000155	701 STOP	RM02000057
000157	773 CONTINUE	RM02000058
000157	IF (NVAR-1)*(135-NVAR) 9090,9000,100K	RM02000059
000164	1008 IF (NSAM-1)9001,9001,1002	RM02000060
000167	1002 IF ((NVAR+NAUD-1)*(135-NVAR-NAUD)) 9002,9002,705	RM02000061
000176	705 IF (IAHS (NHUOL)-9)706,706,9003	RM02000062
000201	706 CALL VFCHECK(KVR)	RM02000063
000203	3 WRITE (6,100)CODE,NVAR,NSAM	RM02000064
000215	IF (NTG-150) 1003,1003,4004	RM02000065
000220	1003 IF (NTG)9005,401,402	RM02000066
000222	402 WRITE (6,403)	RM02000067
000226	WRITE (6,404)	RM02000068
000232	DO 707 I=1,NTG	RM02000069
000234	READ (5,406)TODE,NEW(I),JUMP(I),NA(I),CN(I)	RM02000070
000251	IF (TODE-C123)9006,405,9006	RM02000071
000253	405 WRITE (6,407)I,NEW(I),JUMP(I),NA(I),CN(I)	RM02000072
000271	IF (JUMP(I)-41)2000,707,2005	RM02000073
000274	2000 IF (JUMP(I)*(17-JUMP(I)))2005,2005,707	RM02000074
000301	2005 WRITE (6,4001)	RM02000075
000305	JUMP(I)=99	RM02000076
000307	707 CONTINUE	RM02000077
000312	401 IF (NHUOL) 411,412,411	RM02000078
000313	411 KK=IAHS (NHUOL)*4	RM02000079
000316	WRITE (6,413)	RM02000080
000321	READ (5,414) (SUB(I),HEL(I),CON(I),OP(I),T=1,KK)	RM02000081
000342	WRITE (6,415)	RM02000082
000346	DO 416 I=1,KK	RM02000083
000350	KK1=I	RM02000084
000351	WRITE (6,417)NSUR(I),HEL(I),CON(I),OP(I)	RM02000085
000365	IF (ASK-OP(I))416,1234,416	RM02000086
000370	416 CONTINUE	RM02000087
000373	1234 DO 438 I=1,KK1	RM02000088
000375	438 INDEX(I)=0	RM02000089
000401	DO 437 I=1,KK1	RM02000090
000402	DECODE (1,709,CON(I))PUS	RM02000091
000411	709 FORMAT(A1)	RM02000092
000411	IF (A1-PUS)710,711,710	RM02000093
000413	711 DECODE (5,712,CON(I))CON(I)	RM02000094
000423	712 FORMAT(2X,F3.0)	RM02000095
000423	INDEX(I)=1	RM02000096
000425	GO TO 437	RM02000097
000426	710 DECODE (6,713,CON(I))CON(I)	RM02000098
000436	713 FORMAT(F6.0)	RM02000099
000436	437 CONTINUE	RM02000100
000441	412 IF (NTG*NH00L) 423,418,425	RM02000101
000445	418 IF (NTG)4007,419,424	RM02000102
000447	419 IF (NH00L)422,421,422	RM02000103
000450	421 JESUS=1	RM02000104
000451	GO TO 7	RM02000105
000452	422 JESUS=2	RM02000106
000453	N03=0	RM02000107
000454	GO TO 7	RM02000108
000455	424 JESUS=3	RM02000109
000456	GO TO 7	RM02000110
000457	425 JESUS=4	RM02000111
000460	N03=0	RM02000112
000461	GO TO 7	RM02000113
000462	423 JESUS=5	RM02000114
000463	N03=0	RM02000115
000464	7 M=0	RM02000116
000465	LCASE=0	RM02000117
000466	LEFT=NSAM	RM02000118
000467	NVAR1=NVAR+NAUD	RM02000119
000471	DO 4 I=1,NVAR1	RM02000120

```

000473      AMIN(I)=10.**10
000476      AMAX(I)=-AMIN(I)
000477      SX(I)=0.0
000500      SX2(I)=0.0
000501      DO 4 J=1,NVARI
000503      4 SXY(I,J)=0.0
000513      KL=0
000514      H=0.
000515      6 KVM=KVR*6
000517      HEAD(      5,103)(FMT(I),I=1,NV4)
000531      77 DO 600 II=1,NSAM
000533      HEAD(      MFAPE,FMT)(DATA(I),I=1,NVARI)
000546      GO TO (407,427,428,424,430),JESUS
000557      427 CALL COOL(NSUB,REL,CUN,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N
      ITAPE)
      GO TO (600,431,999,701),NTEST
000574      431 NOB=NOB+1
000606      GO TO 407
000606      428 CALL TRNGEN(DATA,NVARI,NTG,NSAM,LEF,LCASE,NF,JUMP,NA,BN,M,II)
000622      IF(LCASE) 409,407,407
000624      409 LCASE=0
000625      GO TO 600
000626      429 CALL TRNGEN(DATA,NVARI,NTG,NSAM,LEF,LCASE,NF,JUMP,NA,BN,M,II)
000642      IF(LCASE) 409,433,433
000644      433 CALL COOL(NSUB,REL,CUN,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N
      ITAPE)
      GO TO (600,431,999,701),NTEST
000661      430 CALL COOL(NSUB,REL,CUN,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N
      ITAPE)
      GO TO (600,435,999,701),NTEST
000706      435 NOB=NOB+1
000716      CALL TRNGEN(DATA,NVARI,NTG,NSAM,LEF,LCASE,NF,JUMP,NA,BN,M,II)
000720      IF(LCASE) 409,407,407
000733      407 H=H+1.
000737      IF(H-1)13,101,13
000741      101 H=0.0
000742      GO TO 112
000743      13 H=H/(H-1.)
000746      112 DO 8 I=1,NVARI
000750      KL=KL+1
000752      IF(KL-255)1004,1004,1004
000754      1005 WRITE(      IT1)C
000761      KL=1
000762      1004 C(KL)=DATA(I)
000765      AMIN(I)=AMIN1(AMIN(I),DATA(I))
000771      AMAX(I)=AMAX1(AMAX(I),DATA(I))
000775      SX(I)=SX(I)+DATA(I)
000777      Q(I)=DATA(I)-SX(I)/H
001003      QQ=Q(I)*HH
001005      DO 8 J=1,I
001006      8 SXY(I,J)=SXY(I,J)+Q(I)*QQ
001021      600 CONTINUE
***** WARNING -- ORIGINAL PROGRAM HAD DIVIDE CHECK, ACCUMULATOR OVERFLOW
***** QUOTIENT OVERFLOW TEST AT THIS POINT. CORRECT HAS REPLACED IT W
***** JUMP TO SECOND STATEMENT NUMBER, THUS IGNORING THE TEST.
001024      GO TO 1011
001024      1011 DO 1012 I=1,NVARI
001026      SX2(I)=SXY(I,I)
001032      DO 1012 J=1,I
001033      1012 SXY(J,I)=SXY(I,J)
001047      WRITE(      IT1)C
001054      GO TO (508,505,505,407,407),JESUS
001065      505 NSAM=NOB

```

```

RMD2000121
RMD2000122
RMD2000123
RMD2000124
RMD2000125
RMD2000126
RMD2000127
RMD2000128
RMD2000129
RMD2000130
RMD2000131
RMD2000132
RMD2000133
RMD2000134
RMD2000135
RMD2000136
RMD2000137
RMD2000138
RMD2000139
RMD2000140
RMD2000141
RMD2000142
RMD2000143
RMD2000144
RMD2000145
RMD2000146
RMD2000147
RMD2000148
RMD2000149
RMD2000150
RMD2000151
RMD2000152
RMD2000153
RMD2000154
RMD2000155
RMD2000156
RMD2000157
RMD2000158
RMD2000159
RMD2000160
RMD2000161
RMD2000162
RMD2000163
RMD2000164
RMD2000165
RMD2000166
RMD2000167
RMD2000168
RMD2000169
RMD2000170
RMD2000171
RMD2000172
RMD2000173
RMD2000174
RMD2000175
RMD2000176
RMD2000177
RMD2000178
RMD2000179
RMD2000180
RMD2000181
RMD2000182
RMD2000183

```



```

0001+07      639 FORMAT(46)
0001+07      642 FORMAT(F6.0)
0001+07      643 FORMAT(2X,F3.0)
0001+07      647 FORMAT(2H 12,18,214,114.4)
0001+07      502 FORMAT(23HDEFINING SAMPLE SIZE=1E)
0001+07      503 FORMAT(18HEND CASES ACCEPTED)
0001+07      704 FORMAT(46H CONTROL CARDS INCORRECTLY ORDERED OR PUNCHED.//33H ERRO
          1R FOUND AT STATEMENT NUMBER ,14,6A,B-PLUS .14//26H IN MAIN PROGRAM
          2M OF #4002D)
0001+07      400L FORMAT(115HO THE STANDARD DEVIATIONS, VARIANCE-COVARIANCE MATRIX AN
          XD CORRELATION MATRIX ARE UNDEFINED FOR A SAMPLE SIZE OF ONE.)
0001+07      400I FORMAT(97HINCORRECT TRANSGENERATION CODE ON CARD ABOVE. PROGRAM W
          ALL PROCEED WITHOUT THIS TRANSGENERATION.)
0001+07      1700 FORMAT(13G,AZ)
C
0001+07      END

```

PROGRAM LENGTH INCLUDING I/O BUFFERS
011070

0204

FUNCTION ASSIGNMENTS

STATEMENT	ASSIGNMENTS	4	5	6
-	000203	000503	15	001530
-	000604	001006	10	001160
-	000604	001277	50	001206
-	001262	001270	77	000531
-	000741	001576	103	001603
-	001610	001613	108	001620
-	001527	001634	112	000746
-	001241	001315	350	001325
-	000222	001543	404	001650
-	001564	000735	409	000624
-	000441	001567	414	001674
-	000370	001704	418	000445
-	000450	000452	423	000462
-	000657	000557	428	000606
-	000671	000604	433	000644
-	000436	000376	439	001711
-	001715	001720	500	001105
-	001724	001831	505	001065
-	001071	001074	600	001021
-	001113	000137	701	000155
-	001735	000176	706	000201
-	001545	000426	711	000413
-	001552	000145	742	000147
-	000133	000134	999	000073
-	000157	000220	1004	000762
-	000194	001024	1012	001033
-	000373	000274	2005	000301
-	001772	001332	9001	001335
-	001343	001343	9005	001351
-	001357	001342	9009	001365
-	001372	001557		

BLOCK NAMES AND LENGTHS
= 043417

VARIABLE ASSIGNMENTS

AMAX	=	002412	A4IN	=	002640	ASK	=	002075	A1	=	002102
A123	=	002076	AN	=	005271	R123	=	002077	C	=	005673
CODE	=	002104	CON	=	005517	C123	=	002100	DATA	=	004141
D123	=	002101	F4I	=	002215	FN	=	006552	H	=	006545
HH	=	006550	I	=	006532	I4ATA	=	003461C01	IT	=	006546
INDEX	=	005627	IT1	=	006521	J	=	006543	JECUS	=	006535
JUMP	=	004615	KK	=	006533	KA1	=	006534	KL	=	006544
KVM	=	006531	L4ASE	=	006540	LEFT	=	006541	M	=	006537
NTAPE	=	006530	NA	=	005043	NADII	=	006525	N4NJL	=	006526
NEW	=	004367	NI	=	006554	NJ	=	006555	N00	=	006536
NPAGE	=	006553	NS	=	003275	N344	=	006523	NSCL	=	006524
NSUB	=	005563	NTAPE	=	006520	NTEST	=	006547	NTC	=	006527
NVAR	=	006522	NVARI	=	006542	OP	=	002151	PC7	=	002406
PQ1	=	002407	PW2	=	002410	PW3	=	002411	PJc	=	002405
Q	=	006272	Q4	=	006551	REL	=	002105	SK	=	003465
SKY	=	000000C01	SK2	=	003713	SYM	=	003105	Time	=	002103
WK	=	002065	YY	=	003066						

START OF CONSTANTS
001411

START OF TEMPORARIES
002011

START OF INDIRECTS
002051

UNUSED COMPILER SPACE
014700

```

SUBROUTINE COOL (NSUB,REL,CON,OP,INDEX,DATA,NTEST,WK,KK1,A123,D123,
10123,NTAPE)
CCOOL      SUBROUTINE COOL FOR BM0020
C          REWRITTEN BY DU BUTS      3-25-64
C
C  DEFINITION OF NTEST:
C  NTEST=1 IF CASE FAILS BOOLEAN TEST
C  NTEST=2 IF CASE SATISFIES BOOLEAN TEST
C  NTEST=3 IF NEW PROBLEM CARD IS DETECTED
C  NTEST=4 IF FINISH CARD IS DETECTED
C
000020      TYPE INTEGER WK,OP,REL,A123,H123,D123,X
000020      DIMENSION NSUB(36),REL(36),CON(36),OP(36),INDEX(36),WK( 8),DATA(15
10), SKY(135,135),IN(37),IDATA(14)
C
000020      COMMON SKY,IDATA
000020      DO 100 I=1,KK1
000021      IS=NSUB(I)
C
C  EXAMINE BOOLEAN EXPRESSION FOR G1, GE, LI, LE, FN, NF
C
000023      DO 55 J=1,6
000025      IF (REL(I)-WK(J)) 55,26,55
000031      55 CONTINUE
000033      GO TO 311
000034      26 IF (INDEX(I)) 27,27,28
000037      27 CC=CON(I)
000042      GO TO 29
000042      28 K=CON(I)
000045      CC=DATA(K)
000047      29 H=DATA(IS)-CC

```

```

000052 IF ((CC.EQ.0.).OR.(DATA(15).EQ.0.)) 2299,2300
000061 2300 A=ABS (B/CC)-2**(-35)
000071 GO TO(1,2,3,4,5,6),J
000103 2299 GO TO(11,12,13,14,15,16),J
000115 1 IF(A)50,50,11
000117 11 IF(H)50,50,20
000121 2 IF(A)20,20,12
000123 12 IF(H)50,20,20
000125 3 IF(A)50,50,13
000127 13 IF(H)20,50,50
000131 4 IF(A)20,20,14
000133 14 IF(H)20,20,50
000135 5 IF(A)20,20,15
000137 15 IF(H)50,20,50
000141 6 IF(A)50,50,16
000143 16 IF(H)20,50,20
000144 20 IN(I)=1
000147 GO TO 100
000147 50 IN(I)=0
000151 100 CONTINUE
000154 NTEST=IN(I)
000155 KK=KK1+1
000157 IF(KK)500,500,501

```

```

C
C EXAMINE BOOLEAN OPERATOR FOR OR/AN
C

```

```

000160 501 DO 200 I=1,KK
000162 IF(OP(I)=WK(7))222,191,222
000166 191 IF(NTEST) 199,199,321
000170 199 NTEST=IN(I+1)
000173 GO TO 200
000173 222 IF(OP(I)=WK(8))301,223,301
000177 223 NTEST=NTEST*IN(I+1)
000203 200 CONTINUE
000206 500 IF(NTEST)320,320,321
000210 321 NTEST=2
000212 GO TO 333
000212 320 NTEST=1
000214 GO TO 333

```

```

C
C ERROR LOOK FOR NEXT PROBLEM OR FINISH CARD
C
C A123=6HFINISH
C B123=6HPRUHLN
C D123=6HPLUTSL
C

```

```

000214 341 X=REL(I)
000217 GO TO 313
000217 301 X=OP(I)
000222 313 WRITE( 5,2000)X
000230 IF(NTAPE=5) 302,302,304
000237 302 J=NTAPE
000241 312 REAU(J,1000)(IDATA(K),K=1,14)
000253 IF(IDATA=D123)305,312,305
000261 305 IF(IDATA=B123)307,306,307
000263 307 IF(IDATA=A123)312,309,312
000266 306 NTEST=3
000270 GO TO 333
000270 309 NTEST=4
000272 GO TO 333
000272 304 RE=IND NTAPE
000274 J=5
000275 GO TO 312

```

```

RM02000336
RM02000337
RM02000338
RM02000339
RM02000340
RM02000341
RM02000342
RM02000343
RM02000344
RM02000345
RM02000346
RM02000347
RM02000348
RM02000349
RM02000350
RM02000351
RM02000352
RM02000353
RM02000354
RM02000355
RM02000356
RM02000357
RM02000358
RM02000359
RM02000360
RM02000361
RM02000362
RM02000363
RM02000364
RM02000365
RM02000366
RM02000367
RM02000368
RM02000369
RM02000370
RM02000371
RM02000372
RM02000373
RM02000374
RM02000375
RM02000376
RM02000377
RM02000378
RM02000379
RM02000380
RM02000381
RM02000382
RM02000383
RM02000384
RM02000385
RM02000386
RM02000387
RM02000388
RM02000389
RM02000390
RM02000391
RM02000392
RM02000393
RM02000394
RM02000395
RM02000396
RM02000397

```

```

000276      333 RETURN
000277      1000 FORMAT(13A6,42)
000277      2000 FORMAT(31H1ILLEGAL OPERATOR OR RELATION IN CASE SELECTION
IN CARD. PROGRAM SKIPPED TO NEXT PROBLEM.)
000277      END

```

```

BM02000399
BM02000399
BM02000400
BM02000401
BM02000402

```

```

SUBPROGRAM LENGTH
000410

```

COOL

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	=	000115	2	=	000121	3	=	000125	4	=	000131
5	=	000135	6	=	000141	11	=	000117	12	=	000123
13	=	000127	14	=	000133	15	=	000137	16	=	000143
20	=	000144	25	=	000034	27	=	000037	28	=	000042
29	=	000047	50	=	000147	55	=	000031	100	=	000151
191	=	000156	199	=	000170	200	=	000203	222	=	000173
223	=	000177	301	=	000217	302	=	000237	304	=	000272
305	=	000251	306	=	000255	307	=	000253	308	=	000270
311	=	000214	312	=	000241	313	=	000222	320	=	000212
321	=	000210	333	=	000275	500	=	000205	501	=	000160
1500	=	000303	2000	=	000305	2200	=	000103	2300	=	000051

```

BLOCK NAMES AND LENGTHS
- 0434/7

```

VARIABLE ASSIGNMENTS

A	=	000405	A123	=	000003	B	=	000405	B123	=	000004
CC	=	000403	D123	=	000005	I	=	000400	DATA	=	043461001
IN	=	000333	IS	=	000401	J	=	000402	K	=	000404
KK	=	000407	KK1	=	000002	NTAPE	=	000005	NEST	=	000000
SXY	=	000000001	WK	=	000001	X	=	000332			

```

START OF CONSTANTS
000301

```

```

START OF TEMPORARIES
000322

```

```

START OF INDIRECTS
000330

```

```

UNUSED COMPILER SPACE
021300

```

```

SUBROUTINE CONV(CC, SYM, KK)
CCONV SUBROUTINE CONV FOR BM0020 (3600 FORTRAN VERSION)
DIMENSION CL(13005), SYM(65), CHAR(6)
TYPE INTEGER CC, SYM, CHAR
DO 2 I = 1, KK
DECODE (6,10,CC(I)) (CHAR(J),J=1,6)
10 FORMAT (6H1)
DO 1 J = 1, 6
CHAR(J)=SYM(65-CHAR(J))
1 CONTINUE
2 ENCODE (6,20,CC(I)) (CHAR(J),J=1,6)
20 FORMAT (6A1)
RETURN
END

```

```

BM02000403
BM02000404
BM02000405
BM02000406
BM02000407
BM02000408
BM02000409
BM02000410
BM02000411
BM02000412
BM02000413
BM02000414
BM02000415
BM02000416

```

```

SUBPROGRAM LENGTH
000112

```

CONV

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

10 = 000071 20 = 000073

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

CHAR = 000102 I = 000110 J = 000111

START OF CONSTANTS

000070

START OF TEMPORARIES

000075

START OF INDIRECTS

000100

UNUSED COMPILED SPACE

022400

NOT REPRODUCIBLE

```

SUBROUTINE PATTY(A,I)
CPATTY SUBROUTINE PATTY FOR BN0020 RECOMPIATION DATE 4-26-63
C
000005 DIMENSION A(135,135),NN(4)
C
000005 IT=1
000006 KK=0
000007 K1=IT
000010 K2= MIN0 (8,N)
000013 5 KK=KK+8
000015 IF(N-KK)3,3,+
000017 * IT=IT+1
000021 GO TO 5
000021 3 DO 50 JA=1,IT
000023 LLL=K2-K1+1
000026 LL=0
000027 DO 40 JJ=K1,K2
000030 LL=LL+1
000032 40 NN(LL)=JJ
000036 WRITE( 6,300) (NN(I),I=1,LLL)
000050 DO 10 I=1,N
000053 10 WRITE( 6,20) I,(A(I,J),J=K1,K2)
000076 K1=K2+1
000100 K2=K1+7
000101 K2= MIN0 (K2,N)
000104 20 FORMAT(1H I3,F11.4,7F14.4)
000104 300 FORMAT(1H08X,4HCOL.7(10X,4HCOL.),/BX.13.7( 11X.13),/4H ROW//)
000104 50 CONTINUE
000107 RETURN
000107 END

```

SUBPROGRAM LENGTH

000157

FUNCTION ASSIGNMENTS

3 - 000021 4 - 000017 5 - 000013 20 - 000113
300 - 000117

VARIABLE ASSIGNMENTS

I	-	000155	II	-	000154	II	-	000144	J	-	000156
JJ	-	000153	JX	-	000150	KK	-	000145	KJ	-	000146
KZ	-	000147	LL	-	000152	LLL	-	000151	VV	-	000134

000111

000126

000133

022300

NOT REPRODUCIBLE

24

```

000062      CM(1)=7700000000000000000H
000063      CM(2)=0077000000000000000H
000065      CM(3)=0000770000000000000H
000066      CM(4)=0000007700000000000H
000070      CM(5)=0000000077000000000H
000071      CM(6)=0000000000770000000H
000073      C1=6HAAAAAA
000074      C4=(+6H+++++)
000076      DO 31 I=1,6
000077      31 CJ(I)=C1.AND.CM(I)
000105          IT=4
000106          ENCODE(66,400,RUF) A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
000141          DECODE(65,401,RUF) (SYM(I),I=1,65)
000160              K=(K-1)/6+1
000164      400 FORMAT(11A6)
000164      401 FORMAT(72A1)
000164      94 REWIND IT
000166          K=0
000167          DO 570 II=1,NPL
000174          READ(5,500) (X(I),I=1,M)
000205      5000 FORMAT (8A10)
000205          DECODE(72,2,X)Z,LL,N,(MM(I),I=1,20)
000232      2    FORMAT(A6,I3,I2,2G13)
000232          IF(LL*(NV1-LL))3,5,77
000241      77 IF(Z-PL)3,5,3
000243      3 WRITE(        6,10)II,(X(I),I=1,12)
000257      10  FORMAT(24H0EKKRW ON SELECTION CARD13,5X,12A4)
000257          GO TO 570
000263      5    DO 6 I=1,20
000265          IF((NV-MM(I))*MM(I))3,7,7
000271      7    IF(MM(I))770,770,6
000274      8    K=K+1
000276          IX(K)=MM(I)
000301      6    IY(K)=LL
000305      770 IF((NPL-II)*(K-60))<70,<00,>600
000312      600 KN=K
000314          L1=0
000315      510 L0=L1+1
000317          L1= MINO (L1+15,KN)
000323          K=0
000324          DO 500 L=L0,L1
000326          K=K+1
000330          IX(K)=IX(L)
000333      500 IY(K)=IY(L)
000337      9    KK=M67*K
000342          DO 32 N=1,KK
000343      32   IC(N)=0
000347          KL=255
000350          DO 13 JJ=1,N0
000352      301 DO 12 J=1,NV
000354          KL=KL+1
000356          IF(KL-255)12,12,14
000360      14  READ(        II)CIC
000365          KL=1
000366      12  X(J)=CIC(KL)
000377      302 DO 13 J=1,K
000401          L=IX(J)
000403          M=IY(J)
000405          L=51.5-(X(L)-AMIN(L))/R(L)*50.
000415          IF (52-L)13,13,70
000417      70  M=(X(M)-AMIN(M))/R(M)*100.+1.5
000427          IF ((102-M)*M)13,13,71
000432      71  M1= MOD (M-1,6)+1
000440          M2=(M+5)/6

```

```

000444      M3=L+51*(M2+17*J-19)
000452      IF(M)-1) 60,60,61
000455      60  MUG=CM4.AND..NOT.CC(M3)
000460      IF(MUG)61,63,61
000461      63  IF(CC(M3))13,64,64
000464      64  CC(M3)=CC(M3).AND..NOT.CM.UR.CCC
000471      GO TO 13
000471      61  MUG=CM(M1).AND..NOT.CC(M3)
000476      IF(MUG)16,13,16
000477      16  IC(M3)=ISIGN (IAHS (IC(M3))+JC(M1)+IC(M3))
000506      13  CONTINUE
000513      /01  RE*IND IT
000515      CALL CONV(CC,SYM,KK)
000520      P=(+M.)
000522      DO 40 N=1,101
000526      40  C(N)=P
000533      DO 41 N=1,101,5
000534      41  C(N)=C4
000541      DO 50 J=1,K
000542      L=IX(J)
000544      M=IY(J)
000546      Q=AMIN(M)
000550      D=R(M)/10.
000552      DO 51 N=1,11
000553      I(N)=Q
000555      51  Q=Q+D
000560      Q=AMIN(L)+R(L)
000564      D=R(L)/50.
000566      DO 52 N=1,51
000567      X(N)=Q
000571      52  Q=Q-D
000574      50  WRITE(
           6,54)M,L,(I(N),N=1,11,2),(I(N),N=2,10,2),(C(N),N
           1=1,101),(X(K00),C(K00),(DD(N,J),N=K00,867,5))C(K00),      K00=1.
           251),(C(N),N=1,101),(I(N),N=1,11,2),(I(N),N=2,10,2)
000702      54  FORMAT(11H1  VARIAB E+RX,8HVARIA BLE[3/17/2X,F15.3,5F20.3/7X,5F60.3
           1/13X,101A1.51(/1X,F10.3,1X,A1,1A46,A5,A1      )/13X,101A1/2X,F15.3
           2,5F20.3/7X,5F20.3)
000702      IF(L1-KN)510,580,580
000704      580  K=0
000705      570  CONTINUE
000710      900  RETURN
000711      END

```

9M02000546
 9M02000547
 9M02000548
 9M02000549
 9M02000550
 9M02000551
 9M02000552
 9M02000553
 9M02000554
 9M02000555
 9M02000556
 9M02000557
 9M02000558
 9M02000559
 9M02000560
 9M02000561
 9M02000562
 9M02000563
 9M02000564
 9M02000565
 9M02000566
 9M02000567
 9M02000568
 9M02000569
 9M02000570
 9M02000571
 9M02000572
 9M02000573
 9M02000574
 9M02000575
 9M02000576
 9M02000577
 9M02000578
 9M02000579
 9M02000580
 9M02000581
 9M02000582
 9M02000583
 9M02000584
 9M02000585
 9M02000586
 9M02000587

SUBPROGRAM LENGTH

001137

PLU1

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

2	=	000770	3	=	000243	5	=	000263	7	=	000271
8	=	000274	9	=	000337	10	=	000770	12	=	000366
13	=	000506	14	=	000360	18	=	000477	33	=	000100
32	=	000344	40	=	000527	41	=	000535	56	=	001012
60	=	000455	61	=	000471	63	=	000461	64	=	000464
70	=	000417	71	=	000432	77	=	000241	94	=	000164
301	=	000352	302	=	000377	400	=	000761	401	=	000763
510	=	000315	570	=	000705	580	=	000704	600	=	000312
701	=	000513	770	=	000305	900	=	000710	5000	=	000766

BLOCK NAMES AND LENGTHS

= 0327/2

A1	-	001053	A10	-	001064	A11	-	001065	A2	-	001054
A3	-	001055	A4	-	001072	A5	-	001057	A6	-	001060
A7	-	001061	A8	-	001062	A9	-	001063	A0	-	001073
C	-	031543C01	CC	-	000000C01	CLC	-	001105	CTR	-	031543C01
CJ	-	032756C01	CM	-	032764C01	CMM	-	001104	C1	-	001066
C4	-	001057	D	-	001135	DU	-	000000C01	MD	-	001121
I	-	001107	IC	-	000000C01	II	-	001114	IT	-	001112
IX	-	032142C01	IY	-	032370C01	J	-	001126	JC	-	032756C01
JJ	-	001125	JJJJ	-	001110	K	-	001113	KK	-	001123
KL	-	001124	KN	-	001117	K00	-	001136	L	-	001122
LL	-	001115	LO	-	001121	LI	-	001120	M	-	001127
MM	-	032016C01	MUG	-	001133	MI	-	001130	M2	-	001131
M3	-	001132	N	-	001116	NV1	-	001106	P	-	001071
PL	-	001056	Q	-	001134	SYM	-	032645C01	I	-	032642C01
X	-	031315C01	Z	-	001070						

NOT REPRODUCIBLE

4M02000544
 4M02000545
 4M02000546
 4M02000547
 4M02000548
 4M02000549
 4M02000550
 4M02000551
 4M02000552
 4M02000553
 4M02000554
 4M02000555
 4M02000556
 4M02000557
 4M02000558
 4M02000559
 4M02000560
 4M02000561
 4M02000562
 4M02000563
 4M02000564
 4M02000565
 4M02000566
 4M02000567
 4M02000568
 4M02000569
 4M02000570
 4M02000571
 4M02000572
 4M02000573
 4M02000574
 4M02000575
 4M02000576
 4M02000577
 4M02000578
 4M02000579
 4M02000580
 4M02000581
 4M02000582
 4M02000583
 4M02000584
 4M02000585
 4M02000586
 4M02000587
 4M02000588
 4M02000589
 4M02000590
 4M02000591
 4M02000592
 4M02000593
 4M02000594
 4M02000595
 4M02000596
 4M02000597
 4M02000598
 4M02000599
 4M02000600
 4M02000601
 4M02000602
 4M02000603
 4M02000604
 4M02000605
 4M02000606
 4M02000607
 4M02000608
 4M02000609
 4M02000610
 4M02000611
 4M02000612
 4M02000613
 4M02000614
 4M02000615
 4M02000616
 4M02000617
 4M02000618
 4M02000619
 4M02000620
 4M02000621
 4M02000622
 4M02000623
 4M02000624
 4M02000625
 4M02000626
 4M02000627
 4M02000628
 4M02000629
 4M02000630
 4M02000631
 4M02000632
 4M02000633
 4M02000634
 4M02000635
 4M02000636
 4M02000637
 4M02000638
 4M02000639
 4M02000640
 4M02000641
 4M02000642
 4M02000643
 4M02000644
 4M02000645
 4M02000646
 4M02000647
 4M02000648
 4M02000649
 4M02000650
 4M02000651
 4M02000652
 4M02000653
 4M02000654
 4M02000655
 4M02000656
 4M02000657
 4M02000658
 4M02000659
 4M02000660
 4M02000661
 4M02000662
 4M02000663
 4M02000664
 4M02000665
 4M02000666
 4M02000667
 4M02000668
 4M02000669
 4M02000670
 4M02000671
 4M02000672
 4M02000673
 4M02000674
 4M02000675
 4M02000676
 4M02000677
 4M02000678
 4M02000679
 4M02000680
 4M02000681
 4M02000682
 4M02000683
 4M02000684
 4M02000685
 4M02000686
 4M02000687
 4M02000688
 4M02000689
 4M02000690
 4M02000691
 4M02000692
 4M02000693
 4M02000694
 4M02000695
 4M02000696
 4M02000697
 4M02000698
 4M02000699
 4M02000700
 4M02000701
 4M02000702
 4M02000703
 4M02000704
 4M02000705
 4M02000706
 4M02000707
 4M02000708
 4M02000709
 4M02000710
 4M02000711
 4M02000712
 4M02000713
 4M02000714
 4M02000715
 4M02000716
 4M02000717
 4M02000718
 4M02000719
 4M02000720
 4M02000721
 4M02000722
 4M02000723
 4M02000724
 4M02000725
 4M02000726
 4M02000727
 4M02000728
 4M02000729
 4M02000730
 4M02000731
 4M02000732
 4M02000733
 4M02000734
 4M02000735
 4M02000736
 4M02000737
 4M02000738
 4M02000739
 4M02000740
 4M02000741
 4M02000742
 4M02000743
 4M02000744
 4M02000745
 4M02000746
 4M02000747
 4M02000748
 4M02000749
 4M02000750
 4M02000751
 4M02000752
 4M02000753
 4M02000754
 4M02000755
 4M02000756
 4M02000757
 4M02000758
 4M02000759
 4M02000760
 4M02000761
 4M02000762
 4M02000763
 4M02000764
 4M02000765
 4M02000766
 4M02000767
 4M02000768
 4M02000769
 4M02000770
 4M02000771
 4M02000772
 4M02000773
 4M02000774
 4M02000775
 4M02000776
 4M02000777
 4M02000778
 4M02000779
 4M02000780
 4M02000781
 4M02000782
 4M02000783
 4M02000784
 4M02000785
 4M02000786
 4M02000787
 4M02000788
 4M02000789
 4M02000790
 4M02000791
 4M02000792
 4M02000793
 4M02000794
 4M02000795
 4M02000796
 4M02000797
 4M02000798
 4M02000799
 4M02000800
 4M02000801
 4M02000802
 4M02000803
 4M02000804
 4M02000805
 4M02000806
 4M02000807
 4M02000808
 4M02000809
 4M02000810
 4M02000811
 4M02000812
 4M02000813
 4M02000814
 4M02000815
 4M02000816

000161	30 IF(D)99,99,14	RM02000622
000163	14 DATA(NEWA)=ALOG(D)*.434294*819	RM02000623
000174	GO TO 3	RM02000624
000174	40 DATA(NEWA)=EXP (D)	RM02000625
000204	GO TO 3	RM02000626
000205	50 IF(U)99,7,17	RM02000627
000207	17 IF(U-1.0)18,19,99	RM02000628
000212	19 DATA(NEWA)=3.14159265/2.0	RM02000629
000215	GO TO 3	RM02000630
000215	18 A=SQRT (D)	RM02000631
000220	DATA(NEWA)=ASNF(A)	RM02000632
000230	GO TO 3	RM02000633
000230	60 A=D/(SAMP+1.0)	RM02000634
000233	B=A+1.0/(SAMP+1.0)	RM02000635
000237	IF(A)99,23,24	RM02000636
000240	23 IF(B)99,7,27	RM02000637
000242	27 DATA(NEWA)=ASNF(SQRT (B))	RM02000638
000255	GO TO 3	RM02000639
000255	24 IF(B)99,28,29	RM02000640
000257	28 DATA(NEWA)=ASNF(SQRT (A))	RM02000641
000272	GO TO 3	RM02000642
000272	29 A=SQRT (A)	RM02000643
000275	B=SQRT (B)	RM02000644
000277	DATA(NEWA)=ASNF(A)+ASNF(B)	RM02000645
000313	GO TO 3	RM02000646
000313	70 IF(D)31,99,31	RM02000647
000314	31 DATA(NEWA)=1.0/D	RM02000648
000317	GO TO 3	RM02000649
000320	80 DATA(NEWA)=U+BNEM	RM02000650
000324	GO TO 3	RM02000651
000324	90 DATA(NEWA)=U+BNEM	RM02000652
000327	GO TO 3	RM02000653
000330	100 IF(D)33,7,33	RM02000654
000331	33 DATA(NEWA)=U**BNEM	RM02000655
000336	GO TO 3	RM02000656
000337	110 DATA(NEWA)=U+DATA(NFNB)	RM02000657
000344	GO TO 3	RM02000658
000344	120 DATA(NEWA)=U-DATA(NFNB)	RM02000659
000351	GO TO 3	RM02000660
000351	130 DATA(NEWA)=U*DATA(NFNB)	RM02000661
000356	GO TO 3	RM02000662
000356	140 IF(DATA(NEWB))34,99,34	RM02000663
000360	34 DATA(NEWA)=U/DATA(NFNB)	RM02000664
000365	GO TO 3	RM02000665
000365	150 HNE=NEWB	RM02000666
000367	IF(U-HNE)7,11,11	RM02000667
000372	160 IF(U=DATA(NEWB))7,11,11	RM02000668
000376	170 IF(U)3,503,3	RM02000669
000377	503 IF(SIGN (10.0*D))504,3,3	RM02000670
000403	504 DATA(NEWA)=HNE	RM02000671
000406	3 CONTINUE	RM02000672
000411	GO TO 42	RM02000673
000411	99 LCASE=-999	RM02000674
000412	IF(MERRY-J)402,401,402	RM02000675
000415	402 MERRY=J	RM02000676
000417	WRITE(6,1404)J	RM02000677
000425	401 WRITE(6,1405)ITEM	RM02000678
000433	WRITE(6,1406)	RM02000679
000437	ISAMP=ISAMP-1	RM02000680
000444	42 RETURN	RM02000681
000445	1404 FORMAT(30H)THE INSTRUCTIONS INDICATED ON/254 TRANS GENERATOR CARD	RM02000682
	INO.12,4M RE-/29M SULTED IN THE VIOLATION OF A/31M RESTRICTION FOR	RM02000683
	2THIS TRANSFER-/31M MATION. THE VIOLATION OCCURRED/2TH FOR THE CASE	RM02000684
	3 LISTED BELOW./)	RM02000685

000445 1405 FORMAT(9H CASE NO.15)
 000445 1406 FORMAT(45H0THIS CASE WILL BE DELETED FOR ALL VARIABLES)

RM02000686
 RM02000687
 RM02000688
 RM02000689

000445 C END

SUBPROGRAM LENGTH
 000574
 TANGEN

FUNCTION ASSIGNMENTS
 ASAF - 000021

STATEMENT ASSIGNMENTS

3	-	000406	4	-	000064	5	-	000062	7	-	000117
8	-	000122	10	-	000115	11	-	000135	12	-	000140
14	-	000163	17	-	000207	18	-	000215	19	-	000212
20	-	000133	23	-	000240	24	-	000255	27	-	000242
28	-	000257	29	-	000272	30	-	000161	31	-	000314
33	-	000331	34	-	000360	40	-	000174	42	-	000444
50	-	000205	60	-	000230	70	-	000313	80	-	000320
90	-	000324	99	-	000411	100	-	000330	110	-	000337
120	-	000344	130	-	000351	140	-	000356	150	-	000365
160	-	000372	170	-	000376	305	-	000043	310	-	000051
315	-	000057	401	-	000425	402	-	000415	500	-	000071
503	-	000377	504	-	000403	1404	-	000457	1405	-	000505
1408	-	000510									

BLOCK NAMES AND LENGTHS
 - 043461

VARIABLE ASSIGNMENTS

A	-	000572	H	-	000573	BBNEW	-	000003	BNEW	-	000567
D	-	000571	ITEM	-	000561	J	-	000563	LCODE	-	000565
LLCODE	-	000001	LLVA	-	000002	LVA	-	000566	MEORY	-	000004
N	-	000005	NEWA	-	000564	NEWH	-	000570	NNEWA	-	000000
SAMP	-	000562	VECTOR	-	000000001						

START OF CONSTANTS
 000447

START OF TEMPORARIES
 000517

START OF INDIRECTS
 000553

UNUSED COMPILER SPACE
 020600

SUBROUTINE VFCMCK(NVF)
 CVCFCMCK SUBROUTINE TO CHECK FOR PROPER NUMBER OF VARIABLE FORMAT CDS
 000003 IF(NVF)10,10,20
 000004 10 WRITE(A,4000)
 000010 NVF=1
 000012 50 RETURN
 C
 000013 20 IF(NVF-10)50,50,10
 C
 000016 4000 FORMAT(1H023X7)NUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF
 AIED, ASSUMED TO BE 1.)
 000016 END

RM02000690
 RM02000691
 RM02000692
 RM02000693
 RM02000694
 RM02000695
 RM02000696
 RM02000697
 RM02000698
 RM02000699
 RM02000700
 RM02000701

SUBPROGRAM LENGTH
 000036

NOT REPRODUCIBLE

0-23698-0

1

SUBROUTINE SCALE(YMIN,YMAX,YINT,JY,TYMIN,TYMAX,YIJJ)
 SCALES
 SUBROUTINE SCALE FOR BM0022 AUGUS

201	20	5
-----	----	---

SUPPERS
900241

**PREFACE TO THE CDC 6500
STATISTICAL PROGRAMS**

PURDUE UNIVERSITY

IA. Tape Input

Most of the statistical programs have provision for input of data prepared on tape by means of an Alternate Input Tape. Use of tapes for data provides a very compact storage of data files. The tape for one data file may be reread several times to allow several analyses from the same data file by rewinding the Alternate Input Tape. Of course, data input may be from data cards if preferred.

IB. Standard Data Input

The form of Standard Data Input is given below:

		Variables				
		x_1	x_2	x_3	$\dots$	x_p
1		x_{11}	x_{12}	x_{13}	$\dots$	x_{1p}
2		x_{21}	x_{22}	x_{23}	$\dots$	x_{2p}
Cases 3		x_{31}	x_{32}	x_{33}	$\dots$	x_{3p}
		.	.	.		.
		.	.	.		.
n		x_{n1}	x_{n2}	x_{n3}	$\dots$	x_{np}

The headings $x_1, x_2, \dots, x_p$ represent variables, e.g., age, sex, weight, etc. Each row in the table represents a set of corresponding values of these variables, e.g., the age, sex, weight, etc. of a given individual. The entries x_{ij} in the table are called data values, the whole array of these numeric values is called the data matrix, each row of the data matrix is called a case, and each column is called a variable.

The Standard Data Input is keypunched case-wise. That is, all the data values of the first case are keypunched in order on one or more cards. Then starting on a new card the second case is punched, etc. Each case must have the same format (see Section II-B). This means that from one case to the

next, each variable value must occupy the same physical location on the card into which it is punched.

In general, not all of the fields on a card will be considered as part of the data matrix. For example, identification fields such as the subject number are seldom included. The desired fields are selected by means of the Variable Format Card specification (see Section II-B).

II. Preparation of Program Control Cards

The statistical programs are written in a general form so that a wide variety of problems combined with optional computations may be handled by each program. The user specifies certain parameter values, optional computations and optional output, the form of the data input, etc. on Program Control Cards. Standard program control cards which are used in many programs are described in this section. Instructions for the preparation of other control cards specifically required for an individual program appear in the program description.

Unless otherwise stated, each numeric field of a control card should be punched without a decimal point. The decimal point is assumed to be at the extreme right of the field, and blank columns will be interpreted as zeros. Thus, for example, a field including Columns 1-6 (1 through 6) which contains only a 1 in Column 5 will be interpreted to mean 10. In general, on control cards numbers should be punched in the rightmost columns of the field. This is called right-justification.

IIA. Transgeneration Cards

The term transgeneration is used to include transformations of input variables and creation of new variables prior to the normal computations performed by the various programs.

The transformations described below are performed on the values of the variables in each case. In these examples, the symbol x_i will denote the i^{th} variable as well as its value.

Examples:

$$\log_{10} X_4 \rightarrow X_4$$

$\log_{10} X_4$ replaces X_4

$$X_5^c \rightarrow X_1$$

X_5^c replaces X_1

$$X_2 + X_3 \rightarrow X_2$$

$X_2 + X_3$ replaces X_2

By successive transformations, more complicated relationships may be obtained. For example:

(i) To replace X_5 by $\sqrt{X_1^2 + X_3^2}$ four transformations are required:

Variables as they are stored at each step

Transformation	X_1	X_2	X_3	X_4	X_5
$X_1^2 \rightarrow X_1$	X_1^2	X_2	X_3	X_4	X_5
$X_3^2 \rightarrow X_3$	X_1^2	X_2	X_3^2	X_4	X_5
$X_1 + X_3 \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$X_1^2 + X_3^2$
$\sqrt{X_5} \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$\sqrt{X_1^2 + X_3^2}$

In this example, it can be seen that the original values of X_5 are irrelevant. Actually the variable X_5 may be a dummy variable introduced by the program specifically to provide capacity for creating new variables by transgeneration. Dummy variables may be required for intermediate storage in order to effect some transformations.

- (ii) To replace X_1 by $\exp(-1/2 X_1^2)$ three transformations are required:

Transformation	X_1	X_2	X_3
$X_1^2 \rightarrow X_1$	X_1^2	X_2	X_3
$-1/2 X_1 \rightarrow X_1$	$-1/2 X_1^2$	X_2	X_3
$\exp(X_1) \rightarrow X_1$	$\exp(-1/2 X_1^2)$	X_2	X_3

- (iii) To replace X_4 by $X_2 + \log_{10}(X_4 - X_3 + 100)$ four transformations are required:

Transformation	X_1	X_2	X_3	X_4
$X_4 - X_3 \rightarrow X_4$	X_1	X_2	X_3	$X_4 - X_3$
$X_4 + 100 \rightarrow X_4$	X_1	X_2	X_3	$(X_4 - X_3 + 100)$
$\log_{10} X_4 \rightarrow X_4$	X_1	X_2	X_3	$\log_{10}(X_4 - X_3 + 100)$
$X_2 + X_4 \rightarrow X_4$	X_1	X_2	X_3	$X_2 + \log_{10}(X_4 - X_3 + 100)$

The transformations are performed in the order in which the Transgeneration Cards appear, so that, for example, the two transgenerations $2X_1 \rightarrow X_1$ followed by $X_1 - 2 \rightarrow X_1$ will result in $2X_1 - 2$, whereas $X_1 - 2 \rightarrow X_1$ followed by $2X_1 \rightarrow X_1$ will result in $2(X_1 - 2)$.

TRANSGENERATION LIST

Notation to be used in the following transgeneration list:

i, j, k are variable indices (need not be different)

c is a constant

$a_1, a_2, a_3, \dots$ are constants

n is the number of cases, or sample size

The mean $\bar{X}_i = \frac{1}{n} \sum_{j=1}^n X_{ji}$

The standard deviation $s_i = \left[\frac{1}{n-1} \sum_{j=1}^n (X_{ji} - \bar{X}_i)^2 \right]^{1/2}$

<u>Code</u>	<u>Transgeneration</u>	<u>Restriction</u>
01	$\sqrt{X_i} \rightarrow X_k$	$X_i \geq 0$
02	$\sqrt{X_i} + \sqrt{X_i+1} \rightarrow X_k$	$X_i \geq 0$
03	$\log_{10} X_i \rightarrow X_k$	$X_i > 0$
04	$e^{X_i} \rightarrow X_k$	-
05	$\arcsin \sqrt{X_i} \rightarrow X_k$	$0 \leq X_i \leq 1$
06	$\arcsin \sqrt{X_i/(n+1)} + \arcsin \sqrt{(X_i+1)/(n+1)} \rightarrow X_k$	$0 \leq (X_i/n) \leq 1$
07	$1/X_i \rightarrow X_k$	$X_i \neq 0$
08	$X_i + c \rightarrow X_k$	-
09	$X_i^c \rightarrow X_k$	-
10	$X_i^c \rightarrow X_k$	$X_i \geq 0$
11	$X_i + X_j \rightarrow X_k$	-
12	$X_i - X_j \rightarrow X_k$	-
13	$X_i X_j \rightarrow X_k$	-
14	$X_i / X_j \rightarrow X_k$	$X_j \neq 0$
15	If $X_i \geq c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
16	If $X_i \geq X_j$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
17	$\log_e X_i \rightarrow X_k$	$X_i > 0$
18	$X_i - \bar{X}_i \rightarrow X_k$	-

<u>Code</u>	<u>Transgeneration</u>	<u>Restriction</u>
19	$X_i/s_i \rightarrow X_k$	-
20	$\sin X_i \rightarrow X_k$	-
21	$\cos X_i \rightarrow X_k$	-
22	$\arctan X_i \rightarrow X_k$	-
23	$X_i^{X_j} \rightarrow X_k$	$X_i > 0$
24	$c^{X_i} \rightarrow X_k$	$c > 0$
25	$X_i \rightarrow X_k$	-
26	$c \rightarrow X_k$	(Leave code i blank)
27-39 Not defined		
40	If $X_i = a_1$ or a_2 or $a_3 \dots, a_7$, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	
41	If X_i is blank, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	
	*Note that in reading numeric fields, a blank field and -0 are equivalent.	
42	If $X_i = a_1$ or a_2 or $a_3 \dots, a_7$, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	
43	If X_i is blank, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	

When a violation of a restriction in the right-hand column occurs during transgeneration, the program will print a diagnostic message. Most programs will proceed to the next problem, if any.

Some programs will delete the case where the violation occurred and continue the computation. Other programs will screen all the input data from additional restriction violations before proceeding to the next problem, if any.

1. Standard Transgeneration Cards

Standard Transgeneration Cards are used with programs which use Standard Data Input (see section IB). Let p denote the number of variables in the data matrix, m the maximum number of variables allowed by the program for any problem and q the number of variables added through transgeneration. Any of the variables $x_1, \dots, x_m$ may be used in transgeneration. The initial values of the first p variables are read from the input data file (Data Cards or Alternate Input Tape). The initial values of the remaining $m-p$ variables are left over from previous calculations. After transgeneration action of a particular case, the values of the first $p+q$ variables for that case are used as the values of the trans-generated variables. If the $p+q$ variables required for the computation are not the first $p+q$, they must be relocated. This may be done by using transgeneration code number 25. The numbers p and q (q may be positive, negative, or zero) are specified on the Problem Card. The indices i, j , and k from the transgeneration list may exceed p or $p+q$ but must never exceed m .

Card Preparation

Col. 1-6	TRNGEN	(Mandatory)
Col. 7-9	Variable index k	
Col. 10,11	Code from transgeneration list (restricted by availability in particular program)	
Col. 12-14	Variable index i	
Col. 15-20	Variable index j or constant c	
Col. 21-25	Blank	
Col. 26	Number of a_i 's for transformation 40 or 42	
Col. 27-32	a_1 value	
Col. 33-38	a_2 value	

...

Col. 63-68 a_7 value

The constants c , a_1 , ..., a_7 are punched with a decimal point if used with variables which have an F-type format and without a decimal point if used with variables which have an I-type format (see Section II-B).

The Standard Transgeneration Cards for the three samples on pages 3 and 4 are:

- (i) TRNGEN001100012.0000
TRNGEN003100032.0000
TRNGEN00511001000003
TRNGEN00501005000000
- (ii) TRNGEN001100012.0000
TRNGEN00109001-0.500
TRNGEN00104001000000
- (iii) TRNGEN00412004000003
TRNGEN00408004100.00
TRNGEN00403004000000
TRNGEN00411004000002

IIB. Variable Format Cards (for Input)

The word "format" usually refers to the arrangement of information keypunched on a card. The format of a data card is a sequence of fields (variables), each of which occupies one or more columns. For the computer programs a format is a set of specifications according to which information is read into the program from punched cards. The specifications tell the program which parts (or columns) of the card to skip, which parts to regard as all one number, and which parts to regard as several numbers in a row. For instance, it is the format which tells the program whether a card punched "345890021" is to be read in as "34.5, 890.0, .021", or "34., 9002.1", or "589.002", or "3, 4, 58, 90, 0, 2, 1", etc. It does this by giving the program a sequence of specifications

which indicate the size of a field and the method of handling that field (i.e., skipping it, entering it into the computer as a whole number, entering it into the computer as a number with two decimal digits, etc.).

The format also tells the program how to read in a certain set of cards when more than one are required to contain all the data for one case. In most programs the format describes the variables for each case. Each successive case is assumed to have the same format.

If the formats for the programs were fixed in advance, all data would have to be punched on cards in the same way for every study. Since this is not usually convenient, the statistical programs have been written so that the user may vary the formats according to his preference for a particular study. For this reason, they are referred to as "variable formats". The program is informed of the format which is to be used by Variable Format Cards. The user must specify on the Problem Card the number of cards used to keypunch the variable format.

In addition to providing an economical method of preparing data input cards (by defining fields to be as small as possible, or "packing" the data), the variable format permits considerable freedom in controlling data input. For instance:

- . It allows the user to select for each case only those cards which have fields of interest. (See Examples ii, iii, vi below.)
- . It allows the user to select only those fields of interest from among the fields of each card. (See Examples iv, v, vi.)
- . It allows the user to scale the data input, i.e., shift the decimal point.

A complete description of formats can be found in FORTRAN programming manuals such as those available from IBM representatives. The features commonly required for the statistical programs are described below.

1. F-type Variable Format

The F-type format is the most frequently used in the statistical programs. It is required when the decimal point is keypunched on the card or when the decimal point is to be placed by the program. All data input values must be signed (+) or unsigned numbers with or without a decimal point punched.

Specifications:

- (a) "nFw.d" F is the floating point (decimal) indicator; n is the number of fields of width w (includes sign and decimal point if punched); and d is the number of digits to the right of the decimal point if the decimal is not punched ($0 \leq d \leq w$). If the decimal is punched, d is ignored. If n is not specified, it is assumed to be 1.
- (b) "sPnFw.d" P is the scale indicator; s is a scale factor (explained below); and n, w and d are defined in specification (a).
- (c) "mX" (alphabetic X) X is the skip indicator, and m is the number of columns to be skipped.
- (d) "/" (slash) indicates "go to the next data card."

Depending on its location in the format statement, the "/" will either direct the program

to go immediately to the next card (ignoring any further information on the current card) or skip one card altogether. For example, if a format begins with "/", the program will automatically skip the first card, read the second, skip the third card, etc. If a format ends with "/", the program will automatically read the first card, skip the second, read the third card, etc. (See Examples ii, iii, v, vi.)

"//" indicates "go to the card after next."

Two slashes "/" will direct the program to skip two cards. Any number of slashes may be used.

The format is keypunched beginning with a left parenthesis, a sequence of specifications, and closed by a right parenthesis. Specifications (a), (b), and (c) are followed by a comma, except preceding a slash or right-hand parenthesis. Blank columns within the format are ignored. Columns 1-80 may be used unless otherwise specified.

Examples:

- (i) (12F3.0,F4.0,11F2.0), punched in the first 20 columns of the Variable Format Card, will describe 12 three-column fields followed by 1 four-column field, followed by 11 two-column fields. Each data card will be read according to this format.
- (ii) (12F3.0,F4.0,11F2.0 /), punched in the first 22 columns, will describe the same fields as in Example (i), but will also instruct the program to read the first card, skip the second, read the third card, skip the fourth, etc.

Two slashes at the end

...,11F2.0 //)

will instruct the program to read the first card, skip the next two cards, read the fourth card, skip the next two, read the seventh card, etc.

- (iii) (/12F3.0,F4.0,11F2.0), punched in the first 21 columns, will describe the same fields as in Example (i), but will also instruct the program to skip the first card, read the second, skip the third card, read the fourth, etc.

Two slashes at the beginning

(//12F3.0,...

will instruct the program to skip the first two cards, read the third, skip the next two, read the sixth card, skip the next two, etc.

- (iv) (10X,F6.0,2X,2F3.0), punched in the first 19 columns, will instruct the program to direct entry of data from each card as follows:

- (1) Skip 10 columns.
 - (2) Pick up a six-digit field in Col. 11-16.
 - (3) Skip 2 columns.
 - (4) Pick up 2 three-digit fields in Col. 19-21 and 22-24.
- (v) (5X,2F6.0,F1.0,3X,F5.0 /5X,F6.0), punched in the first 32 columns, will instruct the program to direct entry of data from each pair of cards as follows:
- (1) Skip 5 columns.
 - (2) Pick up 2 six-digit fields in Col. 6-11 and 12-17.
 - (3) Pick up a one-digit field in Col. 18.
 - (4) Skip three columns (Col. 19-21).
 - (5) Pick up a five-digit field in Col. 22-26.
 - (6) Go to second data card.
 - (7) Skip 5 columns.
 - (8) Pick up a six-digit field in Col. 6-11 of second card.
 - (9) Repeat for each pair of data cards.
- (vi) (5X,2F6.0,F1.0,3X,F5.0 //5X,F6.0,/F4.0,2X,F1.0), punched in the first 47 columns, will instruct the program to direct entry of data from each set of four cards as follows:
- (1)-(5) Sample as Example v.
 - (6) Skip second card and go to third card.
 - (7) Skip 5 columns.
 - (8) Pick up a six-digit field in Col. 6-11 of third card.
 - (9) Go to fourth card.
 - (10) Pick up a four-digit field in Col. 1-4 of fourth card.
 - (11) Skip 2 columns.
 - (12) Pick up a one-digit field in Col. 7.
 - (13) Repeat for each set of four cards.

SCALING

Scaling may be indicated by using either the "nFw.d" specification or the "sPnFw.d" specification. This specification will not often be required.

"nFw.d" Specification:

When the decimal point is not punched, the d of the above specification instructs the program to divide the whole number picked up by 10^d . For example, F6.1 will specify that the number picked up in a six-column field be divided by 10, F6.2 will specify division by 100, and F6.6 will specify division by 1000000.

$$\text{Stored number} = \text{punched number} / 10^d.$$

When the decimal point is punched, d is ignored.

Examples:

<u>Punched Number</u>	<u>Format Specification</u>	<u>Stored Number</u>
2468	F4.0	2468.0
3691	F4.1	369.1
4810	F4.3	4.810
4911.32	F7.0	4911.32
4911.32	F7.4	4911.32
172115	F6.6	0.172115

The entire format for these specifications might be punched as:

(F4.0,F4.1,F4.3,F7.0,F7.4,F6.6)

"sPnFw.d" Specification

Whether the decimal point is punched or not, the s of the above specification instructs the program to divide the number picked up by 10^s . For example, 2PF6.0 will specify division by 100, -2PF6.0 will specify division by .01.

Decimal point punched,

$$\text{Stored number} = \text{punched number} / 10^s, -8 < s < 8$$

Decimal point not punched,

$$\text{Stored number} = \text{punched number} / 10^{s+d}$$

Examples

<u>Punched Number</u>	<u>Format Specification</u>	<u>Stored Number</u>
7432	1PF4.0	743.2
74.32	-5PF5.0	7432000.0
7432	2PF4.3	.07432
7432	1P2F2.0	7.4 & 3.2
7432	2P2F2.1	0.074 & 0.032
7432	OPF4.1	743.2
7.432	-3PF5.0	7432.0

The entire format for these specifications might be punched as: (1PF4.0,-5PF5.0,2PF4.3,1P2F2.0,2P2F2.1,OPF4.1,-3PF5.0)

Mixed "nFw.d" and "sPnFw.d" Specifications:

Once the sPnFw.d specification has been used, it will hold for all Fw.d specifications to the right of it until another sPnFw.d is encountered.

If the sPnFw.d specification is not necessary to the right of its occurrence within the variable format statement, then OPFw.d should be used for the next specification to the right, which will hold then for the remaining specifications.

Examples:

(vii) (2F3.1,2X,F4.2,-6PF3.0,F4.0)

The program interprets the last field as -6PF4.0.

(viii) (2F3.1,2X,F4.2,-6PF3.0,OPF3.0)

The program interprets the last field as F3.0.

(ix) (1PF3.0,F2.0,OPF3.1,F4.2,F4.0)

The program interprets the second field as 1PF2.0.

2. I-type Variable Format

This format is required for programs designed to process only integer values. The specification is "nIw", where w is the width of the field (includes sign if punched), and n is the number of fields (assumed to be 1 if not punched). All data must be signed (+) or unsigned integers with no decimal point punched. Examples corresponding to those given for F-type are:

- (i) (12I3,I4,11I2)
- (ii) (12I3,I4,11I2 /)
 ...,11I2 //)
- (iii) (/12I3,I4,11I2)
 //12I3,...
- (iv) (10X,I6,2X,2I3)
- (v) (5X,2I6,I1,3X,I5 /5X,I6)
- (vi) (5X,2I6,I1,3X,I5 //5X,I6 /I4,2X,I1)

No scaling is permitted with I-type format.

3. A-type Variable Format

This format is required for programs designed to process data with alphabetic, numeric, or special characters, or combinations of these. The specification is "nAw", where w is the width of the field, $1 \leq w \leq 10$, and n is the number of fields (assumed to be 1 if not punched). Each specification of a field results in a computer word consisting of exactly 10 characters. When $w < 10$ the characters are positioned in the left of the field and the remaining characters are filled in with blanks. The following examples illustrate certain rules.

<u>Punched Data</u>	<u>Format Specification</u>	<u>Stored* Data</u>
12.0	A4	12.0bbbbbb
AGE	A3	AGEbbbbbbb
CANCER	A6	CANCERbbbb
\$	A1	\$bbbbbbbbbb
X+Y=A	A5	X+Y=Abbbb
DX,ØR,DY	A8	DX,ØR,DYbb
AGE	A6	bbbAGEbbbb

*b indicates blank character

IIC. Variable Output Formats

All the statistical programs use Variable Format Cards to describe the input data; a few require their use to describe output data, that is, data to be printed, punched, or written on tape by the computer. The function of the Variable Format Card is the same for input or for output: it is a description of the data in the medium external to the computer. Input and output formats are identical except for the following minor differences:

1. In F-type formats ("nFw.d" specifications), the decimal point is present (except when d=0) in the output medium, and a column must be allowed for it.
2. In using the scale factor specification of the form "sPnFw.d", the external representation of the number is 10^s times the internal number. Thus, if the internal number is -15.9357, a specification of 2PF9.1 would give "bb-1593.6" in the external medium (punched, printed, or tape output).
3. Each "line" of an input format for cards or of an output format for punched cards must not exceed 80 characters in length. Each "line" of an input format for an alternate BCD input tape or of an output format for printing or for a BCD output tape must not exceed 136 characters in length.
4. Position 1 of the printed line is used to control paper spacing and normally should be left blank (to produce

single spacing) by using "1X" as the first specification of the format. (Thus, when printing, only 135 positions are actually available to contain information.) If double spacing is desired, it may be obtained by using "1H0" instead of "1X".

Note: Care must be taken to allow sufficient width for the maximum size number that may be described by the format specification. (In describing input formats, this is essentially automatic because it is known how many columns of a card are devoted to a particular number.)

Example: Suppose it is desired to write an output format to print 27 signed numbers in F-type format to 4 decimal places, the maximum absolute value of the first 13 being less than 1000 and of the last 14, less than 50. The specification 13F10.4 allows room in each number for: 1 space, sign, 3 digits before the decimal point (< 1000), the decimal point, and 4 digits to the right of the point. This gives 130 characters on the first line, plus 1 for position 1 (for spacing control), and the first line is filled. Similarly, 14F9.4 may be used for the last 14 numbers. A new line is started by the slash ("/") in the format statement. If the spacing control is to be a double space before printing the first line of the group and single spacing within the group, then the complete format would be (note the use of "1X" in line 2):

(1H0,13F10.4/1X,14F9.4)

If it is desired to separate the numbers by more than one space, the following format might be used (three lines will be necessary):

(1H0,11F11.4/1X,2F11.4,10F10.4/1X,4F10.4)

For further control of line spacing in output formats, n consecutive slashes will produce n-1 blank lines.

IID. Finish Card

This card will notify the program that the entire job is finished. The program will complete its computations and will return control to the system monitor.

The preparation of this card is as follows:

Col. 1-6 FINISH

III. Preparation of System Cards

The system control cards listed in this section are described in detail in the User's Manual.

A. Job Card

See User's Manual section 3.1.1.1 THE JOB CARD.

B. REQUEST Card

See User's Manual section 3.1.1.13 THE REQUEST CARD.

C. LIBCOPY Card

See User's Manual section 5.1.1 BINARY LIBRARY (replacing CSCBIN with STATBIN).

D. LGO Card

See User's Manual section 3.1.1.5 THE PROGRAM CALL CARD.

IV. Examples of System Cards

A. Job Card

1. 7777,JONES,CM60000,T100.
CM field length = 60000 (octal)
time limit = 100 seconds (decimal)
priority level = 1 (decimal)
2. 11111,SMITH,P5,T64,L1100.
CM field length = 4000 (octal)
time limit = 64 seconds (decimal)
priority level = 5 (decimal)
line limit = 1100 lines (decimal)
3. 33333-BROWN,CM50000,T85,P10,L3300.
CM field length = 50000 (octal)
time limit = 85 seconds (decimal)
priority level = 10 (decimal)
line limit = 3300 lines (approx. 50 pages) (decimal)

B. LIBCOPY CARD

1. LIBCOPY(STATBIN,LGO,BMD8V)
2. LIBCOPY(STATBIN,LGO,WRAP)

C. Typical Deck Set-up

Jobcard with,CM70000,P5,T32,L1100.

LIBCOPY(STATBIN,LGO,WRAP)

LGO.

7-8-9 (end-of-record card, multiple-punched in column 1)
(program control cards and data)

6-7-8-9 (end-of-information card, multiple-punched in
column 1)

D. Source Library

If a source listing of the program is needed, the source decks are contained in the COMMON file named STATLIB.

The following example shows how to obtain a source listing of, compile and execute program BMD3R.

```
Jobcard with CM77000,L3300,T64.  
COMMON(STATLIB)  
EDITSYM(C=COMPILE,OPI=STATLIB)  
RETURN(STATLIB)  
RUN(S,,,COMPILE)  
LGO.  
7-8-9 (end-of-record, multiple-punched in column 1)  
*COPY,BMD3R  
7-8-9 (end-of-record, multiple-punched in column 1)  
    (program control cards and data)  
6-7-8-9 (end-of-information, multiple-punched in  
        column 1)
```

For further details of the Source Library see the User's Manual, Section 5.1.2. SOURCE LIBRARY.

GENERAL PLOT INCLUDING HISTOGRAM

1. Program Name: BMD5D
2. Central Memory (CM): 65000
3. General Description
 - a. This program provides a method by which graphs and histograms can be produced.
 - b. Output for this program includes:
 - 1) GRAPHS. Two methods of plotting are available:
 - a) This first method gives a one-page graph which has 50 units vertically and 100 units horizontally. The points are automatically scaled to conform to these dimensions, and a scale is printed both horizontally and vertically. The points (data cards) need be in no special order.
 - b) The second method gives a multiple-page graph with as many units vertically as there are values of the base variable. The values of the base variable (data cards) must be ordered and consecutive. The base variable is not scaled. The cross variables are scaled by the computer to conform to a horizontal dimension of 100 units.
 - 2) HISTOGRAMS

A one-page histogram can be produced, with a maximum of 34 intervals. The width of the interval must be specified; however, if the specified width would result in more than 34 intervals, the program will print comments to this effect and will compute a new width which will give exactly 34 intervals. Scales are printed on the vertical and horizontal axes.
 - c. Limitations per problem:
 - 1) p, number of original variables ($1 \leq p \leq 500$)
 - 2) n, number of cases ($2 \leq n \leq 15000$)
 - 3) q, number of variables added to the original set after transgeneration ($-499 \leq q \leq 499$)
 - 4) p+q total number of variables ($1 \leq p+q \leq 500$)
 - 5) (p+q)n total number of data ($2 \leq (p+q)n \leq 15000$)
 - 6) m, number of Transgeneration Cards ($0 \leq m \leq 999$)
 - 7) k, number of Variable Format Cards ($1 \leq k \leq 10$)
 - d. This program allows transgeneration. Codes 01-14 of the trans-generation list may be used.

4. Order of Cards

Cards indicated by letters enclosed in parentheses are optional. All other cards must be included in the order shown.

- a. Job Card (**) - III A)
- (b.) REQUEST Card(s) (**) - III B)
- c. LIBCOPY Card (**) - III C)
- d. LGO Card (**) - III D)
- e. 7-8-9 Card (multiple-punched in column 1)
- f. Problem Card
- g. F-type Variable Format Card(s) (**) - II B)
- (h.) Data Input Cards (**) - I B)
- (Place data input deck here if data is from cards)
- (i.) Standard Transgeneration Card(s) (**) - II A)
- j. Selection Card
- k. Heading Card(s) } Repeat from each
- (l.) Cross-Variable Card } graph or histogram
- ...
- Repeat f. through (l.) as desired
- ...
- m. Finish Card (**) - II D)
- n. 6-7-8-9 Card (multiple-punched in column 1)

5. Preparation of Cards Specific for this Program

f. Problem Card

Col.	1-6	PROBLM
	7-12	Alphanumeric problem name
	13-15	Number of original variables ($1 \leq p \leq 500$)
	16-20	Number of cases ($2 \leq n \leq 15000$)
	21-23	Number of Selection Cards
	24-27	Number of variables added to original set after transgeneration ($-499 \leq q \leq 499$)

Note: ($2 \leq (p+q)n \leq 15000$)

	28-65	Leave blank.
	66-68	Number of Transgeneration Cards ($0 \leq m \leq 999$)
	69-70	00 Data input from cards
		08 Data input from logical tape unit 8
	71-72	Number of Variable Format Cards ($1 \leq k \leq 10$)

j. Selection Card

A Selection Card has seven purposes:

- 1) To indicate whether a list of the data input is desired.
- 2) To indicate whether a graph or a histogram is to be produced.
- 3) To indicate the base variable of the graph or histogram.
- 4) To indicate the number of lines of heading desired for each graph or histogram.
- 5) To indicate for graphs how many variables are to be plotted against the base variable. (≤ 14)
- 6) To indicate for graphs the choice of the type of graph.
- 7) To indicate for histograms the width of an interval.

**** - PREFACE to the CDC STATISTICAL PROGRAMS**

If the Selection Card specifies that a graph is to be printed, the Heading Card is followed by a Cross-Variable Card which indicates the cross variables to be plotted against the base variable and the symbols used for each cross variable.

Col.	1-6	SELECT
	7	Number of lines in a heading. Each Heading Card specifies one line of printed output. The maximum number of lines allowed in the heading is two. (See card k.)
	8	0 Listing of input data is not desired 1 Listing of input data is desired.
	9-10	Number of cross variables to appear on this graph (maximum is 14)
	11-13	Index of the base variable. On graphs, the base variable will appear on the vertical axis. On histograms, the base variable will appear on the horizontal axis.
	14-24	Form of the graph or width of interval if a histogram.
		Col. 14-15 01 If a one-page graph is desired -1 If a multiple-page graph is desired, or
		Col. 14-24 Width of the interval for a histogram (punch the decimal point).

k. Heading Card(s)

Col.	1-72	Punch the desired heading. Each card is a line of the heading. There must be <u>at least one Heading Card</u> , but no more than two, per graph or histogram.
------	------	---

(l.) Cross-Variable Card

The Cross-Variable Card is punched as follows (for graphs only, not histograms). The cross variables specified to be crossed with one base variable will appear on one graph; the cross variables will appear on the horizontal axis.

Col.	1-6	CRSVAR
	7-9	Index of the 1st cross variable
	10	Symbol for the 1st cross variable (see below)
	11-15	Leave blank.
	16-18	Index of the 2nd. cross variable
	19	Symbol for the 2nd. cross variable
	20-24	Leave blank.
	.	
	.	
	.	
	61-63	Index of the 7th cross variable
	64	Symbol for the 7th cross variable
	65-69	Leave blank.

The symbols to be used for each cross variable must be specified.
Allowable symbols are:

. , - J K L M N O P Q R S T U V W X Y Z *

The following symbols may not be used because they have been used to represent ties (more than one point occurring at the same coordinates):

<u>Symbol</u>	<u>No. of Points</u>	<u>Symbol</u>	<u>No. of Points</u>	<u>Symbol</u>	<u>No. of Points</u>
2	2	8	8	E	14
3	3	9	9	F	15
4	4	A	10	G	16
5	5	B	11	H	17
6	6	C	12	I	18
7	7	D	13	/	more than 18

If there are more than seven cross variables, continue punching a second card in the same manner

Col. 1-6 CRSVAR
 7-9 Index for the 8th cross variable
 10 Symbol for the 8th cross variable
 11-15 Leave blank.
 .
 .
 .
 61-63 Index for the 14th cross variable
 64 Symbol for the 14th cross variable
 65-69 Leave blank.

The maximum number of cross variables for a specified base variable is 14.

- For a brief description of the computational procedure, refer to the Biomedical Computer Programs manual.

CORRELATION PROGRAM

1. Program Name: BMD2D

2. Central Memory (CM): 75000

3. General Description

- a. This program computes simple correlation coefficients, averages and measures of dispersion on entering variables and/or trans-generated variables.

A special feature of this program is the selection of cases from the input data by specifying a Boolean expression; i.e., case is accepted if it is in agreement with the expression; otherwise, the case is skipped. The expression consists of variables and constants involving relationships of equality or inequality written in a logical form using the operations AND and OR.

- b. Output from this program includes:

- 1) Sums
- 2) Means
- 3) Standard deviations
- 4) Correlation matrix

Optional output includes:

- 5) Cross-product deviations
- 6) Variance-covariance matrix
- 7) One-page cross-tabulation plots of any two variables, automatically scaled to 50 (vertical) by 100 (horizontal) character spaces or units.

- c. Limitations per problem:

- 1) p, number of original variables ($2 \leq p \leq 135$)
- 2) n, number of original cases ($2 \leq n \leq 99,999$)
- 3) j, number of Plot Selection Cards ($0 \leq j \leq 99$)
- 4) q, number of variables added to the original set after transgeneration ($-133 \leq q \leq 133$)
- 5) b, number of Case Selection Cards ($0 \leq b \leq 9$)
- 6) m, number of Transgeneration Cards ($0 \leq m \leq 150$)
- 7) k, number of Variable Format Cards ($1 \leq k \leq 10$)

- d. The program allows transgeneration of the input data. Codes 01-16 and 41 of the transgeneration list may be used.

4. Order of Cards

Cards indicated by letters enclosed in parentheses are optional. All other cards must be included in the order shown.

a. Job Card

(b.) REQUEST Card(s)

(** - III A)
(** - III B)

** - PREFACE to the CDC 6500 STATISTICAL PROGRAMS

- c. LIBCOPY Card (**) - III C)
- d. LGO Card (**) - III D)
- e. 7-8-9 Card (multiple-punched ... column one)
- f. Problem Card
- (g.) Standard Transgeneration Card(s) (**) - II A)
- (h.) Case Selection Card(s)
- i. F-type Variable Format Card(s) (**) - II B)
- (j.) Data Input Card (**) - I B)
- (Place data input deck here if data is from cards)
- (k.) Plot Selection Card(s)

...

Repeat f. through (k.) as desired.

...

- l. Finish Card (**) - II D)
- m. 6-7-8-9 Card (multiple-punched in column one)

5. Preparation of Cards Specific for this Program

f. Problem Card

Col.	1-6	PROBLM
	7-12	Alphanumeric problem name
	13-15	Number of original variables ($2 \leq p \leq 135$)
	16-20	Number of original cases ($2 \leq n \leq 99999$)
	21-22	Number of Plot Selection Cards; if none, leave blank. ($0 \leq j \leq 99$)
	23-26	0000 No variables added to, or subtracted from, the original set after trans-generation
		+q q variables added to the original set after transgeneration ($2 \leq p + q \leq 135$)
		-q q variables subtracted from the original set after transgeneration
	27-28	00 No Case Selection Cards
		+b b cards used for Boolean expression; case selection occurs after trans-generation ($b \leq 9$)
		-b b cards used for Boolean expression; case selection occurs prior to trans-generation ($ b < 9$)
	29-30	NO if matrix of cross products is not desired
	31-32	NO if covariance matrix is not desired
	33-34	NO if alternate input tape is not to be rewound
	35-65	Leave blank.
	66-68	000 No transgeneration
		m m Transgeneration Cards ($0 \leq m \leq 150$)
	69-70	00 Data input from cards
		08 Data input from logical tape unit 8
	71-72	Number of Variable Format Cards ($1 \leq k \leq 10$)

** - PREFACE to the CDC 6500 STATISTICAL PROGRAMS

(h.) Case Selection Card(s)

It is often useful to select cases if the value of a particular variable is less than some constant, greater than some constant, equal to some constant, etc. Symbolically,

$$\begin{aligned} V(I) &< C \\ V(I) &> C \\ V(I) &= C \end{aligned}$$

where I is the index of some variable. To select only those cases where the values of a variable are between two constants involves the operation AND.

$$V(I) > C \text{ AND } V(I) < B$$

To select only those cases where either of two variables must satisfy a relationship involves the operation OR.

$$V(I) > C \text{ OR } V(J) < B$$

Perhaps a more complicated expression is desirable, e.g.,

$$(V(I) > A) \text{ OR } (V(J) < B) \text{ AND } (V(K) = C), \dots$$

According to rule, the entire Boolean expression is either true or false for the case being tested. It is examined from left to right. If an OR is encountered, and the expression preceding the OR is true, the entire expression is considered to be true for this case, and the case is selected for inclusion.

Since parentheses cannot be used for compound AND/OR expressions, AND is assumed to precede OR. The statement

$$W \text{ OR } X \text{ AND } Y \text{ OR } Z$$

will operate as

$$W \text{ OR } (X \text{ AND } Y) \text{ OR } Z.$$

A Case Selection Card is written as a sequence of conditions separated by an operation. A condition is a variable and a constant separated by a relationship.

Variables: A variable is specified by the alphabetic V and the variable index; V(100), V(010), V(149), V(008). The three-digit index is necessary; it is enclosed by parentheses.

Constants: Constants are specified by their literal value, e.g., -22.43, .99090, 1.0000, .00009. Five numeric characters with a decimal point are allowed. If the sign (+, -) is used, then only four numeric characters are allowed.

Relationships: Relationships are specified by using the following two-character codes: GT (greater than), LT (less than), GE (greater than or equal to), LE (less than or equal to), EQ (equal to), NE (not equal to).

Operations: Operations are specified by using the following two-character codes: AN (and), OR (or), ** (end of expression).

NOTE: (not greater than) -- LE
 (not less than) -- GE
 (not greater than or equal to) -- LT
 (not less than or equal to) -- GT

Examples:

i) (V(002NEV(100))**

The case is accepted if variable 2 is not equal to variable 100.

ii) (V(010)GE100.00)AN(V(010)LT200.00)**

The case is accepted if variable 10 is greater than or equal to 100.00 and variable 10 is less than 200.00.

The preparation of the Case Selection Card is as follows:

Col.	1-3	(V(
	4-6	Three-digit variable index
	7	)
	8-9	Two-character relationship
	{ 10-11	V(
	12-14	Three-digit variable index
	15	)
	{ 10-15	Constant (Keypunch decimal)
	16	)
	17-18	Two-character operation

This format is repeated four times per card ending in Column 72. The maximum number of cards is nine. The last operation of the expression must be **. Therefore, the user may specify from one to 36 conditions, each condition followed by an operation, the last operation being **.

(k.) Plot Selection Card(s)

Col.	1-6	PLOTSL
	7-9	Index of the base variable (X - axis)

10-11 Number of variables to be cross-plotted with this base variable (≤ 20)

12-14 Index of the 1st variable to be cross-plotted with this base variable

15-17 Index of the 2nd variable to be cross-plotted with this base variable

69-71 Index of the 20th variable to be cross-plotted with this base variable

Each Plot Selection Card is independent. The same or different base variables may be specified on additional cards. The maximum number of Plot Selection Cards is 99.

6. For a brief description of the computational procedure, refer to the Biomedical Computer Programs Manual.