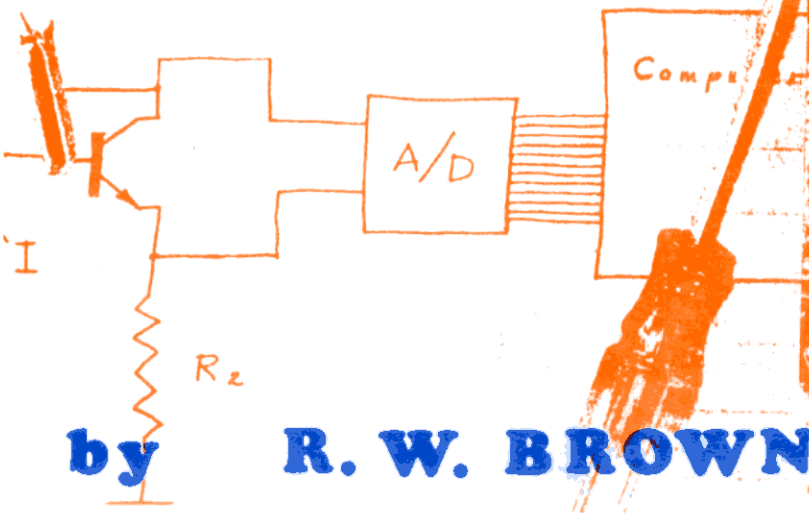
$ 9.95

BASIC SOFTWARE LIBRARY

VOLUME V

EXPERIMENTER'S PROGRAMS



by R. W. BROWN

BASIC SOFTWARE LIBRARY

VOLUME V

EXPERIMENTER'S PROGRAMS

1ST. PRINTING - - MARCH 1977

SCIENTIFIC RESEARCH INST.
P.O. BOX 3692
CROFTON, MD 21114

INTRODUCTION

The programs presented here are set out for the individual who has a specific need in mind. Because a detailed discussion of these programs would require a text several times the present size of this Library it has been omitted. Individuals who have a specific requirement will have to be at least knowledgeable in the area the program is written about; ie: Statistical programs require the user to be familiar with the terms mean, median, etc. This is because the programs are written in the vernacular of their subject matter. With this knowledge alone, no programming experience on the part of the user is required in order to use any of these programs in most systems. Once it is determined that a particular program may be useful the user merely types in a copy of the BASIC source code exactly as it appears in the program listing. Then follow the instructions for running the program as presented in the Instruction portion of the write up, immediately preceding the program. Also included in the write ups are statements that appear in the source code which may possibly need to be changed to run in the user's computer system; ie: RND statements may have to be changed to FRAND in order to compile in certain systems.

PUBLISHERS NOTE: Appendix B included at the end of Volume V was not mentioned in the preface by the author. We feel this appendix is the most important single item included in this library. We see this appendix as a fore runner that might lead the way toward standardizing a computer language among the manufacturers. This is in addition to the obvious benefits to all users of this Basic Software Library.

TABLE OF CONTENTS

VOLUME ONE

TABLE OF CONTENTS

VOLUME ONE (CONT.)

PICTURES

TABLE OF CONTENTS

VOLUME TWO

Part 3 - Math & Engineering Programs

TABLE OF CONTENTS

VOLUME TWO (CONT.)

Part 4 - Plotting & Statistics Programs

TABLE OF CONTENTS

VOLUME THREE

Part 5 - Advanced Business Programs

TABLE OF CONTENTS

VOLUME FOUR

General Purpose Programs

TABLE OF CONTENTS

VOLUME FIVE

PAGE

Experimenter's Programs

TABLE OF CONTENTS

VOLUME FIVE (CONT.)

TABLE OF CONTENTS

VOLUME SIX

PREFACE

The information contained in these pages represents the culmination of a very large project. That of compiling a versatile and complete Software Library that will be of use to a large number of diverse individuals. The programs presented here when combined in a system will represent a very powerful library bank. Such a work as this has been attempted in the past in such areas as cook books, electronic source books, mathematical tables and even computer games. But to date such a collection as this has yet to be offered to the average individual to use as he chooses. The word "at - temped" was used as no work is ever considered complete by everyone regar dless of its thoroughness.

The programs presented here were chosen for their uniqueness and general usefulness. There should be at least one program included that will be of use to every type of individual whether they have access to a computer or not. Computers are a wonderful and very useful tool. Through this Library I hope to interest more people into becoming involved with computers. The Library is written so that little or no computer programming experience is required to invoke any of the programs. The programs that are present- ed here are all written in the computer language called BASIC. Each pro - gram has been successfully run on a G.E. 635 computer. The entire source code is presented as well as a short narrative page which defines the pro gram, tells who might be interested in using it, a brief set of instruc- tions or how to get them and then any limitations in the program are no- ted. In the limitations section the storage length in K Bytes is given so the prospective user will know how much memory to allow for the program. Where possible the amount of memory space required for full execution is given for the programs, this space is independent of the space already oc cupied by your BASIC compiler.

The programs are broken down into five sections or parts. Each part deals with a specific type of program. Part 1 contains business type programs. These programs will be of interest to individuals who have businesses, play the stock market, balance their own checkbooks, do installment buy- ing, figure taxes, etc. There are a total of 20 programs in this section. Part 2 is the lighter side of the Library as it contains 16 games and 12 picture programs. No computer library is complete without some fun. Among the games presented in this section is one called Checkers. The game is rather long but it is virtually machine independent as it doesn't use over lay techniques or use files. Most of the other games included here are as exciting as this version of Checkers. Each was chosen so as not to mimic others that the reader may have seen. The pictures are as unusual in their own way as are the games. Most of the pictures are spread over several pa ges, this was done not only so the reader will need to run the program to see the details of a particular picture but also in the hopes of getting as many of these programs into use as possible. As the picture programs are very simple it is an easy place for the novice to start learning about programming.

Part 3 is comprised of Math and Engineering programs. Some of these programs will be of use to high school students, professional people, sailors, engi- neers, astronomers, airplane pilots, etc. Most of these programs are very

technical but they can perform every day calculations quickly and easily and they are extremely simple to use. There are 23 general usage programs presented in this section.

Part 4 is made up of Plotting and Statistical Analysis programs. These programs can be readily utilized by a number of people in widely different disciplines from fishermen to statisticians. The data gathered may be from a poll, a census, a test sample or even the number of fish caught on various days. The stat programs will be of invaluable aid to anyone who gathers data of any kind. The plotting routines will be of use to most of the people who use the stat programs or programs in Parts 1 and 3. The plotting is done on any standard teletype or terminal and does not require a special plotter or plotting terminal. There are a total of five direct plotting programs and 18 stat programs in this section.

All of the programs presented here may be run by simply typing the source code as listed, exactly as it is, into your computer. Now before the program will run it will have to be converted into machine code. This is done automatically and requires no forethought except to make certain the operating system you are working in is BASIC. In the larger computer systems you are asked what system you want — to this type BASIC; the smaller systems only have BASIC, in these you are O.K.

Immediately following Part 4 is Appendix A. Here, all of the Basic Statements used throughout these pages are defined. Each statement is explained sufficiently well to enable one unfamiliar with this subset to modify any necessary statements so that the program or programs will compile and execute with the Basic compiler or interpreter available with their particular computer. Most of the Basic compilers available today, that require more then 10K Bytes of storage, will execute all of the programs presented in these volumes with the possible exception of a few of the games and the program "Variable". Multiple line statements are not used in most of the programs and only a few programs use string manipulations extensively. A few of the programs may require more on line storage then is available on some of the small micro computer systems; these longer programs will not be executable due to the limited amount of memory. However most of the programs will execute in 10K Bytes of memory or less, thereby making most of the programs in this Library executable in virtually any Basic speaking computer without any required modifications.

Volume III is comprised of ADVANCED BUSINESS programs, part 5. This volume as well as subsequent volumes are intended to make this Library complete and useful to all individuals.

Each of these programs are written in a subset of the Dartmouth language. The specific subset is that which was used by General Electric on their 635 systems. These programs have operated without problem on a variety of small and large machines even several of the new micro computers. The programs that use string manipulations may require slight modifications before fully executing on some systems. These programs are mainly found in Part 2 — Games.

All of the programs in this Library were written or edited by the author. All of the programs edited by him were given for inclusion, "swapped" for traded, or made public. A few of the original authors of the "swaps" are not known, for this I apologize. The others, unless specifically mentioned in the text, are presented here. In addition I would like to thank the fol lowing for their cooperation in making this work possible.

ACKNOWLEDGMENTS

MY WIFE MARY AND MY FAMILY

DONALD ALVAREZ

DAVE BEETLE

MORTON BERGER

COPY CAT INC

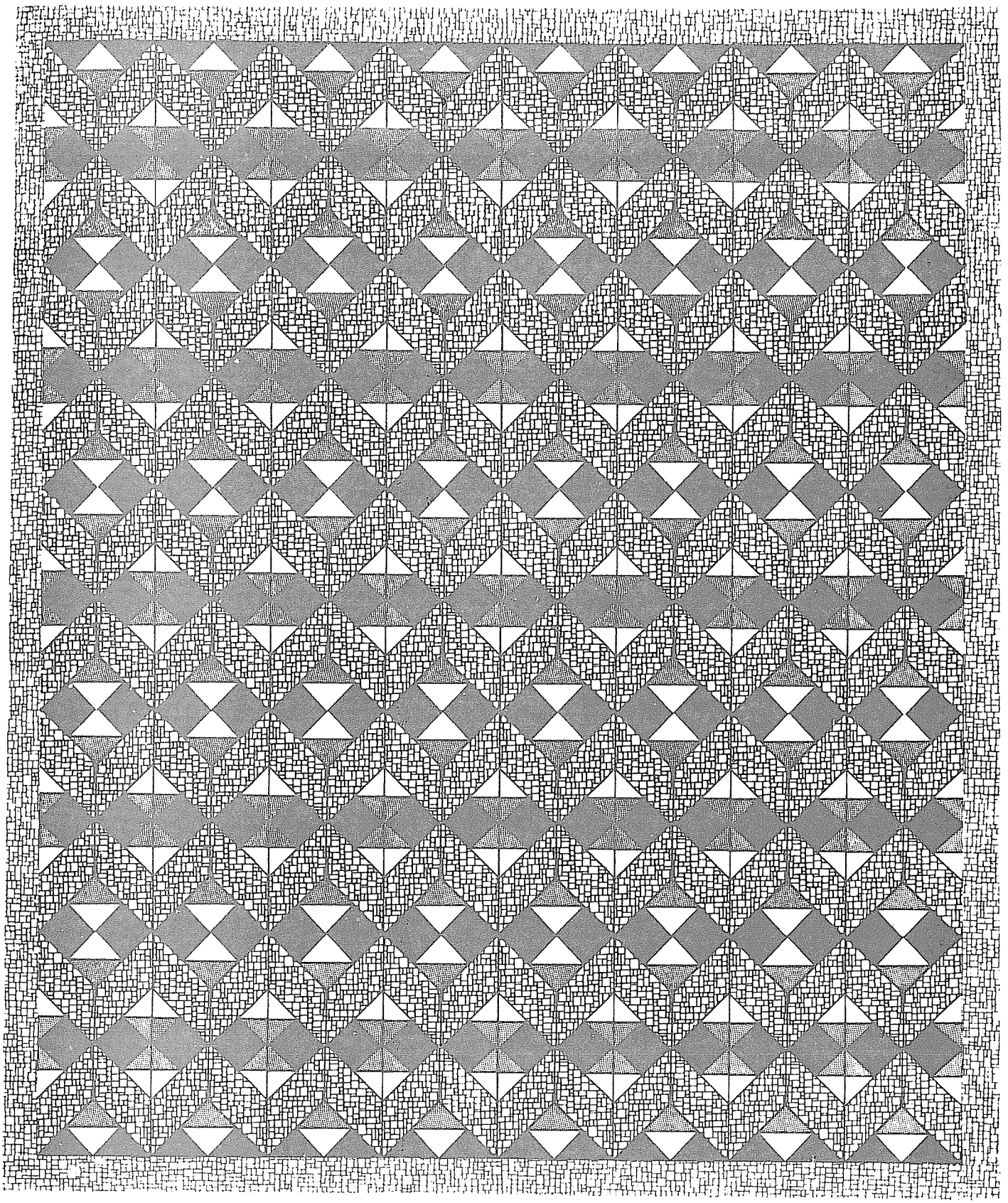
GE TIMESHARING

BILL JONES

GEORGE LONG

TOM ROSE

ARTWORK COURTESY OF MELISSA



EXPERIMENTER'S

PROGRAMS

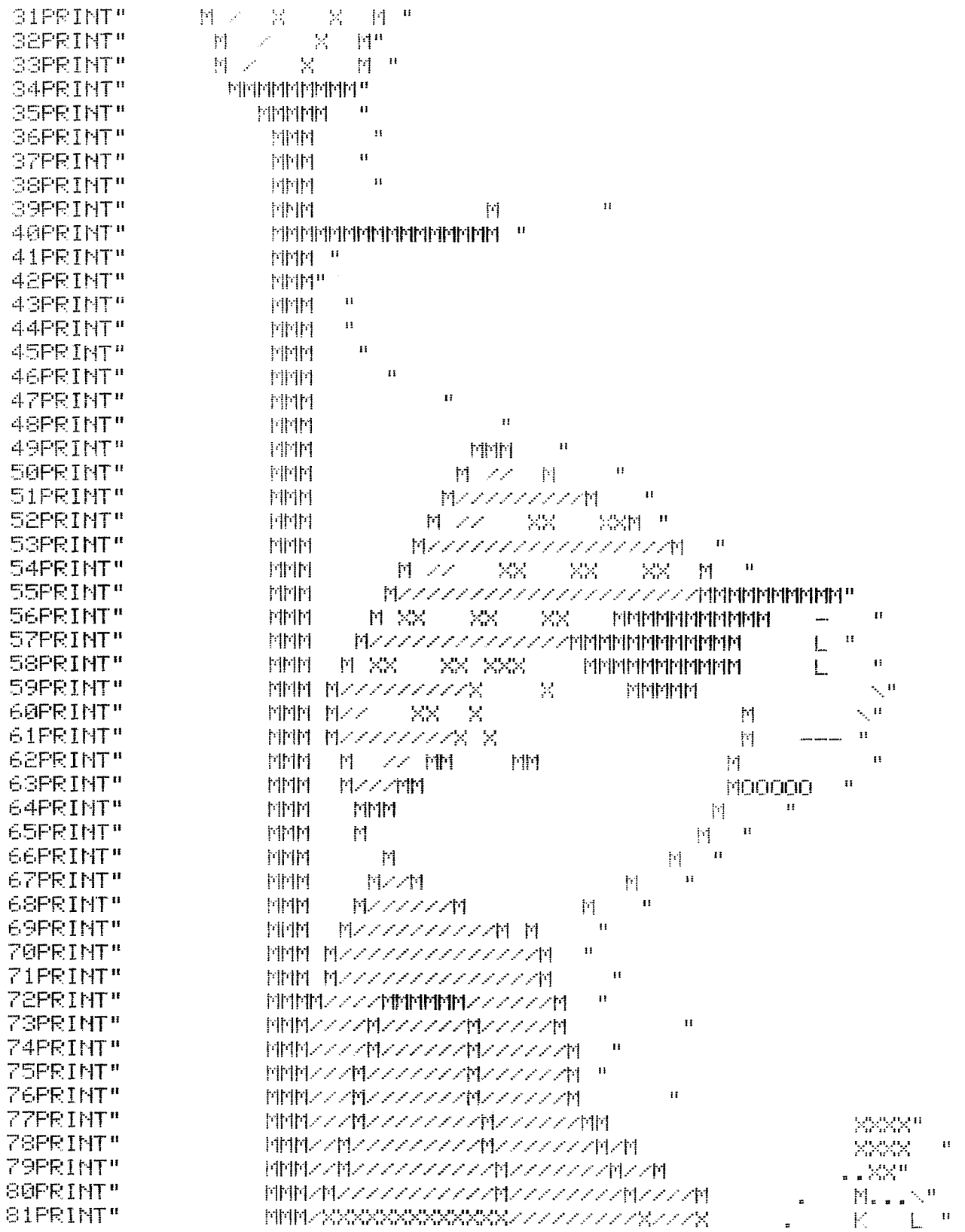
ANDY CAP:

DESCRIPTION

Here is our famous cartoon character taking time out from his daily toil for a bit of nourishment. The source code for this program will require 6K Bytes of memory for storage and it will execute in 7K in any Basic speaking computer.

```
ANDY CAP


001 PRINT
002PRINT
003PRINT
004PRINT
005PRINT
06PRINT
07PRINT
08PRINT
09PRINT
10PRINT
11PRINT"           MMMMM        "
12PRINT"         MMMMMMM   "
13PRINT"        MMMMMMMMM    "
14PRINT"       MMMMMMMMMMM    "
15PRINT"      MMMMMMMMMMMMM    "
16PRINT"     MMMMMMMMMMMMMMM      "
17PRINT"    MMMMMMMMMMMMMMMMM    "
18PRINT"  MMMMMMMMMMMMMMMMMMM      "
19PRINT" MMMMMMMMMMMMMMMMMMMMM   "
20PRINT"MMMMMMMMMMMMMMMMMMMMMMM     "
21PRINT"MMM MMM MMM MMM MMM MMM"
22PRINT" M      /      X  X  X M "
23PRINT" M     /      X  X  X  M "
24PRINT"  M  /      X  X  X  M   "
25PRINT"  M /      X  X  X   M      "
26PRINT"   M      /  X  X   M "
27PRINT"   M     /  X  X   XM   "
28PRINT"    M  /  X  X   XM     "
29PRINT"    M /  X  X   X M"
30PRINT"     M  /  X   X M "
```

NOTE: The listings presented in these volumes are all generated from a terminal that prints ten (10) characters per horizontal inch. In order to reproduce any pictures presented here, one should lay a ruler below each line as it is being keyed into your computer. When a space separates the characters; generating the picture, simply measure the empty distance between the characters in inches and multiply this length by ten. This will be the number of spaces separating the two characters in question.



```
31PRINT"      M /  %   %  M "
32PRINT"       M  /    %  M"
33PRINT"       M /   %   M "
34PRINT"        MMMMMMMMM"
35PRINT"          MMMMM  "
36PRINT"           MMM    "
37PRINT"           MMM   "
38PRINT"           MMM    "
39PRINT"           MMM              M        "
40PRINT"           MMMMMMMMMMMMMMMMMM "
41PRINT"           MMM "
42PRINT"           MMM"
43PRINT"           MMM  "
44PRINT"           MMM  "
45PRINT"           MMM   "
46PRINT"           MMM     "
47PRINT"           MMM          "
48PRINT"           MMM              "
49PRINT"           MMM            MMM   "
50PRINT"           MMM           M //  M    "
51PRINT"           MMM          M/////////M   "
52PRINT"           MMM         M //   %%   %%M "
53PRINT"           MMM        M/////////////////M  "
54PRINT"           MMM       M //   %%   %%   %%  M  "
55PRINT"           MMM      M/////////////////////MMMMMMMMMMM"
56PRINT"           MMM     M %%   %%   %%   MMMMMMMMMMM   -   "
57PRINT"           MMM    M/////////////MMMMMMMMMMMMM     L "
58PRINT"           MMM  M %%   %% %%%    MMMMMMMMMMM     L   "
59PRINT"           MMM M/////////%    %     MMMMM              \"
60PRINT"           MMM M//   %%  %                    M       \"
61PRINT"           MMM M/////////% %                  M   --- "
62PRINT"           MMM  M  // MM    MM               M         "
63PRINT"           MMM  M//MM                       MOOOOO  "
64PRINT"           MMM   MMM                      M    "
65PRINT"           MMM   M                      M  "
66PRINT"           MMM     M                  M  "
67PRINT"           MMM    M/M                M   "
68PRINT"           MMM   M/////M          M   "
69PRINT"           MMM  M/////////M M    "
70PRINT"           MMM M/////////////M  "
71PRINT"           MMM M/////////////M    "
72PRINT"           MMMM///MMMMMM/////M  "
73PRINT"           MMM///M/////M/////M       "
74PRINT"           MMM///M/////M/////M  "
75PRINT"           MMM//M/////M/////M "
76PRINT"           MMM//M/////M/////M       "
77PRINT"           MMM//M////////M/////MM                  %%%%"
78PRINT"           MMM/M/////////M///////M/M                %%%%  "
79PRINT"           MMM/M/////////M///////M/M              ..%%"
80PRINT"           MMM/M/////////M///////M///M       .   M...\"
81PRINT"           MMM/%%%%%%%%%%/////////%///%     .    K   L "
```

```
82PRINT"            MMM/////       %%%%%%%%%%%%%%//%.....      K   L  "
83PRINT"            MMM/////       %%%%%%%%%%%%%%%%%            \   L      "
84PRINT"            MMMM////       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%\...\       "
85PRINT"            MMM X///       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
86PRINT"            MMM  X//       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
87PRINT"            MMM  M//       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
88PRINT"            MMM  M///       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
89PRINT"            MMM M/////       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
90PRINT"            MMMM///////       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
91PRINT"            MMMMMMM/////       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
92PRINT"            MMMMMMMMMMMMMMM       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
93PRINT"            MMMMMMMMMMMMMMM       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
94PRINT"            MMMMMMMMMMMMMMM    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
95PRINT"            MMM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
96PRINT"            MMM  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
97PRINT"          MMMMM    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
98PRINT"          MMMMM       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
99PRINT"          MMMMM          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
100PRINT"         MMMMM             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
101PRINT"         MMMMM               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
102PRINT"         MMMMM                 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
103PRINT"         MMMMM                   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
104PRINT"         MMMMM                     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
105PRINT"         MMMMM                       %%%%%%%%%%%%%%%%%%%%%%%%%%%%  "
106PRINT"         MMMMM                         %%%%%%%%%%%%%%%%%%%%%%%%%%  "
107PRINT"         MMMMM                           %%%%%%%%%%%%%%%%%%%%%%%%  "
108PRINT"         MMMMM                             %%%%%%%%%%%%%%%%%%%%%%"
109PRINT"         MMMMM                               %%%%%%%%%%%%%%%%%%%%  "
110PRINT"         MMMMM          %%%%                   %%%%%%%%%%%%%%%%%%  "
111PRINT"         MMMMM          %%%%    %%    %%%%%%%%%%%%%%%%%%%%%%%%  "
112PRINT"         MMMMM  %%%%     %%     %%    %%%%%%%%%%%%%%%%%%%%%%%%  "
113PRINT"         MMMMM  %%%%     %%     %%      %%%%%%%%%%%%%%%%%%%%%%"
114PRINT"         MMMMM    %%     %%     %%      %%%%%%%%%%%%%%%%%%%%%  "
115PRINT"         MMMMM    %%     %%    %%%%      %%%%%%%%%%%%%%%%%%%%"
116PRINT"         MMMMM    %%     %%%% %%%%%%      %%%%%%%%%%%%%%%%%%  "
117PRINT"         MMMMM  %%%%   %%%%%%.    %%      %%%%%%%%%%%%%%%%%%"
118PRINT"         MMMMM %%%%%% %     %.    %%      %%%%%%%%%%%%%%%%%  "
119PRINT"         MMMMM %%    . %     %.    %%       %%%%%%%%%%%%%%%%"
120PRINT"         MMMMM %%    . %     %.    %%       %%%%%%%%%%%%%%%    "
121PRINT"         MMMMM %%    . %     %.    %%      ZZZZZZZZZZZZZZZZZZZ"
122PRINT"         MMMMM %%    . %%%%%.    %%     ZZZZZZZZZZZZZZZZZZZZZ"
123PRINT"         MMMMM %%    . %%%%%%%%%%%%     ZZZZ    ZZZZZZZZZZZ"
124PRINT"         MMMMM %%%%%%       %%%%%%   "
125PRINT
126PRINT
127PRINT
128PRINT
129PRINT
130 END
```

BASEBALL:

DESCRIPTION

This is a computer generation of the game of baseball and may be played by one or two players. While the game is not interactive, after it begins it is still exciting and should provide entertainment to all baseball fans.

INSTRUCTIONS

After the program has been loaded into memory type RUN. The program will first ask you for the name of the visiting team and then the name of the home team. After these names have been entered the game commences.

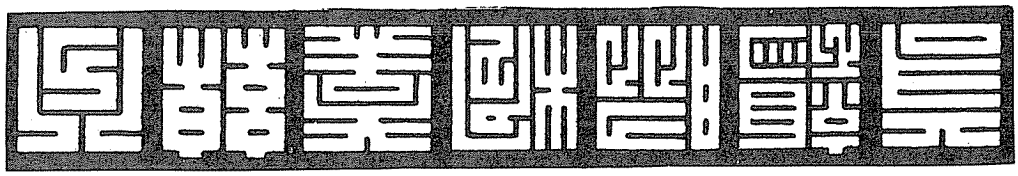
LIMITATIONS

This game requires two dimensional arrays; see line 1020, for execution. Lines 1130, 1300, 1410, and 1420 contain ON__ GOTO statements, line 1890 contains an ABS( ) statement and line 1960 a LEN( ) statement. The source code stores in 3K Bytes and executes in 5K Bytes of memory.

```
1020 DIM P(9,4),Q(4,8)
1030 MAT READ P,Q
1040 PRINT"----------BASEBALL----------"
1050 PRINT"VISITING TEAM NAME";
1060 INPUT A$
1070 PRINT
1080 PRINT"HOME TEAM NAME";
1090 INPUT B$
1100 PRINT
1110 S1=S2=Z=H2=H1=B=0
1120 I=1
1130 ON Z+1 GOTO 1140,1160
1140 PRINT A$;
1150 GOTO 1170
1160 PRINT B$;
1170 PRINT" IS UP     INNING";I
1180 PRINT
1190 S=O=C=0
1200 C=1
1210 IF B<9 THEN 1230
1220 B=0
1230 B=B+1
1240 Y=RND(-4)+5E-3
1250 FOR R=1 TO 4
1260 Y=Y-P(B,R)
1270 IF Y<0 THEN 1300
1280 NEXT R
1290 R=5
1300 ON R GOTO 1310,1340,1360,1380,1400
1310 PRINT"      O U T"
1320 O=O+1
1330 GOTO 1410
1340 PRINT"----------SINGLE----------"
1350 GOTO 1410
1360 PRINT"WALK"
1370 GOTO 1410
1380 PRINT"-*-*-*-*-*DOUBLE*-*-*-*-*-"
1390 GOTO 1410
1400 PRINT"**************OUT OF THE BALLPARK**************"
1410 ON R GOTO 1500,1420,1460,1420,1420
1420 ON Z+1 GOTO 1430,1450
1430 H1=H1+1
1440 GOTO 1460
1450 H2=H2+1
1460 L=Q(R-1,C)
```

```
1470 M=INT(L/10)
1480 C=L-M*10
1490 S=S+M
1491 IF I<9 THEN 1500
1492 IF Z=0 THEN 1500
1493 IF S2+S>S1 THEN 1510
1500 IF O<3 THEN 1210
1510 PRINT
1520 IF S=1 THEN 1550
1530 N$="RUNS"
1540 GOTO 1560
1550 N$="RUN"
1560 PRINT S,N$
1570 PRINT
1580 IF Z=0 THEN 1650
1590 S2=S2+S
1610 PRINT A$,S1
1620 PRINT B$,S2
1621 IF I>=9 THEN 1730
1630 PRINT"

          "
1640 GOTO 1700
1650 S1=S1+S
1660 PRINT
1670 IF Z=1 THEN 1700
1671 IF I<>9 THEN 1680
1672 IF S2>S1 THEN 1740
1680 Z=1
1690 GOTO 1130
1700 Z=0
1710 I=I+1
1720 GOTO 1130
1730 IF S1=S2 THEN 1920
1740 PRINT
1750 PRINT"GAME TOTALS"
1760 PRINT
1770 PRINT A$;
1780 PRINR S1;" RUNS";
1790 PRINT H1;" HITS"
1800 PRINT
1810 PRINT B$;
1820 PRINT S2;" RUNS";
1830 PRINT H2;" HITS"
1840 PRINT
1850 IF S1>S2 THEN 1880
1860 K$=B$
1870 GOTO 1890
1880 K$=A$
1890 PRINT K$;" WINS BY   ";ABS(S1-S2)
1900 PRINT
```

```
1910 GOTO 1940
1920 PRINT"THE GAME IS TIED AND NEEDS ANOTHER INNING"
1930 GOTO 1700
1940 PRINT"WANT ANOTHER GAME";
1950 INPUT K$
1960 IF LEN(K$)=3 THEN 1050
1970 DATA .644,.24,.077,.039,.644,.19,.103,.053,.72,.167,.046,.05
1980 DATA .641,.244,.05,.038,.673,.153,.073,.038,.651,.182,.106,.038
1990 DATA .704,.162,.079,.036,.709,.162,.071,.04,.836,.09,.053
2000 DATA .016,2,5,6,12,8,15,16,18,2,5,6,12,8,15,16,18,3,7,13,13,18
2010 DATA 17,23,27,11,21,21,21,31,31,31,41
2020 END
```



COMPARE:

DESCRIPTION

This program is used to compare two data groups. The comparison is done using the median test. The program prints the Chi-Square stats for the input using one degree of freedom as the calculation limits.

USERS

Users of Compare will be individuals who are interested in determining the gross similarities between groups of data. These individuals would include statisticians, engineers, businessmen, etc.

INSTRUCTIONS

Before this program is run enter the data for the two groups of data in data statements starting in program line 900, using the following sequence:

```
900 Data X, Y
910 Data X1, X2.....
920 Data Y1, Y2,....
```

where X is the number of elements in the first group, and Y is the number of elements in the second group. X1 and conversely Y1, Y2, etc. represent the individual data points for each group. After the data has been entered type RUN. For a detailed list of instructions list the program.

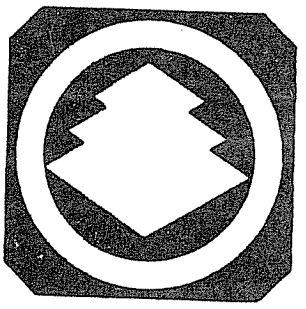
LIMITATIONS

Line 94 contains a Restore statement, line 140 a MAT READ statement and 360 ABS( ). These are defined in the Appendix at the end of Volume II. If your Basic does not contain Mat statements they may be easily generated through a double FOR loop, (Refer to Appendix B at the end of this Volume). The source code is 2K Bytes long and the program executes in 8K.

COMPARE

```
10    REM  *** DESCRIPTION: COMPARES TWO GROUPS OF DATA USING THE
20    REM  MEDIAN TEST.
40    REM  *** INSTRUCTIONS: PUT DATA IN LINE 900 AND FOLLOWING.
60    REM  THE FIRST DATUM IS THE NUMBER OF ENTRIES IN THE FIRST
70    REM  GROUP, THEN ENTER THE NUMBER OF ENTRIES IN THE SECOND GROUP,
80    REM  THEN THE FIRST GROUP ITSELF IS ENTERED, AND THEN THE
90    REM  SECOND GROUP.  THE PROGRAM PRINTS OUT THE
91    REM  CHI SQUARE STATISTIC OF THE 2 BY 2 TABLE, ON 1 DEGREE
92    REM  OF FREEDOM.
93 READ T
94 RESTORE
95 IF T<>9999 THEN 100
96 PRINT "LIST LINES 10 TO 92 FOR INSTRUCTIONS"
97 STOP
99  DIM A(500)
100    READ M, N
110  LET M1 = M + N
120  LET M2 = INT( M1/2 )
140    MAT READ A(M1)
152  LET L = 1
153  LET U = M
155  GOSUB 600
156  LET L = M + 1
157  LET U = M1
158  GOSUB 600
160 LET X=0
161 LET Y=0
180  LET I = 1
190  LET J = M + 1
200  FOR K = 1 TO M2
210     IF A(I) < A(J) THEN 270
220     LET Y = Y + 1
230     LET J = J + 1
240     IF J <= M1 THEN 320
250     LET X = M2 - Y
260     GO TO 330
270     LET X = X + 1
280     LET I = I + 1
290     IF I <= M THEN 320
300     LET Y = M2 - X
310     GO TO 330
320  NEXT K
330  LET U = M - X
340  LET V = N - Y
350  LET Z = X*V - Y*U
360  LET T = M1*( ABS(Z) - M1/2 ) ↑ 2
370  LET C2 = T/((X+U)*(Y+V)*(X+Y)*(U+V))
```

```
380     PRINT "TWO SAMPLE MEDIAN TEST."
382     PRINT "GROUP 1  "; X; U
385     PRINT "GROUP 2  "; Y; V
390     PRINT "CHI-SQUARE = "; C2
400   STOP
600   FOR I = 1 TO U - L
610       LET X = A(L)
620       LET Q = L
630       FOR J = L + 1 TO U - I + 1
640           IF X >= A(J) THEN 670
650           LET X = A(J)
660           LET Q = J
670       NEXT J
680       IF J = Q THEN 710
690       LET A(Q) = A(J)
700       LET A(J) = X
710   NEXT I
720   RETURN
900 DATA 9999
9999 END
```



EXAMPLE:

PROBLEM

TO COMPARE TWO GROUPS OF DATA: (1,2,3,4,5,6,7) AND (1,2,5,6,7,10,16)

SAMPLE RUN

```
900  DATA  7,8
910  DATA  1,2,3,4,5,6,7
920  DATA  1,2,5,6,7,10,10,16

TWO SAMPLE MEDIAN TEST.

GROUP 1      4      3
GROUP 2      3      5

CHI-SQUARE =    .0585938
```

CONFID 10:

DESCRIPTION

This program calculates confidence limits for data. The calculations are based on the normal curve approximations.

USERS

This program will be of use to individuals setting up statistical bases for surveys or studies. This could include poll or census takers or engineers, structuring data bases.

INSTRUCTIONS

The program is self prompting and will request all necessary inputs. After the program has been loaded into memory type RUN. List the program for detailed instructions.

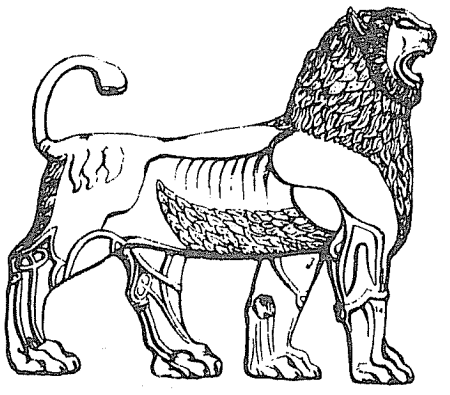
LIMITATIONS

Confid uses the DEF FN__ and FN__ statements through out, starting in line 180. The source code is 2K Bytes long and executes in 4K Bytes in most systems. A routine to convert the FN__ statement for use in a system that does not have this statement is shown in Appendix B.

CONFIN 10

```
10 PRINT
11 PRINT
12 PRINT
100 DATA 5000000,53,8278,5729597,6169114,6554217,7257469
110 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
120 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
130 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
140 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
150 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
160 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
170 DIM X(48)
180 DEF FNQ(U)=M+U*S
190 DEF FND(U)=X(U)-X(U-1)
200 DEF FNB(U)=U-U*(U-1)*(D2/(2*D1)+(U-2)*D3/(6*D1))
210 DEF FNG(U)=1E-3*INT(1E5*U+.5)
220 PRINT
230 PRINT "CONFIDENCE LIMITS FOR A POPULATION PROPORTION BASED ON"
240 PRINT "THE NORMAL CURVE APPROXIMATION.
250 PRINT "                                   WHAT ARE X (NUMBER OF SUCC-"
260 PRINT "ESSES IN SAMPLE), N (SAMPLE SIZE)";
270 INPUT W1,W2
280 DATA .5,.75,.90,.95,.99,.999,.9999,4E44
290 FOR I=0TO48
291 READ X(I)
292 NEXTI
300 LET M=W1/W2
310 LET S=SQR(M*(1-M)/W2)
320 PRINT
330 PRINT "   BEST ESTIMATE OF POPULATION PROPORTION (PCT) =";FNG(M)
340 PRINT
350 PRINT
360 PRINT "CONFIDENCE LIMITS ON POPULATION PROPORTION:"
370 PRINT
380 PRINT "  CONF LEVEL","   LOWER LIM","   UPPER LIM"
390 PRINT
400 READ P
410 IF P=4E44 THEN 9999
420 LET A1=0.5*(1+P)
430 GOSUB 460
440 PRINT 100*P,FNG(FNQ(-A2)-1/(2*W2)),FNG(FNQ(A2)+1/(2*W2))
450 GOTO 400
460 IF A1>0.5 THEN 510
470 LET A1=1-A1
480 GOSUB 540
490 LET A2=-Q
500 GOTO 530
```

```
510 GOSUB 540
520 LET A2=Q
530 RETURN
540 LET Z=1E7*A1
550 FOR I=0 TO 45
560 IF Z<X(I) THEN 580
570 NEXT I
580 LET D1=FND(I)
590 LET E1=FND(I+1)
600 LET D2=E1-D1
610 LET D3=FND(I+2)-E1-D2
620 LET U=(Z-X(I-1))/D1
630 LET Q=FNB(FNB(U))
640 LET Q=0.1*(Q+I-1)
650 RETURN
9999 END
```



DESCRIP:

DESCRIPTION

This program calculates a number of descriptive statistical quantities for a single data set.

USERS

Statisticians and engineers will find the most use for this program.

INSTRUCTIONS

Your data must be entered into the program in the form of data statements, before the program is run. Start entering your data in line 100. The program will hold up to 300 data points. After the data is entered, type RUN. The program will then ask for the Frequency Distributions: L,U. Where "L" is the lower limit of your data and "U" is the upper range of your data. Enter these end points and the program will do the rest.

LIMITATIONS

Descrip contains a Restore statement in line 730 and FN__ statements starting in line 820. The source code stores in 5K and executes in 12K. The execution space may be reduced by reducing the table storage in the DIM statements.

DESCRIP

```
5 REM DESCRIP
10 REM DO NOT RESEQUENCE

700 READ U
720 IF  U = 1E38 THEN 2500
730 RESTORE
740 PRINT
750 PRINT"TYPICAL INTERVAL FOR FREQUENCY DISTRIBUTIONS:";
760 PRINT" L,U =";
770 INPUT W1,W2
780 PRINT
790 PRINT
800 DIM X(300),S(4),Y(4),Z(4)
810 DIM C(50),G(50),E(50),F(50)
820 DEF FNR(M)=.001*INT(1000*M+.5)
830 DATA 1E38,0,0,0,0,0,0,0
840 LET I = 0
850 LET I = I+1
860 READ X(I)
870 IF  X(I) <> 1E38 THEN 850
880 LET N1 = I-1
890 LET N = N1-1
900 FOR I=1 TO 4
910 LET S(I) = 0
920 NEXT I
930 FOR I = 1 TO N1
940 LET Y(1) = X(I)
950 LET Y(2) = X(I)↑2
960 LET Y(3) = Y(1)*Y(2)
970 LET Y(4) = Y(2)↑2
980 FOR J=1 TO 4
990 LET S(J)=S(J)+Y(J)
1000 NEXT J
1010 NEXT I
1020 LET S = S(1)
1030 FOR I = 1 TO 4
1040 LET Y(I)=(1/N1)*S(I)
1050 NEXT I
1060 LET S(2) = Y(2) - Y(1)↑2
1070 LET S(3) = Y(3) - 3*Y(1)*Y(2) + 2*Y(1)↑3
1080 LET S(4) = Y(4) - 4*Y(3)*Y(1) + 6*Y(2)*Y(1)↑2 - 3*Y(1)↑4
1090 LET Y(2) = SQR(S(2))
1100 LET Y(3) = S(3) / (S(2)*Y(2))
1110 LET Y(4) = S(4)/S(2)↑2 - 3
1120 PRINT
1130 PRINT "S U M M A R Y    S T A T I S T I C S"
1140 PRINT
```

```
1150 PRINT"    NUMBER OF VARIATES ="¦N1
1160 PRINT"       ARITHMETIC MEAN ="¦Y(1)
1170 PRINT"    STANDARD DEVIATION ="¦Y(2)
1180 PRINT"              VARIANCE ="¦S(2)
1190 PRINT"    COEFF OF VAR (PCT) =  "¦FNR(100*Y(2)/Y(1))
1200 PRINT"     STANDARD SKEWNESS = "¦FNR(Y(3))
1210 PRINT"       STANDARD EXCESS = "¦FNR(Y(4))
1220 PRINT
1230 PRINT
1240 REM  SORT
1250 FOR I = 1 TO N
1260 FOR J = I+1 TO N1
1270 IF X(I)<X(J) THEN  1310
1280 LET Y1=X(I)
1290 LET X(I)=X(J)
1300 LET X(J)=Y1
1310 NEXT J
1320 NEXT I
1330 PRINT"O R D E R   S T A T I S T I C S"
1340 PRINT
1350 PRINT"      SMALLEST VARIATE ="¦X(1)
1360 LET P1=10
1370 GOSUB 2360
1380 LET T1=P2
1390 PRINT"          LOWER DECILE ="¦T1
1400 LET P1=25
1410 GOSUB 2360
1420 LET T2=P2
1430 PRINT"        FIRST QUARTILE ="¦T2
1440 LET P1=50
1450 GOSUB 2360
1460 PRINT"                MEDIAN ="¦P2
1470 LET P5=P2
1480 LET P1=75
1490 GOSUB 2360
1500 LET T3=P2
1510 PRINT"        THIRD QUARTILE ="¦T3
1520 LET P1=90
1530 GOSUB 2360
1540 LET T4=P2
1550 PRINT"          UPPER DECILE ="¦T4
1560 PRINT"       LARGEST VARIATE ="¦X(N1)
1570 PRINT
1580 LET  U=X(N1)-X(1)
1590 PRINT"           TOTAL RANGE ="¦U
1600 PRINT"          DECILE RANGE ="¦T4-T1
1610 PRINT"   SEMI-QUARTILE RANGE ="¦(T3-T2)/2
1620 PRINT"     BOWLEY'S SKEWNESS = "¦FNR((T3+T2-2*P5)/(T3-T2))
1630 PRINT"      PEARSON SKEWNESS = "¦FNR(3*(Y(1)-P5)/Y(2))
1640 PRINT
1650 PRINT
```

```
1660 PRINT
1670 LET D=ABS(W2-W1)
1680 IF D=0 THEN  2510
1690 LET  Y1=W1-INT((W1-X(1))/D+ 1.99999)*D
1700 LET  L=INT((X(N1)-Y1+.000001)/D)+1
1710 IF  L>50  THEN  2450
1720 FOR I=1 TO L+2
1730 LET  C(I)=Y1+ (I-1)*D
1740 LET  F(I)=0
1750 LET  E(I)=0
1760  NEXT  I
1770 PRINT"      F R E Q U E N C Y     D I S T R I B U T I O N"
1780 PRINT
1790 PRINT"                      UP TO BUT"," ","   PERCENT"
1800 PRINT"         FROM      NOT INCLUDING   FREQUENCY","   FREQUENCY"
1810 PRINT
1820 FOR I = 1 TO N1
1830 LET H=INT((1+1E-8)*(X(I)-C(1))/D) + 2
1840 LET F(H)=F(H)+1
1850 LET E(H)=E(H)+X(I)
1860 NEXT I
1870 LET  J = 1
1880 LET J=J+1
1890 LET G(J)=100*F(J)/N1
1900 PRINT C(J-1),C(J),F(J),FNR(G(J))
1910 IF  J=L+2  THEN  1930
1920 GOTO 1880
1930 FOR J = 3 TO L+1
1940 LET F(J)=F(J-1)+F(J)
1950 LET G(J)=100*F(J)/N1
1960 LET E(J)=E(J-1)+E(J)
1970 NEXT J
1980 PRINT
1990 PRINT
2000 PRINT"      C U M U L A T I V E     D I S T R I B U T I O N"
2010 PRINT
2020 PRINT"                     NUMBER LESS     % LESS           VARIATE SUM %"
2030 PRINT"       VALUE         THAN VALUE      THAN VALUE     LESS THAN VALUE"
2040 PRINT
2050 LET  J = 1
2060 LET J=J+1
2070 LET E(J)=100*E(J)/S
2080 PRINT C(J),F(J),FNR(G(J)),FNR(E(J))
2090 IF  J=L+1  THEN  2110
2100 GOTO 2060
2110 PRINT
2120 PRINT
2130 PRINT"O R D E R E D     A R R A Y"
2140 PRINT
2150 LET M=INT(N1/4-.05)+1
2160 LET H=4*(M-1)
```

```
2170 LET L=N1-H
2180 FOR I = 1 TO M-1
2190 LET K1=I+M
2200 IF L=1 THEN  2230
2210 LET K2=I+2*M
2220 GOTO 2240
2230 LET K2=I+2*M-1
2240 IF (L-1)*(L-2)=0 THEN  2270
2250 LET K3=I+3*M
2260 GOTO 2280
2270 LET K3=I+3*M-2/L
2280 PRINT X(I),X(K1),X(K2),X(K3)
2290 NEXT I
2300 FOR I = 2 TO L+1
2310 PRINT X((I-1)*M),
2320 NEXT I
2330 PRINT
2340 STOP
2350 REM "SUBROUTINE"
2360 LET G5 = P1*(N1+1)/100
2370 LET  P2 = X(1)
2380 IF G5<1 THEN  2440
2390 LET P2=X(N1)
2400 IF G5>N1 THEN  2440
2410 LET Q5 = INT(G5)
2420 LET  H5=G5-Q5
2430 LET P2=H5*X(Q5+1)+(1-H5)*X(Q5)
2440 RETURN
2450 PRINT
2460 PRINT"INTERVAL IS TOO SMALL.  MAXIMUM IS 50 CLASSES"
2470 PRINT"RESPECIFY L,U";
2480 INPUT W1,W2
2490 GOTO 1640
2500 PRINT
2502 PRINT
2510 END
```

SAMPLE RUN

```
100 DATA 261.4, 270.8, 265.4, 261.4, 258.1, 252.1, 268.3, 250.3, 272.3
101 DATA 262.8, 255.5, 249.6, 280.9, 270.3, 263.2, 258.3, 256.3, 259.3
102 DATA 270.1, 259.3, 253.2, 266.4
RUN


TYPICAL INTERVAL FOR FREQUENCY DISTRIBUTIONS: L,U =  ?260,280


        SUMMARY  STATISTICS

  NUMBER OF VARIATES    =  22
  ARITHMETIC MEAN       =  262.0591
  STANDARD DEVIATION    =  7.784011
  VARIANCE              =  60.59082
  COEFF. OF VAR (PCT)   =  2.97
  STANDARD SKEWNESS     =  0.383
  STANDARD EXCESS       =  -0.176


         ORDER  STATISTICS

  SMALLEST VARIATE      =  249.6
  LOWER DECILE          =  250.84
  FIRST QUARTILE        =  256.1
  MEDIAN                =  261.4
  THIRD QUARTILE        =  268.75
  UPPER DECILE          =  271.85
  LARGEST VARIATE       =  280.9


  TOTAL RANGE           =  31.3
  DECILE RANGE          =  21.01
  SEMI-QUARTILE RANGE   =  6.324999
  BOWLEY'S SKEWNESS     =  0.162
  PEARSON'S SKEWNESS    =  0.254
```

DIFFER:

DESCRIPTION

Differ calculates confidence limits on two data groups. The limits are determined from the difference between the means of the two groups.

USERS

Users of Differ will be mainly restricted to people analyzing or studying data sets or samples.

INSTRUCTIONS

Your data must be entered in data statements before the program is run. Enter your data in the following format:

```
    200  DATA H1, N1, M1, S1, H2, N2, M2, S2
```

where

H1 - is the size of the population. (if the population is infinite let H1 = Ø)
N1 - is the number of points in the first data set
M1 - is the mean of the points in the first data set
S1 - is the standard deviation of the first data set
H2 - is the size of the population for the second set of data
N2 - is the number of points in the second data set
M2 - is the mean of this data
S2 - is the standard deviation for the data in the second set

After the data is entered type RUN. The program may be listed for detailed instructions.

LIMITATIONS

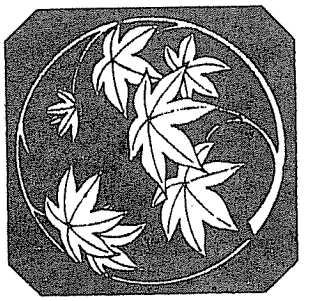
Starting in line 19 the DEF FN__ statement is used. The FN__ statement is used throughout this program. Differ requires 4K Bytes for storage and executes in 6K Bytes of memory.

DIFFER

```
10 REM DIFFER
11 DATA 5000000,5398278,5792597,6179114,6554217,6914625,7257469
12 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
13 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
14 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
15 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
16 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
17 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
18 DIM X(49)
19 DEF FNQ(U)=M3+U*R3
20 DEF FND(U)=X(U)-X(U-1)
21 DEF FNB(U)=U-U*(U-1)*(D2/(2*D1)+(U-2)*D3/(6*D1))
22 DEF FNZ(U)=1+((U↑2)+1)/(4*D)+(((U↑2)+3)*(5*(U↑2)+1))/(96*D↑2)
25 FOR I=1 TO 49
30 READ X(I)
35 NEXT I
202 DATA 1E37
203 DATA .5,.75,.9,.95,.99,.999,.9999,.99999,1E37
204 READ H1,N1,M1,S1,H2,N2,M2,S2
205 IF H1=1E37 THEN 352
206 READ I
207 IF H1<>0 THEN 210
208 LET H1=1E20
210 IF H2<>0 THEN 214
212 LET H2=1E20
214 LET M3=M1-M2
216 LET T1=S1*SQR((N1*(H1-1))/(H1*(N1-1)))
218 LET T2=S2*SQR((N2*(H2-1))/(H2*(N2-1)))
220 LET R1=T1*SQR((H1-N1)/(N1*(H1-1)))
222 LET R2=T2*SQR((H2-N2)/(N2*(H2-1)))
224 LET R5=R1↑2
226 LET R6=R2↑2
228 LET R3=SQR(R5+R6)
230 LET W=R5/(R5+R6)
232 LET D=((N1-1)*(N2-1))/((N2-1)*W↑2+(N1-1)*(1-W)↑2)
234 LET D=.1*INT(10*D+.5)
236 PRINT
238 PRINT "       STATISTIC     ","SAMPLE 1","SAMPLE 2"
240 PRINT
242 PRINT "SAMPLE MEAN          ",M1,M2
244 PRINT "SAMPLE VARIANCE      ",S1↑2,S2↑2
246 PRINT "SAMPLE STD DEVIATION",S1,S2
248 PRINT "SAMPLE SIZE          ",N1,N2
250 PRINT "POPULATION SIZE      ",
252 IF H1<>1E20 THEN 258
254 PRINT "INFINITE",
256 GOTO 260
```

```
250 PRINT H1,
260 IF H2<>1E20 THEN 266
262 PRINT "INFINITE"
264 GOTO 268
266 PRINT H2
268 PRINT "ESTIM POPN STD DEV  ",T1,T2
270 PRINT "STD ERROR OF MEAN   ",R1,R2
272 PRINT
274 PRINT "DIFF BETWEEN MEANS    ","          ";M3
276 PRINT "STD ERROR OF DIFF     ","          ";R3
278 PRINT "DEGR OF FREEDOM (DIFF)","          ";D
280 PRINT
282 PRINT
284 PRINT "CONFIDENCE LIMITS ON DIFFERENCE BETWEEN MEANS:"
286 PRINT
288 PRINT "CONF LEVEL","LOWER LIM","UPPER LIM"
290 PRINT
292 READ P
294 IF P=1E37 THEN 386
296 LET A1=0.5*(1+P)
298 GOSUB 310
300 IF D=0 THEN 306
302 REM  FNZ CONVERTS STUDENT'S T TO A NORMAL DEVIATE:
304 LET A2=A2*FNZ(A2)
306 PRINT 100*P,FNQ(-A2),FNQ(A2)
308 GOTO 292
310 IF A1>0.5 THEN 320
312 LET A1=1-A1
314 GOSUB 328
316 LET A2=-Q
318 GOTO 324
320 GOSUB 328
322 LET A2=Q
324 RETURN
326 REM  REVERSE INTERPOLATION FOR STD NORMAL DEVIATE:
328 LET Z=1E7*A1
330 FOR I=1 TO 46
332 IF Z<X(I) THEN 336
334 NEXT I
336 LET D1=FND(I)
338 LET E1=FND(I+1)
340 LET D2=E1-D1
342 LET D3=FND(I+2)-E1-D2
344 LET U=(Z-X(I-1))/D1
346 LET Q=FNB(FNB(U))
348 LET Q=0.1*(Q+I-2)
350 RETURN
352 PRINT
354 PRINT "THIS PROGRAM COMPUTES CONFIDENCE LIMITS FOR"
356 PRINT "THE DIFFERENCE BETWEEN TWO POPULATION MEANS,"
358 PRINT "BASED ON DATA SUPPLIED FOR TWO SAMPLES, ONE"
```

```
360 PRINT "FROM EACH POPULATION.  TO USE, TYPE:"
362 PRINT
364 PRINT "   200 DATA  M1,N1,M1,S1, M2,N2,M2,S2"
366 PRINT "   RUN"
368 PRINT
370 PRINT "WHERE  M1 = SIZE OF POPULATION 1 (LET M1"
372 PRINT "                EQUAL ZERO IF POPN IS INFINITE)"
374 PRINT "       N1 = SIZE OF SAMPLE 1"
376 PRINT "       M1 = ARITHMETIC MEAN OF SAMPLE 1"
378 PRINT "       S1 = STANDARD DEVIATION OF SAMPLE 1"
380 PRINT "              (BASED ON DIVISOR OF N1)"
382 PRINT
384 PRINT "AND M2,N2,M2,S2 ARE THE SAME FOR SAMPLE 2."
386 END
```



ENGINE:

DESCRIPTION

This program computes various parameters for the Otto cycle engine. The values are generated using a CFR engine. Inputs for the program are from the user during operation.

USERS

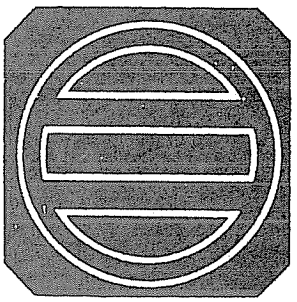
This program can be used by auto enthusiasts but will probably be a bit too technical for all but those with an engineering background.

INSTRUCTIONS

After the program has been loaded into memory type RUN. The program will prompt for all required inputs. The program should be listed for additional instructions.

LIMITATIONS

Engine should store and execute in most Basic speaking systems with 6K of available memory.



ENGINE

```
15 PRINT"DO YOU WANT  INSTRUCTIONS?  ENTER 1 FOR YES AND 0 FOR NO"
16 INPUT Q1
17 IF Q1 = 1 THEN 100
18 IF Q1 = 0 THEN 122
19 PRINT "PLEASE  RESPOND WITH 1 OR 0"
20 GO TO 16
100 PRINT"          THIS IS A PROGRAM WHICH WILL CALCULATE THE"
101 PRINT"          FOLLOWING ITEMS FOR THE OTTO CYCLE USING THE"
102 PRINT"          CFR ENGINE"
103 PRINT"              1. AIR FLOW INTO ENGINE"
104 PRINT"              2. FUEL FLOW INTO ENGINE"
105 PRINT"              3. AIR FUEL RATIO"
106 PRINT"              4. BRAKE HORSEPOWER"
107 PRINT"              5. FRICTION HORSEPOWER"
108 PRINT"              6. INDICATED HORSEPOWER"
109 PRINT"              7. BRAKE THERMAL EFFICIENCY"
110 PRINT"              8. INDICATED THERMAL EFFICIENCY"
111 PRINT"              9. BRAKE SPECIFIC FUEL CONSUMPTION"
112 PRINT"             10. INDICATED SPECIFIC FUEL CONSUMPTION"
113 PRINT"             11. IDEAL THERMAL EFFICIENCY"
114 PRINT"             12. RELATIVE EFFICIENCY"
115 PRINT"             13. VOLUMETRIC EFFICIENCY"
116 PRINT"          THE ABOVE VALUES ARE CALCULATED FROM DATA WHICH IS"
117 PRINT"          REQUESTED BY THE COMPUTER AS IT IS REQUIRED"
118 REM
119 REM
120 REM        ******AIR VOLUME RATE CORRECTED FOR TEMP. AND PRES.******
121 REM
122 PRINT "PRESSURE DROP ACROSS LAMINAR METER (IN. OF WATER) =";
123 INPUT P1
124 IF P1>1 THEN 734
126 IF P1<0 THEN 734
128 PRINT "BAROMETRIC PRESSURE (INCHES OF HG) =";
130 INPUT P0
132 IF P0>31 THEN 738
134 IF P0<28 THEN 738
140 PRINT "ROOM TEMPERATURE (FAHRENHEIT) =";
150 INPUT F1
152 IF F1>100 THEN 742
154 IF F1< 60 THEN 742
160 LET Q0=P1*12.5*(.9358+(P0-28)*.0334)*(1-(F1-70)*.0032)
170 PRINT
180 PRINT "AIR FLOW INTO ENGINE =" Q0; "CUBIC FEET PER MINUTE"
190 PRINT
191 REM
192 REM
193 REM             ******SPECIFIC WEIGHT OF AIR******
```

```
194 REM
200 LET S0=P0*.491*144/(53.35*(F1+460))
201 REM
202 REM
203 REM           ******AIR WEIGHT RATE INTO ENGINE******
204 REM
210 LET M0=Q0*S0
220 PRINT "AIR FLOW INTO ENGINE =" M0; "POUNDS PER MINUTE"
230 PRINT
231 REM
232 REM
233 REM      ******FUEL WEIGHT RATE INTO ENGINE******
234 REM
240 PRINT "TIME REQUIRED FOR 21.5 CC OF FUEL (MINUTES) =";
250 INPUT T2
252 IF T2>3 THEN 746
254 IF T2<.2 THEN 746
260 PRINT "SPECIFIC GRAVITY OF FUEL =";
270 INPUT S1
272 IF S1>1 THEN 750
274 IF S1<.5 THEN 750
275 PRINT
280 LET M1=21.5*62.4*S1/(T2*(2.54)↑3*1728)
290 PRINT "FUEL FLOW INTO ENGINE =" M1; "POUNDS PER MINUTE"
300 PRINT
301 REM
302 REM
303 REM           ******AIR FUEL RATIO******
304 REM
310 LET R0=M0/M1
320 PRINT "AIR FUEL RATIO =" R0
330 PRINT
331 REM
332 REM
333 REM           ******BRAKE HORSEPOWER******
334 REM
340 PRINT "BRAKE WATTMETER READING IN KILOWATTS =";
350 INPUT P2
352 IF P2>2 THEN 754
354 IF P2<.2 THEN 754
360 LET P2=P2/.746
365 PRINT
370 PRINT "BRAKE HORSEPOWER ="P2
380 PRINT
381 REM
382 REM
383 REM           ******FRICTION HORSEPOWER******
384 REM
390 PRINT "FRICTION WATTMETER READING IN KILOWATTS =";
400 INPUT P3
402 IF P3>3 THEN 758
```

```
404 IF P3<1 THEN 758
410 LET P3=P3/.746
415 PRINT
420 PRINT "FRICTION HORSEPOWER =" P3
430 PRINT
431 REM
432 REM
433 REM           ******INDICATED HORSEPOWER******
434 REM
440 LET P4=P2+P3
450 PRINT "INDICATED HORSEPOWER =" P4
460 PRINT
461 REM
462 REM
463 REM           ******BRAKE THERMAL EFFICIENCY******
464 REM
470 PRINT "HIGHER HEATING VALUE OF FUEL =";
480 INPUT H0
482 IF H0>21000 THEN 762
484 IF H0<19000 THEN 762
490 LET E2=P2*42.42/(M1*H0)
495 PRINT
500 PRINT "BRAKE THERMAL EFFICIENCY =" E2
510 PRINT
511 REM
512 REM
513 REM           ******INDICATED THERMAL EFFIGIENCY******
514 REM
520 LET E4=P4*42.42/(M1*H0)
530 PRINT "INDICATED THERMAL EFFICIENCY =" E4
540 PRINT
541 REM
542 REM
543 REM           ******BRAKE SPECIFIC FUEL CONSUMPTION******
544 REM
550 LET S2=M1*60/P2
560 PRINT "BRAKE SPECIFIC FUEL CONSUMPTION =" S2; "LBS/HP-HR"
570 PRINT
571 REM
572 REM
573 REM           ******INDICATED SPECIFIC FUEL CONSUMPTION******
574 REM
580 LET S4=M1*60/P4
590 PRINT "INDICATED SPECIFIC FUEL CONSUMPTION =" S4; "LBS/HP-HR"
600 PRINT
601 REM
603 REM           ******IDEAL THERMAL EFFICIENCY******
604 REM
610 PRINT "COMPRESSION RATIO =";
620 INPUT R1
622 IF R1>10 THEN 766
```

```
624 IF R1< 4 THEN 766
630 LET E5=1-R1↑(-.4)
635 PRINT
640 PRINT "IDEAL THERMAL EFFICIENCY =" E5
650 PRINT
651 REM
652 REM
653 REM           ******RELATIVE EFFICIENCY******
654 REM
655 LET E7=E4/E5
656 PRINT "RELATIVE EFFICIENCY =" E7
657 PRINT
658 REM
660 REM           ******VOLUMETRIC EFFICIENCY******
661 REM
662 PRINT "CYLINDER BORE (ASSUMED) = 3.25 INCHES"
680 PRINT "PISTON STROKE (ASSUMED) = 4.5 INCHES"
700 PRINT "ENGINE SPEED (900<=RPM<=1000) =";
710 INPUT N0
712 IF N0>1000 THEN 778
714 IF N0< 900 THEN 778
715 PRINT
720 LET E6=00*1728/(3.14159*(3.25)↑2/4*4.5*N0)
730 PRINT "VOLUMETRIC EFFICIENCY =" E6
732 GO TO 920
734 GOSUB 900
736 GO TO 120
738 GOSUB 900
740 GO TO 128
742 GOSUB 900
744 GO TO 140
746 GOSUB 900
748 GO TO 240
750 GOSUB 900
752 GO TO 260
754 GOSUB 900
756 GO TO 340
758 GOSUB 900
760 GO TO 340
762 GOSUB 900
764 GO TO 470
766 GOSUB 900
768 GO TO 610
778 GOSUB 900
780 GO TO 700
900 PRINT "DATA OUTSIDE LIMITS"
910 RETURN
920 STOP
99998 REM
99999 END
```

FOURIER:

DESCRIPTION

This program evaluates functions which are time dependent and are sums of exponentials and sine cosine terms.

USERS

Engineers and other technically oriented professionals could find use for this type of program.

INSTRUCTIONS

The data must be entered into data statements before Fourier is run. Data should be entered in the following form:

```
     2Ø  DATA  NP,N1,N2,TØ,Delta-T,Sigma
```

where

NP - is the total number of points to be calculated
N1 - is the number of exponential terms
N2 - is the number of sine cosine exponential terms
TØ - is the time of the first point
Delta-T - is the time between points
Sigma - is the standard deviation of the noise (this is usually set to Ø)

Then enter the coefficients of the equation that satisfies the time function C*EXP(-S*T)

Then enter the coefficients of the following equation that satisfy the time function. (A*COS(W*T)+B*SIN(W*T))*EXP(-G*T)

with	A1,A2,A3,.....on line 56	B1,B2,B3,.....on line 60
and	C1,C2,C3,.....on line 50	W1,W2,W3,.....on line 63
and	S1,S2,S3,.....on line 53	G1,G2,G3,.....on line 66

After the data is entered type RUN. List the program for detailed instructions.

LIMITATIONS

The EXP( ) statement is used extensively throughout the program, starting in line 170. The source code requires 3K Bytes of memory for storage and execution requires 9K Bytes of available memory.

FOURIER

```
5 REM
7 DIM D(500), X(20)

10 REM EVALUATES EXPONENTIAL FUNCTIONS
20 GO TO 596
30 READ N,N1,N2,T1,T2,C1
35 IF N=4E44THEN600
40 LET M=2*N1+4*N2
80 FOR J=1 TO M
90 READ X(J)
100 NEXT J
120 FOR I=1 TO N
130 LET D(I)=0
140 LET T=T1+(I-1)*T2
150 FOR J=1 TO (N1+N2)
160 IF J>N1 THEN 190
170 LET D(I)=D(I)+X(J)*EXP(-X(N1+J)*T)
180 GO TO 230
190 LET L=N1+J
200 LET Y=(COS(X(2*N2+L)*T))*EXP(-X(3*N2+L)*T)
210 LET Z=(SIN(X(2*N2+L)*T))*EXP(-X(3*N2+L)*T)
220 LET D(I)=D(I)+X(L)*Y+X(N2+L)*Z
230 NEXT J
240 LET D(I)=D(I)+C1*(RND(X)-0.5)*SQR(12)
250 NEXT I
260 PRINT
270 PRINT
280 PRINT
290 PRINT"NOISE SIGMA =";C1
300 IF N1=0 THEN 390
310 PRINT
320 PRINT"TERMS OF FORM   C*EXP(-S*T)   ARE:"
330 PRINT
340 PRINT" ","        C","         S"
350 PRINT
360 FOR J=1 TO N1
370 PRINT" ",X(J),X(N1+J)
380 NEXT J
390 IF N2=0 THEN 510
395 PRINT
400 PRINT"TERMS OF FORM   (A*COS(W*T)+B*SIN(W*T))*EXP(-G*T)   ARE:"
450 PRINT
460 PRINT" ","         A","         B","         W","         G"
470 PRINT
480 FOR J=(2*N1+1) TO (2*N1+N2)
490 PRINT " ",X(J),X(N2+J),X(2*N2+J),X(3*N2+J)
500 NEXT J
```

```
510 PRINT
520 PRINT"FIRST POINT AT T1,SPACING=T2"
530 PRINT "T1=";T1,   "T2=";T2
540 PRINT
550 PRINT "DATA POINTS ARE:"
560 PRINT
570 FOR I=1 TO N
580 PRINT D(I),
590 NEXT I
595 GOTO 9999
596 PRINT"        INSTRUCTIONS "
600 PRINT"THIS PROGRAM EVALUATES TIME FUNCTIONS"
610 PRINT"WHICH ARE SUMS OF EXPONENTIALS AND"
620 PRINT"EXPONENTIAL SINE-COSINE TERMS."
630 PRINT
640 PRINT"INPUT DATA MUST BE ENTERED AS FOLLOWS:"
650 PRINT"20 DATA NP,N1,N2,T0,DELTA-T,SIGMA"
660 PRINT"   WHERE"
670 PRINT"   NP=TOTAL NUMBER OF POINTS TO BE COMPUTED"
680 PRINT"   N1=NUMBER OF EXPONENTIAL TERMS"
690 PRINT"   N2=NUMBER OF SINE-COSINE EXPONENTIAL TERMS"
700 PRINT"   T0=TIME OF FIRST POINT"
710 PRINT"   DELTA-T=TIME BETWEEN POINTS"
720 PRINT"   SIGMA=STANDARD DEVIATION OF THE NOISE"
730 PRINT" (IF ADDITIVE NOISE IS NOT DESIRED, SIGMA=0 )"
740 PRINT
750 PRINT"PARAMETERS OF THE FUNCTION ARE ENTERED AS FOLLOWS:"
760 PRINT"  41 DATA C(1),C(2),C(3),...,S(1),S(2),S(3),..."
770 PRINT"  60 DATA A(1),A(2),A(3),...,B(1),B(2),B(3),..."
780 PRINT"  79 DATA W(1),W(2),W(3),...,G(1),G(2),G(3),..."
790 PRINT "STATMENT NUMBERS BETWEEN 41 AND 79 INCLUSIVE"
800 PRINT"MAY BE USED"
810 PRINT
820 PRINT"MAXIMUM NUMBER OF POINTS PERMISSABLE IS"
830 PRINT"500 AND (2N1+4N2=20)."
840 PRINT
850 PRINT"THE COMPUTATION OF E↑-X,WHERE X IS LARGE MAY RESULT"
860 PRINT"IN EXCESSIVE RUNNING TIME."
9998 REM
9999 END
```

SAMPLE RUN

```
20  DATA  30,3,1,0,.001,0
50  DATA  -1.78189E-2,1.66119E-2,4.50666E-3,1.5662,16.2565,136.889
60  DATA  15.438,-.37221,491.834,200.474

NOISE SIGMA =  0

TERMS OF FORM  C*EXP(-S*T)        ARE:

                    C                          S

               -0.0178189                   1.5662
                0.0166119                  16.2565
                0.0045067                  136.889

TERMS OF FORM  (A*COS(W*T)+B*SIN(W*T))*EXP(-G*T)       ARE:

                    A            B           W            G

                 15.438      -0.37221     491.834      200.474

FIRST POINT AT  T1, SPACING = T2
T1 =   0             T2 =    0.001

DATA POINTS ARE:

   15.4413      10.99474      5.521522     0.6030367     -2.827604
 -4.483233     -4.574454     -3.597412     -2.126857     -0.6597066
  0.4720381     1.122184      1.302155      1.126335      0.7520941
  0.3298639    -0.028716     -0.263328     -0.3617165    -0.3465941
 -0.2589358    -0.1426456    -0.0336229     0.0457423     0.0871895
  0.0937954     0.0755308     0.0447195     0.0124639    -0.013472
```

HORSE

DESCRIPTION

A champion if there ever was one, you can tell by the lines. This detailed picture of a horse's head and neck should be held at arms length when viewing. The source code is 4K Bytes long and the program executes in 5K Bytes of memory.

HORSE

```
001 PRINT
02PRINT
03PRINT
04PRINT
05PRINT
06PRINT
07PRINT
08PRINT
09PRINT
10PRINT
11PRINT"                MMMM                  MMMMM    "
12PRINT"               MMMMX                 MMMMMMMM      "
13PRINT"               MMMMMX                MMMMMMMMM"
14PRINT"               MMMMMXX               MMMMMMMMM "
15PRINT"               MMMMMXX               MMMMMMMMM"
16PRINT"               MMMMMMXX              MMMMMMMMM"
17PRINT"               MMMMMMXXX             MMMMMMMM"
18PRINT"               MMMMMMXXX               MMMMMMM   "
19PRINT"               MMMMMMXXX MMMMMMMMMMMMMMMM    "
20PRINT"                MMMMMXXXMMMMMMMMMMMMMMMXX     "
21PRINT"                 MMMMMMMMMMMMMMMMMMMMMMMXX         "
22PRINT"                  MMMMMM MMMMMXXMXXMMMXX    "
23PRINT"                    MMX  OMMMMMXXXXMMXMMX   "
24PRINT"                    MM   MMMMMMXXMXMMMXMMMO    "
25PRINT"                    MMMM MMMMMMXXMXXMMXXMMMM    "
26PRINT"                   MMMMM MMMMMMXXMMXMXXXMMMM    "
27PRINT"                   MMMMMMMXMMMMXXMMMXXMXXXMMMMM"
28PRINT"                 MM MMM M  MMMMMXMMMXXMXXXXXMMMM"
```

```
29PRINT"            MM  MMM M   MMMMMMMMMXMMXXXXXMMMM "
30PRINT"           MMM  MM      MMMMMXXXXXMMXXXXXXMMMM   "
31PRINT"          MMMM   M     MMMMMXXXXXWWWWXXXXXXWWWWW    "
32PRINT"         MXXMMX  M    XMMMXXXXXXXWWWWWXXXXXWWWWWW    "
33PRINT"        XMMXXMO      XMMMMMXXXXXWWWWWWMXXXXXWWWWWO "
34PRINT"        MMXMMM      MMMMWMMMMMWWWXWWWWWWWXXXXXXWWWWW "
35PRINT"        XMMXMMO    XMMMMWWMMMWWOWWWWWWWWWMXXXXXWWWWWWO "
36PRINT"        MMXMMM     MMMMMWWWWWWWWWWWWMMMMMMXXXXXWWWWWWW     "
37PRINT"        MMMMMM    MMMMMMMWWWMMMMMMMMMMMMMMXXXXX WWWWWWX"
38PRINT"         MMMM     MMMMMMMWWWMMMMMMMMMMMMMMXXXXX WWWWWWX "
39PRINT"          MMM     MMMMMXXXMMMMMMMMMMMMMMMMXXXX MMMMMMMM "
40PRINT"          MM      MMMMMXXXXWWWWWXMMMMMMMMMXXXX WWWWWWWWW      "
41PRINT"          MM     XMMMXXXXXXXWWWWXMMMMMMMMMMXXXXX   WWWWWWWW    "
42PRINT"          MM     MMMXXWWXWXXXWWWWMMMMMMMMMMMMMXXXM WWWWWWWWW   "
43PRINT"          M      MMXXWWWXXXWWWWWXMMMMMMMMMMMWXXXXM WWWWWWWWW   "
44PRINT"          M     MMXXWWWWXXWWWWWXMMMMMMMMMMMWXXXMMWWW WWWWWWW"
45PRINT"          M     XXXWWWWWXXXWWWWXXXMMMMMMMMMMMWWXXXMWWW WWWWWWWWW "
46PRINT"         M     XXWWWWXXXMWWWWMXXMMMMMMMMMWWWWMMMMMXXWWXWMXWWWWWW    "
47PRINT"        M     XXXWWWWXXXMWWWMXXMMMMMMMMMWWWWMMMMMXXWWWMXWWWWWWW"
48PRINT"        M     XXWXWXXXXXWWWWMMXXMMMMMMMMMWWWWWMMMMM WWWW WWWWWWW  "
49PRINT"       M     XWWXXWXXXXXWWWWXXXMMMMMMMMMWWWWMMMMMMMXWWWXWWWWWWW"
50PRINT"       M    XWWXMMMXXXWWWWXMWMMMMMMMMMMWWWWWWMMMMMM WWWWWWWWWWW"
51PRINT"      MM   XWXXXXMXXXXWWWXXWWWMMMMMMWWWWWWWMMMMMMMMMWWWWWWWWWWWWM"
52PRINT"      M    XXXXXMXXXXMWWXXWWWWWMMMMMMWWWWWWWMMMMMXMMMWWWWWWWWWWW"
53PRINT"      M   XXXXXXMXXXXWWXWWWWWWWWMMMMWWWWWWWWMMMMMMMMMMXW WW WWW"
54PRINT"     MM   XXXXXMMXXXXWXXWWWWWWWWWWWMMMWWWWWWWWWWMMMMMMMMM W WW WWW"
55PRINT"     MM  XXXXXXMXXXXXXXWWWWWWWWWWWWWWWWWWWWWWWWWXXMMMMMMMM W W WWW "
56PRINT"    MMW  MXXXXXWWWWWWWWWWWWWWWWWWWWWWWWWWMMMMMMWXXMMMMMMMMOWOWWWWW   "
57PRINT"    MW  WWWWXXWWXWWWWWWWWWWWWWWWWWWWWWWWWWMMMMWWMMMMMMMMMXM M   WWWW "
58PRINT"    MMXXWWWWXWWWXXXXXXWWWWWWWWWWWWWWWWWWWMMMMWWWXXXMMMMMMM     WMW"
59PRINT"   MM   WWWWWWWWWWXXXXXWWWWWWWWWMWWWWWWWWWWWWWWWWWXXXMMMMMMM    XOXH"
60PRINT"   MM   WWWWWWWWWWXXXWWWWWWWWMMXXWWWWWWWWWWWWWWWMXXMMMMMMMXX   HHXX"
61PRINT" MM     WWWWWWWXWWXXMWWWWWWMMM     WWWWWWWWWWWWXXXMMMMXX    W"
62PRINT" MM     WWW WWWWXWMXWWWWWWWMM      WWWWWWWWWWWWXXXMMMMMM    "
63PRINT" MM    WWW   WWWWWWWMMWWMWMMM      MMMMMMMMMMMXXXXXMXX "
64PRINT" WWWWWWWW WWWWWWWWWWWWMWMM         MMWWWWWWWWWXXXXXMMM "
65PRINT" XWWWWWWMXWWWWWWWWWWWWWMM          XWWWWWWWWWWXXXXXMMX  "
66PRINT"   MWWWWWMW    WWWWMMM              WWWWWWWWWWWXXXX "
67PRINT"    MMMWWWMWWWWWWWWMM               WWWWWWWWWWX XX  "
68PRINT"       WWWWWWWWWW                   WWWWWWW    "
69PRINT
70PRINT
71PRINT
72PRINT
73PRINT
74PRINT
75 END
```

INTEGERS:

DESCRIPTION

Integers generates four numbers that when squared and summed equal the input number.

USERS

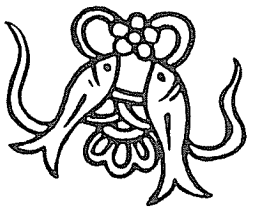
This program can be used as a teaching aid by teachers and parents alike. It can also be used to stimulate an interest in mathematics through a type of game approach.

INSTRUCTIONS

Load the program into memory and type RUN. The program is self prompting and only requires that you enter an integer for it to factor.

LIMITATIONS

Line 80 uses an ABS( ) statement. Except for this statement the program should execute in most systems without incident. Integers requires 1K Byte of memory for storage and executes in 2K Bytes of memory excluding the amount of memory required to store the Basic compiler.

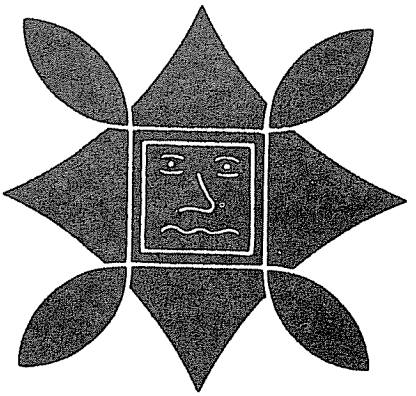


INTEGERS

```
10 PRINT "INTEGERS"
20 PRINT "N = A↑2 + B↑2 + C↑2 + D↑2"
30 PRINT
40 PRINT "N    A     B     C     D"
50 PRINT"-------------------------"
60 PRINT
70 INPUT M
80 LET N=ABS(M)
90 LET R=SQR(N)
100 FOR A = 0 TO R
110 LET A2=A*A
120 LET N1=N-A2
130 FOR B = 0 TO N1
140 LET B2=B*B
150 LET N2=N1-B2
160 FOR C = 0 TO N2
170 LET C2 = C*C
180 IF A2+B2+C2 > N THEN 270
190 LET D = SQR(N-A2-B2-C2)
200 IF D > INT(D) THEN 260
210 IF M>=0 THEN 240
220 PRINT " ";-A;-B;-C;-D
230 GO TO 60
240 PRINT " ";A;B;C;D
250 GO TO 60
260 NEXT C
270 NEXT B
280 NEXT A
290 GO TO 60
300 END
```

SAMPLE RUN

```
  INTEGERS

  N = A↑2 + B↑2 + C↑2 + D↑2

  N     A      B      C      D

  ?12
        0      2      2      2

  ?-452
        0      0    -14    -16

  ?39
        1      1      1      6

  ?STOP
```



LOGIC:

DESCRIPTION

Logic calculates the highest possible conclusion for a specific set of variables. After the calculations are made the program prints truth tables for the various variables.

USERS

This program can be used to teach digital logic to students or to help interested individuals to better understand TTL circuit configurations. Interested groups would include teachers, radio amateurs, engineers, etc.

INSTRUCTIONS

After Logic has been loaded into memory, type RUN. The program is self prompting and will ask for all required inputs. The program should be listed for detailed instructions.

LIMITATIONS

Line 170 contains a Change statement, line 300 a MAT A = CON( ), line 350 a MAT READ, line 450 a Restore statement, and starting in line 540 an ON GOTO statement is used and appears again in lines 1380, 2270, 2300 and 2530. The source code requires 8K Bytes of memory for storage. Logic will require 20K Bytes of memory for execution with the DIM statements as they are presently set. If the dimension statements are reduced the program will execute in less than 20K Bytes of memory, but this will also reduce the number of logic statements the program can handle.

LOGIC

```
15 REM LOGIC
18REM DESCRIPTION--DETERMINES THE STRONGEST POSSIBLE CONCLUSION IN
20REM SPECIFIED VARIABLES WHICH FOLLOWS AS A LOGICAL CONSEQUENCE FROM
22REM A GIVEN SET OF STATEMENTS OF PROPOSITIONAL LOGIC AND PRINTS ITS
24REM TRUTH TABLE.  STATEMENTS CAN CONTAIN THE VARIABLES A, B, ..., T,
26REM PARENTHESES, AND THE CONNECTIVES  - (NOT),  & (AND),  V (OR),
28REM => (IF..THEN), <=> (IF AND ONLY IF), AND  / (NOT BOTH). UNLESS
30REM OTHERWISE INDICATED BY PARENTHESES, THE CONNECTIVES ARE EVALUATED
32REM FROM LEFT TO RIGHT OBSERVING THE PRIORITIES ESTABLISHED BY THE
34REM ORDER OF THE CONNECTIVES IN THE ABOVE LIST.
36REM
38REM-----------------------------------------------------------------
40REM
42REM INSTRUCTIONS--TYPE "RUN".  WHEN THE PROGRAM PRINTS "PREMISE?",
44REM ENTER A STATEMENT OR TYPE "DONE" TO INDICATE THAT ALL PREMISES
46REM HAVE BEEN ENTERED.  AFTER ALL PREMISES HAVE BEEN ENTERED, THE
48REM THE PROGRAM WILL ASK FOR A LIST OF VARIABLES FOR WHICH TO DRAW
50REM CONCLUSIONS.  ENTER THE NAMES OF THESE VARIABLES, ONE AT
52REM A TIME.  ENTER DONE WHEN ALL DESIRED VARIABLES HAVE BEEN
54REM ENTERED.  ALTERNATIVELY, TYPE 'BEST' TO HAVE THE PROGRAM FIND
56REM THE STRONGEST POSSIBLE CONCLUSIONS IN THE FEWEST POSSIBLE
58REM VARIABLES.
60REM
62REM  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
64REM
100 PRINT "-LIST FOR INSTRUCTIONS-"
110 PRINT
120 DIM A(100), F(200)
130 LET L = -1
140 PRINT "PREMISE";
150 INPUT A$
160 IF A$ = "DONE" THEN 270
170 CHANGE A$ TO A
180 LET L = L+2
190 LET F(L-1) = 38
200 LET F(L) = 40
210 FOR I = 1 TO A(0)
220 LET F(L+I) = A(I)
230 NEXT I
240 LET L = L + A(0) + 1
250 LET F(L) = 41
260 GOTO 140
270 LET F(0) = L
280 PRINT
290 DIM O(12), T(120), A$(12)
300 MAT A = CON(100)
310 REM  **********  INITIALIZATION  **********
320 REM
```

```
330 DIM B(100), C(100), G(127)
340 DIM N(100), S(120), V(20), V$(21)
350 MAT READ G
360 LET G(0) = 9
370 DATA 9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
380 DATA 9,11,9,9,9,9,9,9,3,9,1,8,9,9,9,2,9,7,9,9,9,9,9,9,9,9,9,9,9,9,6
390 DATA 5,9,9,9,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10
400 DATA 10,10,10,10,9,4,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
410 DATA 9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
420 FOR I=1 TO 21
430 READ V$(I)
440 NEXT I
450 RESTORE
460 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,F
470 LET P = S = N = F1 = 0
480 REM
490 REM  **********  COMPILATION  **********
500 REM
510 FOR I = 1 TO L
520 LET C = G(F(I))
530 IF C > 11 THEN 800
540 ON C GOTO 550,550,680,680,640,600,680,720,860,780,1080
550 REM    PUT  (  OR  -  INTO SYMBOL CELLAR
560 IF F1 = 1 THEN 1050
570 LET S = S+1
580 LET S(S) = C
590 GOTO 1080
600 REM    CHECK FOR REST OF  =>  AND  <=
610 IF I = L THEN 860
620 LET I = I+1
630 IF F(I) <> 61 THEN 860
640 IF I = L THEN 860
650 LET I = I+1
660 IF F(I) <> 62 THEN 860
670 REM    HANDLE BINARY CONNECTIVES
680 IF F1 = 0 THEN 1050
690 LET F1 = 0
700 GOSUB 880
710 GOTO 560
720 REM    HANDLE  )
730 IF F1 = 0 THEN 1050
740 GOSUB 880
750 IF S = 0 THEN 1050
760 LET S = S-1
770 GOTO 1080
780 REM    HANDLE NEW VARIABLES
790 LET C = G(F(I)) = F(I)
800 REM    HANDLE OLD VARIABLES
810 IF F1 = 1 THEN 1050
820 LET N = N+1
830 LET N(N) = C+36
```

```
840 LET F1 = 1
850 GOTO 1080
860 PRINT "A PREMISE CONTAINS AN ILLEGAL CHARACTER."
870 STOP
880 REM     SUBROUTINE TO COMPILE INSTRUCTIONS
890 IF S = 0 THEN 1030
900 LET D = S(S)
910 IF D = 1 THEN 1030
920 IF D >= C THEN 1030
930 LET P = P+1
940 LET C(P) = D
950 IF D = 2 THEN 1040
960 IF N < 2 THEN 1050
970 LET N = N-1
980 LET B(P) = N(N+1)
990 LET A(P) = N(N)
1000 LET N(N) = P
1010 LET S = S-1
1020 GOTO 880
1030 RETURN
1040 IF N > 0 THEN 990
1050 PRINT "A PREMISE IS NOT WELL-FORMED."
1060 STOP
1070 REM    END OF COMPILATION LOOP
1080 NEXT I
1090 REM
1100 REM    CLEAN OUT SYMBOL CELLAR AND CHECK FOR ERRORS
1110 IF F1 = 0 THEN 1050
1120 IF S = 0 THEN 1160
1130 LET C = 8
1140 GOSUB 880
1150 IF S > 0 THEN 1050
1160 IF N <> 1 THEN 1050
1170 LET R = N(1)
1180 REM     FIND VARIABLES OCCURRING IN FORMULA
1190 LET V = 0
1200 FOR I = 65 TO 84
1210 IF G(I) = 10 THEN 1240
1220 LET V = V+1
1230 LET V(V) = I+36
1240 NEXT I
1250 IF V <= 12 THEN 1280
1260 PRINT "TOO MANY VARIABLES"
1270 STOP
1280 REM  **********  CALCULATION OF TRUTH TABLE  **********
1290 REM
1300 FOR Q = 0 TO 2^V-1
1310 LET X = Q
1320 FOR I = V TO 1 STEP -1
1330 LET Y = INT(X/2)
1340 LET S(V(I)) = X - Y*2
```

```
1350 LET X = Y
1360 NEXT I
1370 FOR I = 1 TO P
1380 ON C(I) GOTO 1390,1390,1410,1430,1450,1470,1490
1390 LET S(I) = 1 - S(A(I))
1400 GOTO 1500
1410 LET S(I) = SGN( S(A(I)) + S(B(I)) )
1420 GOTO 1500
1430 LET S(I) = S(A(I)) * S(B(I))
1440 GOTO 1500
1450 LET S(I) = ( 1 - S(A(I)) ) * S(B(I))
1460 GOTO 1500
1470 LET S(I) = ABS( S(A(I)) - S(B(I)) )
1480 GOTO 1500
1490 LET S(I) = 1 - SGN( S(A(I)) + S(B(I)) )
1500 NEXT I
1510 LET T(0) = S(R)
1520 NEXT 0
1530 REM ***************** FIND A CONCLUSION *********
1540 PRINT "VARIABLE";
1550 INPUT A$(1)
1560 LET F3 = 0
1570 IF A$(1) = "BEST" THEN 1910
1580 LET F3 = 1
1590 FOR N1=2 TO 12
1600 PRINT "VARIABLE";
1610 INPUT A$(N1)
1620 IF A$(N1) = "DONE" THEN 1640
1630 NEXT N1
1640 LET N1 = N1 -1
1650 FOR I = 1 TO V
1660 LET O(I) = 0
1670 NEXT I
1680 FOR I = 1 TO N1
1690 CHANGE A$(I) TO A
1700 FOR J = 1 TO A(0)
1710 LET C = G(A(J))
1720 IF C = 11 THEN 1790
1730 IF C > 11 THEN 1820
1740 IF C = 10 THEN 1770
1750 PRINT "ILLEGAL CHARACTER IN VARIABLE LIST"
1760 GOTO 2820
1770 PRINT"THE VARIABLE ";A$(I);" DOES NOT OCCUR IN A PREMISE."
1780 GOTO 2820
1790 NEXT J
1800 PRINT "ILLEGAL VARIABLE LIST"
1810 GOTO 2820
1820 FOR J = 1 TO V
1830 IF V(J) = C+36 THEN 1870
1840 NEXT J
1850 PRINT "ERROR IN PROGRAM"
```

```
1860 STOP
1870 LET O(J) = 1
1880 NEXT I
1890 LET C = N1
1900 GOTO 2000
1910 FOR C = 1 TO V
1920 REM TRY TO FIND A CONCLUSION IN  C  VARIABLES
1930 REM MARK FIRST SET OF  C  VARIABLES
1940 FOR I = 1 TO C
1950 LET O(I) = 1
1960 NEXT I
1970 FOR I = C+1 TO V
1980 LET O(I) = 0
1990 NEXT I
2000 LET F1 = 0
2010 LET F2 = 0
2020 FOR P = 0 TO 2↑C-1
2030 LET F4 = 1
2040 REM ASSIGN TRUTH VALUES TO MARKED VARIABLES
2050 LET X = P
2060 FOR I = V TO 1 STEP -1
2070 IF O(I) = 0 THEN 2110
2080 LET Y = INT(X/2)
2090 LET S(I) = X - Y*2
2100 LET X = Y
2110 NEXT I
2120 FOR Q = 0 TO 2↑(V-C)-1
2130 REM ASSIGN TRUTH VALUES TO UNMARKED VARIABLES
2140 LET X = Q
2150 FOR I = V TO 1 STEP -1
2160 IF O(I) = 1 THEN 2200
2170 LET Y = INT(X/2)
2180 LET S(I) = X - Y*2
2190 LET X = Y
2200 NEXT I
2210 REM COMPUTE LINE IN TRUTH TABLE
2220 LET X = 0
2230 FOR I = 1 TO V
2240 LET X = X*2 + S(I)
2250 NEXT I
2260 IF T(X) = 1 THEN 2300
2270 ON F4 GO TO 2280,2320,2570
2280 F4 = 2
2290 GO TO 2320
2300 ON F4 GO TO 2310,2570,2320
2310 F4 = 3
2320 NEXT Q
2330 REM A TAUTOLOGY EXISTS FOR THIS SET
2340 IF F2 = 1 THEN 2520
2350 REM THIS FACT HAS NOT BEEN NOTED PREVIOUSLY
2360 IF F1 = 1 THEN 2440
```

```
2370 IF F3 = 1 THEN 2430
2380 PRINT
2390 PRINT "CONCLUSIONS IN";C;"VARIABLE";
2400 IF C = 1 THEN 2420
2410 PRINT "S";
2420 PRINT ":"
2430 PRINT
2440 REM PRINT TRUTH TABLE
2450 LET F1 = F2 = 1
2460 PRINT
2470 FOR I = 1 TO V
2480 IF O(I) = 0 THEN 2500
2490 PRINT V$(V(I)-100);"    ";
2500 NEXT I
2510 PRINT
2520 GOSUB 2990
2530 ON F4 GO TO 1850,2540,2560
2540 PRINT "T"
2550 GO TO 2570
2560 PRINT "F"
2570 NEXT P
2580 IF F3 = 0 THEN 2670
2590 IF F1 = 1 THEN 2820
2600 PRINT
2610 PRINT "NO CONCLUSIONS CAN BE MADE BASED ON ONLY";
2620 PRINT " THE SPECIFIED VARIABLE";
2630  IF C = 1 THEN 2650
2640 PRINT "S";
2650 PRINT
2660 GOTO 2820
2670 REM GENERATE THE NEXT SET OF  C  VARIABLES
2680 FOR I = 1 TO V
2690 IF O(I) = 1 THEN 2710
2700 NEXT I
2710 IF I >= V THEN 2750
2720 FOR J = I+1 TO V
2730 IF O(J) = 0 THEN 2900
2740 NEXT J
2750 IF F1 = 1 THEN 2810
2760 PRINT "NO CONCLUSIONS CAN BE MADE BASED ON ONLY";C;"VARIABLE";
2770 IF C = 1 THEN 2790
2780 PRINT "S";
2790 PRINT
2800 IF F1 <> 0 THEN 2820
2810 NEXT C
2820 PRINT
2830 PRINT "DO YOU WISH TO DRAW A CONCLUSION IN OTHER VARIABLES";
2840 INPUT A$
2850 IF A$ = "YES" THEN 1530
2860 IF A$ = "NO" THEN 3040
2870 IF A$ = "NEW" THEN 130
```

```
2880 PRINT "ANSWER 'YES', 'NO', OR 'NEW'";
2890 GO TO 2840
2900 LET O(J) = 1
2910 FOR K = J-1 TO I STEP -1
2920 LET O(K) = 0
2930 NEXT K
2940 IF J-I < 2 THEN 2980
2950 FOR K = 1 TO J-I-1
2960 LET O(K) = 1
2970 NEXT K
2980 GOTO 2010
2990 FOR I = 1 TO V
3000 IF O(I) = 0 THEN 3020
3010 PRINT V$(S(I)+20);"    ";
3020 NEXT I
3030 RETURN
3040 END
```

EXAMPLE:

PROBLEM

TO DETERMINE THE STRONGEST POSSIBLE CONCLUSION WHERE THE PREMISES ARE:

(A & B) = >C

(A/D) < = >C

SAMPLE RUN

```
LIST FOR INSTRUCTIONS.

PREMISE  ?(A&B)=>C
PREMISE  ?(-A/D)<=>C
PREMISE  ?DONE

VARIABLE  ?BEST
NO CONCLUSIONS CAN BE MADE BASED ON ONLY  1 VARIABLE

CONCLUSIONS IN  2 VARIABLES

A    C
T    T    T
T    F    F

C    D
T    F    T
F    F    F


DO YOU WISH TO DRAW A CONCLUSION IN OTHER VARIABLES  ?NO
```

PLAYBOY

DESCRIPTION

Shades of Hugh Hefner, centerfolds and Bunnies, this is a recreation of that famous symbol known throughout the world. Drawn first as a silhouette against a white background it is then reversed and drawn again with a dark background. Playboy will execute in 4K Bytes of memory in most Basic speaking computers.

PLAYBOY

```
001 FOR I=1 TO 10
002 PRINT
003 NEXT I
10 REM
11PRINT" ********* ************************************************ ********* "
12PRINT"********* ********************************** ***********"
13PRINT"********** ************************** ***********"
14PRINT"********** ************** ***********"
15PRINT"************** ********** ***********"
16PRINT"***************** ******* ********** "
17PRINT"********************* **** ********** "
18PRINT"***************************** ** ********** "
19PRINT"****************************** ********** "
20PRINT"********************** ********** "
21PRINT"****************** ****** ********** "
22PRINT"***************** ********** "
23PRINT"***************** ********** "
24PRINT"******************* ********** "
25PRINT"***************************** ********** "
26PRINT"******************************** * ********** "
27PRINT"********************** * ****** ********** "
28PRINT" ********************** *** ********** "
29PRINT"    ************************************************************ "
30PRINT
```

```
31PRINT"*****************************************************************************  "
32PRINT"*                                                                           *  "
33PRINT"*         ****                                        ***                   *    "
34PRINT"*         ***********                                 *********             *"
35PRINT"*           ***************                           ***********           *  "
36PRINT"*              ***************                      ***********           *    "
37PRINT"*                  ***************                  ***********           *  "
38PRINT"*                     ***************             ***********           *  "
39PRINT"*                        ***************       ***********             *    "
40PRINT"*                                  *******  *********                  *  "
41PRINT"*                           ********************                       *    "
42PRINT"*                        ******    ********************               *    "
43PRINT"*                       ********   ********************               *     "
44PRINT"*                     ***************************************          *    "
45PRINT"*                      ***************************************         *  "
46PRINT"*                          ***************************                 *    "
47PRINT"*                                ********************                  *  "
48PRINT"*                             **   **  *  **          *               *  "
49PRINT"*                             ***********  ******                     *"
50PRINT"*                              *         *                            *  "
51PRINT"*****************************************************************************  "
52 FOR I=1 TO 10
53 PRINT
54 NEXT I
55 END
```

PRIMES:

DESCRIPTION

This program factors any input number into its primes. The input number should be less then eight digits.

USERS

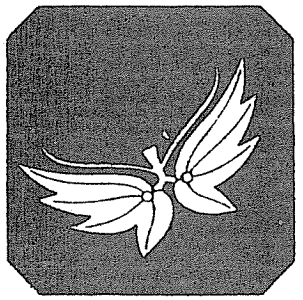
This program can be used as a teaching aid for the student as well as the teacher.

INSTRUCTIONS

Primes is self prompting and is ready to run as soon as it is loaded into memory. The program can be listed for additional instructions.

LIMITATIONS

Primes requires 2K Bytes for storage and should execute in 4K in most systems.



```
110REM
120REM
130REM  DESCRIPTION--THIS BASIC PROGRAM FINDS THE PRIME
140REM                FACTORIZATION OF A NUMBER.
150REM
160REM  INSTRUCTIONS--TYPE "RUN" AND FOLLOW INSTRUCTIONS.
170REM                 THE PROGRAM WILL STOP IF THE NUMBER
180REM                 TO BE FACTORED IS 0 .
190REM
200REM
210 PRINT "THIS PROGRAM FINDS THE PRIME FACTORIZATION OF A NUMBER."
220 PRINT "IF YOU ASK IT TO FACTOR 0, IT WILL STOP."
230 PRINT
240 PRINT
250 PRINT "WHAT NUMBER IS TO BE FACTORED";
260 INPUT A
270 IF A<134217728 THEN 310
280 PRINT "SORRY!  THIS PROGRAM IS ONLY DESIGNED TO FACTOR NUMBERS"
290 PRINT "OF 8 DIGITS OR LESS!  YOU MAY TRY AGAIN--"
300 GOTO 230
310 LET D=A
320 PRINT
330 IF A=2 THEN 630
340 LET Q=0
350 IF A>0 THEN 370
360 STOP
370 LET C=2
380 GOSUB 420
390 FOR C=3 TO SQR(A) STEP 2
400 GOSUB 420
410 GOTO 580
420 LET B=0
430 IF A=C*INT(A/C) THEN 450
440 GOTO 480
450 LET A=A/C
460 LET B=B+1
470 GOTO 430
480 IF B<1 THEN 570
490 IF Q=1 THEN 560
500 LET Q=1
510 PRINT "THE PRIME FACTORS OF";D;"ARE:"
520 PRINT
530 PRINT "          PRIME       MULTIPLICITY"
540 PRINT "          -----       ------------"
550 PRINT
560 PRINT C,B
570 RETURN
580 NEXT C
590 IF A=1 THEN 230
600 IF Q=0 THEN 630
610 PRINT A,1
620 GOTO 230
630 PRINT "THE NUMBER";A;"IS PRIME."
640 GOTO 230
650 END
```

SAMPLE RUN

```
THIS PROGRAM FINDS THE PRIME FACTORIZATION OF A NUMBER.
IF YOU ASK IT TO FACTOR 0, IT WILL STOP

WHAT NUMBER IS TO BE FACTORED  ?1872

THE PRIME FACTORS OF 1872 ARE:

     PRIME            MULTIPLICITY

       2                   4
       3                   2
      13                   1

WHAT NUMBER IS TO BE FACTORED  ?134217728
SORRY!  THIS PROGRAM IS ONLY DESIGNED TO FACTOR NUMBERS
OF 8 DIGITS OR LESS!  YOU MAY TRY AGAIN

WHAT NUMBER IS TO BE FACTORED  ?1342177

THE NUMBER 1342177 IS PRIME.

WHAT NUMBER IS TO BE FACTORED  ?0
```

PROBAL:

DESCRIPTION

This program determines probabilities using Chi-Square calculations for 2 x 2 tables.

USERS

Statisticians will find the most use for this program, however other individuals using Chi-Square calculation analysis for testing the mathematical goodnes of certain curves will also find this program useful.

INSTRUCTIONS

The data must be entered into data statements prior to program execution. Starting in line 900 enter the table data in the following format:

```
A    B      A1    B1     ..........
C    D      C1    D1

900 DATA    A,B,C,D,A1,B1,C1,D1,..........
```

After the data has been entered type RUN. List the program for additional program details.

LIMITATIONS

Line 92 contains a Restore statement and line 380 contains an ABS( ) statement. The source code is 2K Bytes long and the program will require 3K Bytes of memory for execution.

```
10  REM  *** DESCRIPTION:  THIS PROGRAM COMPUTES CHI-SQUARE VALUES
20  REM  AND PROBABILITIES FOR ANY NUMBER OF TWO BY TWO TABLES.
30  REM
40   REM  *** INSTRUCTIONS:  PUT DATA IN LINES 900 TO 997.  ENTER
50  REM  THE TABLES BY ROWS:  FIRST TABLE 1, THEN TABLE 2, ETC..
80  REM
90 READ T
92 RESTORE
94 IF T <> 1E30 THEN 100
96 PRINT "LIST LINES 10 TO 80 FOR INSTRUCTIONS"
98 STOP
100 READ A
101 IF A = 1E30 THEN 98
102 READ B,C,D
105 PRINT
110 PRINT "TABLE", "     ", "CHI SQUARE"
120 PRINT
130 LET N = A + B + C + D
140 LET E = A*D - B*C
150 LET G = N * E * E
160 LET R1 = A + B
170 LET R2 = C + D
180 LET C1 = A + C
190 LET C2 = B + D
200 LET X = G/(R1*R2*C1*C2)
210 PRINT A, B
220 PRINT C, D, X
250 LET L = 1
260 LET G = X
270 LET P = 1
280 IF G < 1 THEN 330
290 LET A = L
300 LET B = 1000
310 LET F = G
320 GO TO 360
330 LET A = 1000
340 LET B = L
350 LET F = 1/G
360 LET A1 = 2/(9*A)
370 LET B1 = 2/(9*B)
380 LET Z = ABS((1-B1)*F↑(.333333) - 1 + A1)
390 LET Z = Z / SQR(B1*F↑(.666667) + A1)
400 IF B < 4 THEN 440
410 LET P = (1 +Z*(.196854 +Z*(.115194 +Z*(.000344 +Z*.019527))))↑4
420 LET P = .5/P
430 GO TO 460
440 LET Z = Z * (1 + .08*Z↑4/B↑3)
450 GO TO 410
460 IF G >= 1 THEN 480
470 LET P = 1 - P
480 PRINT
490 PRINT "EXACT PROBABILITY IS "; INT(100000*P+.5)/100000
500 PRINT
510 GO TO 100
998 DATA 1E30
999 END
```

EXAMPLE:

PROBLEM

TO PERFORM A CHI-SQUARE ANALYSIS ON THE FOLLOWING 2 X 2 TABLES

6	10		14	8	
8	16		41	8	

8	16
12	17

SAMPLE RUN

```
  900  DATA  6,10,8,16, 14,8,41,8, 8,16,12,17
  RUN


  TABLE                                  CHI-SQUARE

            6           10
            8           16                .0732601

  EXACT PROBABILITY IS      .78316


  TABLE                                  CHI-SQUARE

           14            8
           41            8                3.491921

  EXACT PROBABILITY IS      .05856


  TABLE                                  CHI-SQUARE

            8           16
           12           17                .3618251

  EXACT PROBABILITY IS      .55491
```

QUADRAC:

DESCRIPTION

Quadrac solves quadratic equations.

USERS

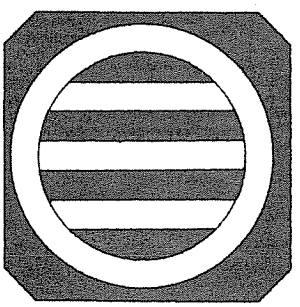
This program would be useful to persons requiring solutions for quadratic equations, this would include teachers, students, engineers, mathematicians, etc.

INSTRUCTIONS

The program is ready to run after it is loaded into memory. The program is self prompting and will request all required data. The program may be listed for additional instructions.

LIMITATIONS

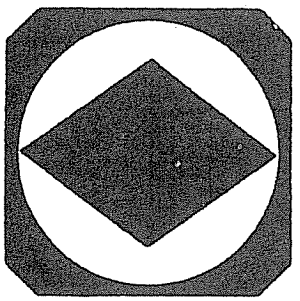
The ABS( ) statement is used in lines 130, 400 and 480. Quadrac stores in 2K Bytes and executes in 3K in most Basic systems.



QUADRAC

```
10 REM
20 DIM A(3)
30 PRINT
40 PRINT
50 PRINT "I SOLVE THE QUADRATIC EQ.  A*X*X+B*X+C=0"
60 PRINT
70 PRINT "INPUT A,B,C";
80 INPUT A(1),A(2),A(3)
90 PRINT
100 LET Y1=1E36
110 LET Y2=1E-36
120 FOR I=1 TO 3
130 LET X=ABS(A(I))
140 IF X=0 THEN 160
150 GOSUB 560
160 NEXT I
170 IF Y1<=1 THEN 200
180 LET D=Y1
190 GO TO 220
200 IF Y2>=1 THEN 250
210 LET D=Y2
220 FOR I=1 TO 3
230 LET A(I)=A(I)/D
240 NEXT I
250 IF A(1)<>0 THEN 360
260 IF A(2)=0 THEN 310
270 LET N=1
280 LET R1=-A(3)/A(2)
290 PRINT "ONLY ROOT IS ";R1
300 GO TO 500
310 IF A(3)=0 THEN 340
320 PRINT "NO ROOTS"
330 GO TO 500
340 PRINT "ALL COMPLEX #'S ARE ROOTS"
350 GO TO 500
360 LET D=A(2)*A(2)-4.*A(1)*A(3)
370 LET A2=A(1)+A(1)
380 LET R=-A(2)/A2
390 IF D<0 THEN 480
400 LET E=ABS(SQR(D)/A2)
410 IF R<0 THEN 440
420 LET R1=R+E
430 GO TO 450
440 LET R1=R-E
450 LET R2=A(3)/(A(1)*R1)
460 PRINT "REAL ROOTS:";R1;" AND ";R2
470 GO TO 500
480 LET E=ABS(SQR(-D)/A2)
```

```
490 PRINT "COMPLEX ROOTS:";R;"(+ AND -)";E;" I"
500 PRINT
510 PRINT
520 PRINT "MORE EQ'S TO SOLVE (1=YES, 0=NO)";
530 INPUT I
540 IF I=1 THEN 60
550 GO TO 610
560 IF X>=Y1 THEN580
570 LET Y1=X
580 IF X<=Y2 THEN 600
590 LET Y2=X
600 RETURN
610 END
```



RED BARON

DESCRIPTION

Here comes Snoopy's arch enemy! "Ten, Twenty, Thirty, Forty, Fifty or more, the bloody Red Baron is rolling up the score". Execute the program and see him emerge right before your very eyes. The program is 5K Bytes long and will execute in 6K Bytes of memory.

```
RED BARON


001 FOR I=1 TO 10
002 PRINT
003 NEXT I
10 REM
11PRINT"                                                    ×XXXXXXXXXXXXX"
12PRINT"                                                 ×XXXXXXXXXXXXXXXXXX "
13PRINT"                                              ×XX XXXXX            ×XXXXXXXX"
14PRINT"                                            □OOO  XXXXX   OOOOOOO   ×XXXXXXXXX"
15PRINT"                                           □OOOO   ×X   □OOOOOOOOO  ×XXXXXXXXX"
16PRINT"                                          □OOOO   ×X   □OOOOOOOOOO  XXXXXXXXXX"
17PRINT"                                         □OOOO   ×X   □OOOOOOOOOO   ×XXXXXXXXXX"
18PRINT"                                        □OOOO    ×X  OOOOOOOOOO   × XXXXXXXXXX"
19PRINT"                                       □OOO      ×X    □OOOOOOO    ×  ×XXXXXXXXX"
20PRINT"                                       ×      ×   ×    □OOO     ×X   X XXXXXXXX"
21PRINT"                                       ×XXXX X XXX XXXX  ×XXXXXXXXX   ×XXXXXXX"
22PRINT"              ×XXXXXXXXXX XXXX  XXX XX  ×X   ×XX  ×XXXXXXX    ×XXXXX"
23PRINT"          ×XX              ×X     XXX    ×XXX       XXXXXXXX      XXXX"
24PRINT"        ×XX                     ×XXXX   ×XXX         ×XXXXX   ×X   ×XX"
25PRINT"      ×XX                       ×XXXX  XXXXX          ×XXXX  XXXXX  ×XX"
26PRINT"    ×XX                         ×XXXX   ×XXX          ×XXXX  XXXXX  ×X"
27PRINT"  ×XX                           ×XX     ×X             ×XX  XXXXXX  XX"
28PRINT" XXX                                                   XXX  XXXXXX  XX"
29PRINT" ×XX             ×XXXXX                                ×XX  XXXXXX  XX"
30PRINT"XXX            ×XXXXXXXX                               ×XX  XX  XXX"
```

```
31PRINT"×××          ×××××××××                              ××  ×××××  ××××"
32PRINT"××          ×××××××××                              ×××   ×××  ×××××"
33PRINT"×××           ×××××                                 ××××       ×××××"
34PRINT"×××                                                 ××              ××××××××××××××"
35PRINT" ×××                                                ××              ××××××××××××××"
36PRINT"  ×××                                               ××              ××××××××××××××"
37PRINT"  ×××                                               ××              ××××  ××××××××××"
38PRINT"   ×××       ××                        ×××              ××××××××××××××"
39PRINT"    ×××       ××                      ××               ××  ××××××××× "
40PRINT"     ×××        ××××××××××××××××××                    ×××  ××   ××××"
41PRINT"        ××××××××××××××××××××××××                     ××  ××   ×××× "
42PRINT"                              ×××                     ××××   ×××× "
43PRINT"                              ×××                   ×××××××   "
44PRINT"                              ×××                ××××× "
45PRINT"                               ×××      ××××××××××"
46PRINT"                                ××××××××××××    "
47PRINT"                               ××××××××××   "
48PRINT"                              ××        ××   "
49PRINT"                             ××××××××××××××                  ×"
50PRINT"                            ××××××××××××××                 ×××  ×  "
51PRINT"                          ×××                ××       ××××     ××"
52PRINT"                         ×××                 ×  ×  ×××   × ×××  "
53PRINT"                        ××                   ××    × ×××     ×××"
54PRINT"                       ××      ×××××         ××    ××    ×××× "
55PRINT"                     ××     ×××    ×××     ××    ×× ××× × "
56PRINT"                    ××     ×          ××  ××    ×××     ×  "
57PRINT"                   ××     ×            ×××    ××      × "
58PRINT"                  ××       ×                  × ×       × "
59PRINT"                 ××        ×          ×××     × ×        × "
60PRINT"               ××       ××××××××××   ××      ×  ×          ×   "
61PRINT"               ××    ××     ×××        ××     ×  ××           × "
62PRINT"               ×××   ××   ×××              ××   ×  ×××           ×"
63PRINT"               ××   ××                    ×××   ××0×             ×  "
64PRINT"               ××                           ××00×              ××"
65PRINT"               ××                           ××0000×      ××××0  0"
66PRINT"               ××                          ××000000×××××  0  □  □  "
67PRINT"                ××                         ××0×××0     □ □   □  □   □"
68PRINT"                ××                        ××× 0   □   □  □   □  □"
69PRINT"                 ××                       ××  □   □   0  □   □  "
70PRINT"                  ×××                    ××    □   □   □  □   "
71PRINT"                   ××××                ×××     □   □   □"
72PRINT"                      ××××××××       ×     ××    "
73PRINT"                          ××         ×     ××   "
74PRINT"                          ××         ×     ××  "
```

```
75PRINT"                           xX        x       xX                     "
76PRINT"                           XX        x       xX    xXXXXXX    "
77PRINT"             xXXXXXXXXXXX XX         x       xXXXX       xXXXX  "
78PRINT"       xXXXXX           xXXXX         x                xX     xXX "
79PRINT"    xXXX                              x            xX    xX     xX "
80PRINT"   xXX        xX                      x             xX    xX    xX  "
81PRINT"  xXX        xX      xX          xXXXXXXX         xXX   xXX  xX"
82PRINT"   xX     XXXX     xXX      xXXXXX       xXXXXXXXXXXXXXXXXXXXXXX    "
83PRINT"     xXXXXXXXXXXXXXXXXXXXXXXXX   "
84 FOR I=1 TO 8
85 PRINT
86 NEXT I
87 END
```

REGRESSION 2:

DESCRIPTION

This program performs multiple linear regressions on data groups.

USERS

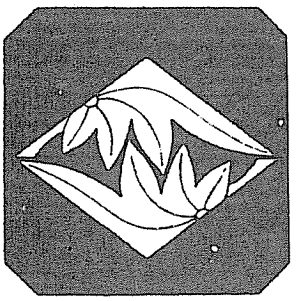
Engineers, poll takers and people studying data samples will find the most use for this program. This program can also be of use to students and statisticians.

INSTRUCTIONS

Your data must be entered before the program is run. List the program for detailed instructions on loading data. After the data is entered type RUN.

LIMITATIONS

This program uses two dimensional arrays starting in line 510. The MAT READ statement is used in lines 540 and 600. The TAB( ) statement is used throughout the program starting in line 780. Six K Bytes of memory are required for program storage. With the DIM statements set as they presently are the program will require 32K Bytes of memory for execution. This number may be reduced by reducing the table sizes in the DIM statements.



```
110 REM REGRESSION 2
110REM
120REM DESCRIPTION--COMPUTES ONE OR MORE MULTIPLE LINEAR
130REM               REGRESSIONS ON A BATCH OF DATA.
140REM
150REM INSTRUCTIONS--
160REM
170REM PLACE DATA BEGINNING IN LINE 2000 IN FOLLOWING ORDER:
180REM
190REM N (NUMBER OF DATA SETS OR OBSERVATIONS)
200REM V (NUMBER OF VARIABLES IN DATA BATCH)
210REM G (NUMBER OF REGRESSIONS TO BE PERFORMED)
220REM DATA VALUES BY DATA SET.  THAT IS:
230REM    ENTER FIRST THE VALUES FOR ALL VARIABLES AT OBSERVATION 1,
240REM    THEN THE VALUES FOR THE VARIABLES FOR OBSERVATION 2, ECT.
250REM    VARIABLES MAY BE ENTERED IN ANY ORDER, BUT THE ORDER
260REM    MUST BE THE SAME FOR EACH DATA SET.  THE POSITION OF THE
270REM    VARIABLE IN ENTERING THE DATA IS THE INDEX OF THE VARIABLE.
280REM    THUS IF THERE ARE 4 VARIABLES ENTERED IN THE ORDER
290REM    X,  Y,  Z,  W, THE INDEX OF Z IS 3.
300REM NEXT, FOR EACH REGRESSION, ENTER, IN ORDER:
310REM    H (THE NUMBER OF THE REGRESSION), THEN
320REM    K (TOTAL NUMBER OF INDEPENDENT VARIABLES IN THIS REGRESSION),
330REM    P1 (ENTER 1 IF YOU WANT THE VARIANCE-COVARIANCE MATRIX
340REM        PRINTED OUT, OTHERWISE ENTER 0),
350REM    P2 (ENTER 1 IF YOU WANT THE RESIDUALS PRINTED OUT,
360REM        OTHERWISE ENTER 0), AND THEN ENTER THE
370REM    INDICES OF THE INDEPENDENT VARIABLES FOLLOWED BY THE
380REM    INDEX OF THE DEPENDENT VARIABLE.
390REM
400REM IF N > 211  OR V > 17, THEN THE DIM STATEMENTS IN LINES
410REM 510 AND 520 MUST BE CHANGED.
420REM
430REM SAMPLE DATA ARE IN LINES 2000 THROUGH 2230.  BE SURE TO
440REM REMOVE THE SAMPLE DATA BEFORE RUNNING THE PROGRAM WITH
450REM YOUR DATA.
460REM
470REM * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
480REM
490 PRINT "LIST THIS PROGRAM FOR INSTRUCTIONS."
500 PRINT
510  DIM X(17,17),A(17,17),D(211,17),Y(17),M(17),S(17)
520  DIM T(17),B(17),U(17,17),R(17,17),C(17,17),Q(211),E(17)
530  READ N, V, G
540 MAT READ D(N,V)
550 FOR I=1 TO N
560  LET D(I,0)=1
570 NEXT I
```

```
580  READ H, K, P1, P2
590 LET M = K + 1
600   MAT READ E(M)
610 PRINT "**REGRESSION NUMBER";H;": DEPENDENT VARIABLE IS";E(M)
620 PRINT
630 IF H>1 THEN 780
640 FOR I=0 TO U
650  FOR J=0 TO U
660     LET X=0
670   FOR L=1 TO N
680       LET X=X+D(L,I)*D(L,J)
690 NEXT L
700    LET X(I,J)=X
710      LET C(I,J)=X
720   NEXT J
730  LET T(I)=X(0,I)/X(0,0)
740 LET B(I)=0
750 IF I=0 THEN 770
760 LET B(I)=SQR(X(I,I)/(N-1)-X(0,I)*X(0,I)/(N*(N-1)))
770 NEXT I
780 PRINT TAB(7);"INDEX";TAB(22);"MEANS";TAB(33);"STANDARD DEV."
790 FOR I=1 TO M
800 LET M(I)=T(E(I))
810 LET S(I)=B(E(I))
820 PRINT E(I), M(I), S(I)
830 NEXT I
840 PRINT
850 PRINT
860 PRINT "CORRELATION COEFFICIENTS"
870  IF H>1 THEN 930
880 FOR I=1 TO U
890  FOR J=1 TO U
900   LET R(I,J)=(N*X(I,J)-X(0,I)*X(0,J))/(N*(N-1)*B(I)*B(J))
910  NEXT J
920 NEXT I
930 FOR I=1 TO M
940   FOR J=1 TO M
950     LET U(I,J)=R(E(I),E(J))
960   PRINT U(I,J),
970   NEXT J
980 PRINT
990 PRINT
1000 NEXT I
1010 PRINT
1020 LET E(0)=0
1030 FOR I=0 TO K
1040  LET Y(I)=C(E(I),E(M))
1050   FOR J=0 TO K
1060    LET X(I,J)=C(E(I),E(J))
1070   NEXT J
1080 NEXT I
```

```
1090 FOR I=0 TO K
1100  FOR J=0 TO K
1110   IF I<>J THEN 1140
1120   LET A(I,J)=1
1130   GO TO 1150
1140   LET A(I,J)=0
1150  NEXT J
1160 NEXT I
1170 FOR I=0 TO K
1180 IF X(I,I)<1E-6 THEN 1930
1190  LET Y(I)=Y(I)/X(I,I)
1200   FOR J=0 TO K
1210    LET A(I,J)=A(I,J)/X(I,I)
1220    IF J=I THEN 1240
1230    LET X(I,J)=X(I,J)/X(I,I)
1240   NEXT J
1250  LET X(I,I)=1
1260   FOR L=0 TO K
1270    IF L=I THEN 1350
1280    LET Y(L)=Y(L)-X(L,I)*Y(I)
1290     FOR J=0 TO K
1300      LET A(L,J)=A(L,J)-X(L,I)*A(I,J)
1310      IF J=I THEN 1330
1320      LET X(L,J)=X(L,J)-X(L,I)*X(I,J)
1330     NEXT J
1340    LET X(L,I)=0
1350   NEXT L
1360 NEXT I
1370 LET S6=C(E(M),E(M))
1380 FOR I=0 TO K
1390   LET S6=S6-Y(I)*C(E(I),E(M))
1400 NEXT I
1410 LET S7=S6/(N-M)
1420 LET R2=1-S7/(S(M)*S(M))
1430 LET R=SQR(R2)
1440 LET S8=SQR(S7)
1450 IF P1=0 THEN 1470
1460 PRINT "VARIANCE-COVARIANCE MATRIX"
1470 FOR I=0 TO K
1480   FOR J=0 TO K
1490  LET A(I,J)=A(I,J)*S7
1500  IF P1=0 THEN 1520
1510 PRINT A(I,J),
1520  NEXT J
1530 IF P1=0 THEN 1560
1540  PRINT
1550  PRINT
1560 NEXT I
1570 PRINT
1580PRINT TAB(7);"INDEX";TAB(25);"B";TAB(32);"STD. ERROR";
1590 PRINT TAB(50);"T-RATIO"
```

```
1600 FOR I=0 TO K
1610 PRINT E(I), Y(I), SQR(A(I,I)), Y(I)/SQR(A(I,I))
1620 NEXT I
1630 PRINT
1640 PRINT "R-Squared=" ;R2, "R=" ;R
1650 PRINT
1660 PRINT "Stand. Error of Est.=";S8;"D.F.=";(N-M)
1670 PRINT
1680 FOR I=1 TO N
1690  LET Z=D(I,E(M))-Y(0)
1700   FOR J=1 TO K
1710   LET Z=Z-Y(J)*D(I,E(J))
1720   NEXT J
1730 LET Q(I)=Z
1740 NEXT I
1750 LET W=0
1760 FOR I=2 TO N
1770 LET W=W+(Q(I)-Q(I-1))*(Q(I)-Q(I-1))
1780 NEXT I
1790 PRINT
1800 IF P2=0 THEN 1870
1810PRINT TAB(6);"ACTUAL";TAB(18);"PREDICTED";TAB(34);"RESIDUAL"
1820 LET I = 0
1830 LET I = I + 1
1840  PRINT D(I,E(M)), D(I,E(M))-Q(I), Q(I)
1850 IF I = N THEN 1870
1860 GO TO 1830
1870 PRINT
1880 PRINT "Durbin-Watson Stat.=" ;W/S6
1890 IF H<G THEN 1910
1900 GO TO 1940
1910 PRINT
1920  GO TO 580
1930 PRINT "Correlation Matrix becoming singular"
1940 PRINT
1950 PRINT "     *****Problem Completed*****"
1960 STOP
2000 DATA 15, 4, 4
2010 DATA 32, 48, 54, 15
2020 DATA 36, 33, 19, 16
2030 DATA  3, 28, 30, 14
2040 DATA 12, 33, 64, 22
2050 DATA 36, 34, 60, 24
2060 DATA 24, 36, 53, 19
2070 DATA 19, 42, 29, 13
2080 DATA 20, 33, 55, 15
2090 DATA 27, 36, 62, 23
2100 DATA 15, 22, 33, 12
2110 DATA 45, 46, 68, 25
2120 DATA  9, 28, 42, 17
2130 DATA 11, 32, 45, 18
```

```
2140 DATA 33, 34, 39, 19
2150 DATA 21, 45, 39, 18
2160 DATA 1,1,1,1
2170 DATA 3,4
2180 DATA 2,2,1,0
2190 DATA 1,3,4
2200 DATA 3,3,0,1
2210 DATA 1,2,3,4
2220 DATA 4,1,0,0
2230 DATA 4,3
9999 END
```

ROAD RUNNER

DESCRIPTION

Has anyone seen the coyote? Well check around before you run this program, because it generates a likeness of the Road Runner, that speedy bird that spends Saturday mornings "foiling" the attempts of the coyote to catch him. It only takes 8K Bytes of memory to ginn up a copy of this bird, if your computer can catch him.

```
ROAD RUNNER



001 FOR I=1 TO 10
002 PRINT
003 NEXT I
10 REM
11PRINT"                                        RRR  "
12PRINT"                                   RRRR       RRRRRRRR   "
13PRINT"                                  RRRRRR   RRRRRRRRRRRR"
14PRINT"                                 RRRRRRRRR RRRRRRRRRRRRRR  "
15PRINT"                                RRRRRRRRRRRRRRRRRRRRRRRRR  "
16PRINT"                                RRRRRRRRRRRRRRRRRRRRRRRRR  "
17PRINT"                                RRRRRRRRRRRRRRRRRRRRRRRRRRR "
18PRINT"                               RRRRRRRRRRRRRRRRRRRRRRRRRRR "
19PRINT"                               RRRRRRRRRRRRRRRRRRRRRRRRRRR "
20PRINT"                              RRRRRRRRRRRRRRRRRRRRRRRRRRRR "
21PRINT"                              RRRRRRRRRRRRRRRRRRRRRRRRRRRR"
22PRINT"                             RRRRRRRRRRRRRRRRRRRRRRRRRRRR "
23PRINT"                             RRRRRRRRRRRRRRRRRRRRRRRRRRRR"
24PRINT"                             RRRRRRRRRRRRRRRRRRRRRRRRRRRR"
25PRINT"                             RRRRRRRRRRRRRRRRRRRRRRRRRRR  "
26PRINT"                             RRRRRR   RRRRRRRRRRRRRRRRR  "
27PRINT"                              RRRR      RRRRRRRRRRRRRRR  "
28PRINT"                               RR        RRRRRRRRRRRRR"
29PRINT"                                R         RRRRRRRRRRRR"
30PRINT"                                           RRRRRRRRRR  "
31PRINT"                                            RRRRRRRRR  "
32PRINT"                                             RRRRRRRR "
33PRINT"                                              RRRRRRR  "
34PRINT"                                              RRRRRRR   "
35PRINT"                                             RRRRRRRR"
36PRINT"                                             RRRRRRR "
```

```
37PRINT"                                                    RRRRRRRR "
38PRINT"                                                    RRRRRRR   "
39PRINT"                                                    RRRRRR"
40PRINT"                                                    RRRRRR  "
41PRINT"                                                     RRRR    "
42PRINT"                                                  XXXXXXXXX "
43PRINT"                                                 XXXXXXXXXXX "
44PRINT"                                                XXXXXXXXXXXXX  "
45PRINT"                                               XXXXXXXXXXXXXXX  "
46PRINT"                                               XXXXXXXXXXXXXXX   "
47PRINT"                                              XXXXXXXXXXXXXXXXX    "
48PRINT"                                              XXXX  XXXXXXX  XXX  "
49PRINT"                                              XXX    XXXXX    XX    "
50PRINT"                                        XXX   XXX    XXXXX    XX   XXX"
51PRINT"                                       XXXXX XXX    XXXXX    XX XXXXXX"
52PRINT"                                      XXXXXXXXXX OO XXXXX OO XXXXXXXXXX"
53PRINT"                                     XXXXXXXXXXXOOXXXXXXXOOXXXXXXXXXXX"
54PRINT"                                     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX "
55PRINT"                                      XXXXXA   XXXXXXXXXXXX   AXXXXX "
56PRINT"                                       XXXXA UXXXXXXXXXXXXU AXXXX "
57PRINT"                                         XXXXA UXXXXXXXXXU AXXXX "
58PRINT"                                          XXXXA UXXXXXXXU AXXXX  "
59PRINT"                                           XXXXA UXXXXXU AXXXX"
60PRINT"                                            XXXXA UXXXU AXXXX "
61PRINT"                                             XXXXA    AXXXX"
62PRINT"                                              XXXXA   AXXXX "
63PRINT"                                               XXXXXXXXXXX"
64PRINT"                        BBBBB                    XXXXXXXXX "
65PRINT"                   BB   BBBBBBBBB               XXXXXX"
66PRINT"                  BB  BBBBBBBBBBBB               XXXXX   "
67PRINT"                 BB  BBBBBBBBBBBBBB              XXXXX "
68PRINT"                BBB BBBBBBBBBB   BBBB            XXXXX"
69PRINT"                BBBBBBBBBBBBBBB     B            XXXXX  "
70PRINT"    B   BBBBBBBBBB   BBBB                        XXXXX"
71PRINT"   BB  BBBBBBBBBB      BBB                       XXXXX  "
72PRINT"   BBB BBBBBBBBB  BB     B                       XXXXX  "
73PRINT"   BBBBBBBBBBBBBBBBBBB                           XXXXX  "
74PRINT"    BBBBBBBBBBBBBBBBBB                          XXXXX   "
75PRINT" B BBBBBBBBBBBBBBB   B                          XXXXX    "
76PRINT" BB BBBBBBBBBBBB                                XXXXX   "
77PRINT"  BBBBBBBBBBBBBB                                XXXXX    "
78PRINT"   BBBBBBBBBBBBB  BBBBB                         XXXXX"
79PRINT"    BBBBBBBBBBBB BBBBBB                         XXXXX  "
80PRINT"     BBBBBBBBBBBBBBBBBBB                        XXXXX  "
81PRINT"      BBBBBBBBBBBBBB   BB                       XXXXX  "
82PRINT"        BBBBBBBBBBBB     B                      XXXXX   "
83PRINT"          BBBBBBBBB                             XXXXX "
84PRINT"           BBBBBBBBB                            XXXXX  "
85PRINT"            BBBBBBBB                            XXXXX  "
86PRINT"         B   BBBBBBB                            XXXXX  "
87PRINT"          BBBBBBBBBB                            XXXXX  "
```

```
88PRINT"              BBBBBBBBBBB                    XXXXX   "
89PRINT"                BBBBBBBBB                    XXXXX "
90PRINT"                   BBBBBBB                   XXXXX    "
91PRINT"                      BBBBBB                 XXXXX  "
92PRINT"                          BBBBB              XXXXX"
93PRINT"                             BBBBB           XXXXX  "
94PRINT"                                BBBBB        XXXXX"
95PRINT"                                    BBBBB    XXXXX  "
96PRINT"                                  XXXXXXXXXXXXXXXXXXXXXXXX "
97PRINT"                                 XXXXXXXXXXXXXXXXXXXXXXXXX"
98PRINT"          YY                    XXXXXXXXXXXXXXXXXXXXXXXXXXXX  "
99PRINT"     YY  YY                    XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
100PRINT"    YYY YYY                    XXXXXXXXXXXXXXXXXXXXXXXXXXXX  "
101PRINT"   YYY  YYY YYY                XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
102PRINT"   YYY  YYYYYYYY               XXXXXXXXXXXXXXXXXXXXXXXXXXXX  "
103PRINT"   YYY  YYYYY                   XXXXXXXXXXXXXXXXXXXXXXXXXX          YYY"

104PRINT"    YYYY YYYYY                    XXXXXXXXXXXXXXXXXXXXXXXX          YYY"

105PRINT"     YYYYYYYYYYYYYYYYYYYYYYYYYYYXXXXXXXXXXXXXXXXXXXXXXX          YYYY"

106PRINT"      YYYYYYYYYYYYYYYYYYYYYYYYYYYY                YYY             YYYY"
107PRINT"                                                 YYY          YY YY Y"

108PRINT"                                                  YYY         YY  Y  Y"

109PRINT"                                                   YYY       YY       Y"

110PRINT"                                                    YYY          YY "
111PRINT"                                                     YYY        YY"
112PRINT"                                                      YYY      YY"
113PRINT"                                                       YYY   YY"
114PRINT"                                                        YYY YY  "
115PRINT"                                                         YYYY  "
116PRINT"                                                         YYYY"
117PRINT"                                                          YY  "
118PRINT"XXX                          XXX          "
119PRINT"X  X                         X  X     "
120PRINT"X  X   XXXX   XXXX   XXX     X  X   XXXX   XXXX   XXX  "
121PRINT"XXX    X      X      X  X    XXX    X      X      X  X "
122PRINT"X  X   XXXX   XXXX   X  X    X  X   XXXX   XXXX   X  X"
123PRINT"X  X   X      X      XXX     X  X   X      X      XXX   "
124PRINT"XXX    XXXX   XXXX   X       XXX    XXXX   XXXX   X "
125 FOR I=1 TO 8
126 PRINT
127 NEXT I
128 END
```

ROULETTE:

DESCRIPTION

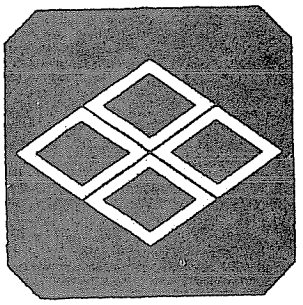
Welcome to the game of Roulette, as played in the casinos in Las Vegas. This version of the game allows you the option of placing your bet in a variety of ways or combinations. The program will also draw the game board for you to place your bets on.

INSTRUCTIONS

The program is self prompting and requires no set up prior to execution. For detailed instructions you may list the program, however upon execution Roulette will ask if you would like instructions. By responding yes to the computer the instructions will be listed.

LIMITATIONS

Roulette should execute in most 4K Basic systems without any problems providing there is sufficient memory for program execution. The source code requires 4K Bytes for storage and 6K to execute.



```
100 REM THIS IS THE GAME OF ROULETTE
115 PRINT "WELCOME TO THE GAME OF ROULETTE"
130 PRINT
145 PRINT "WOULD YOU LIKE INSTRUCTIONS - YES OR NO";
160 INPUT A$
175 IF A$="YES" THEN 265
190 IF A$="Y" THEN 265
205 IF A$="N" THEN 460
220 IF A$="NO" THEN 460
235 PRINT "ANSWER - YES OR NO - PLEASE!"
250 GOTO 145
265 PRINT
280 PRINT "  IN THIS GAME YOU ARE ALLOWED TO BET ON INDIVIDUAL"
295 PRINT "NUMBERS, NUMBER SETS, OR ODD OR EVEN NUMBERS."
310 PRINT "THE NUMBERS GO FROM 00 TO 36.YOU BET BY ENTERING A"
325 PRINT "NUMBER BETWEEN 0 AND 44. THE NUMBERS 0 TO 36"
340 PRINT "REPRESENT THEMSELVES, AND 37 REPRESENTS 00."
355 PRINT "40 REPRESENTS  EVEN. 41 REPRESENTS  ODD BETS."
370 PRINT "38  BETS THE NUMBER SET FROM  1 TO 18"
385 PRINT "39  BETS THE NUMBER SET FROM 19 TO 36"
400 PRINT "42  BETS THE NUMBER SET FROM  1 TO 12"
415 PRINT "43  BETS THE NUMBER SET FROM 13 TO 24"
430 PRINT "44  BETS THE NUMBER SET FROM 25 TO 36."
445 PRINT "00 IS NOT INCLUDED WHEN YOU BET  ODD."
460 PRINT
475 PRINT
490 PRINT "IF YOU WOULD LIKE A PRINT OUT OF THE TABLE"
505 PRINT "TYPE THE WORD  PRINT";
520 INPUT A$
535 IF A$="PRINT" GOTO 580
550 IF A$="P" GOTO 580
565 GOTO 880
580 PRINT
595 PRINT
610 B$(1)="  ************************************"
625 B$(2)="  * "
640 PRINT B$(1)
655 PRINT "  *      0          *         00    *"
670 B$(3)="  * "
685 PRINT B$(1)
700 U=-2
715 FOR I=1 TO 3
730 U=U+3
745 PRINT B$(2);U;B$(2);U+1;B$(2);U+2;B$(2)
760 PRINT B$(1)
775 NEXT I
```

```
790 FOR I=1 TO 9
805 U=U+3
820 PRINT B$(3);U;B$(3);U+1;B$(3);U+2;B$(3)
835 PRINT B$(1)
850 NEXT I
865 PRINT
880 PRINT
895 PRINT "HOW MUCH MONEY DO YOU HAVE TO SPEND";
910 INPUT A5
925 G=G+1
940 IF G>5 GOTO 985
955 PRINT "PICK YOUR BET NUMBER. (0 TO 44)";
970 GOTO 1000
985 PRINT "WHATS YOUR NUMBER";
1000 INPUT B
1015 IF B>44 GOTO 925
1030 IF B<0 GOTO 925
1045 IF G>5 GOTO 1090
1060 PRINT "HOW MUCH MONEY DO YOU WANT TO BET THIS TIME";
1075 GOTO 1105
1090 PRINT " AMOUNT !";
1105 INPUT C
1120 IF A5=C GOTO 1195
1135 IF A5>C GOTO 1195
1150 PRINT "YOU DON'T HAVE THAT MUCH MONEY!"
1165 PRINT "THE MOST YOU CAN BET IS "A5" DOLLARS."
1180 GOTO 1060
1195 N=INT(RND(-1)*100)
1210 IF N>37 GOTO 1195
1225 IF B>37 GOTO 1390
1240 IF B=N GOTO 1330
1255 A5=A5-C
1270 IF A5>0 GOTO 1300
1285 GOTO 1600
1300 PRINT N" YOU LOST BUT YOU STILL HAVE "A5" DOLLARS LEFT."
1315 GOTO 925
1330 D=35*C
1345 A5=A5+D
1360 PRINT N" YOU  ** WON **  !!  NOW YOU HAVE "A5" DOLLARS."
1375 GOTO 925
1390 IF B>39 GOTO 1720
1405 IF B=38 GOTO 1555
1420 IF N>18 GOTO 1450
1435 GOTO 1465
1450 IF N<37 GOTO 1525
1465 A5=A5-C
1480 IF A5>0 GOTO 1510
1495 GOTO 1600
1510 GOTO 1300
1525 A5=A5+C
1540 GOTO 1360
```

```
1555 IF N>18 GOTO 1465
1570 IF N>36 GOTO 1465
1585 GOTO 1525
1600 PRINT N" YOU LOST AND ARE OUT OF MONEY..."
1615 PRINT
1630 PRINT
1645 PRINT "WOULD YOU LIKE TO PLAY IT AGAIN. (YES OR NO)";
1660 INPUT Z$
1675 IF Z$="YES" GOTO 460
1690 IF Z$="Y" GOTO 460
1705 GOTO 2080
1720 IF B>41 GOTO 1855
1735 Y=(N/2)-INT(N/2)
1750 IF B=41 GOTO 1795
1765 IF Y>.01 GOTO 1465
1780 GOTO 1525
1795 IF N=37 GOTO 1465
1810 IF N=0 GOTO 1465
1825 IF Y>.01 GOTO 1525
1840 GOTO 1465
1855 IF B=42 GOTO 2005
1870 IF B=43 GOTO 1945
1885 IF N>24 GOTO 1915
1900 GOTO 1465
1915 IF N=37 GOTO 1465
1930 GOTO 2050
1945 IF N>12 GOTO 1975
1960 GOTO 1465
1975 IF N>24 GOTO 1465
1990 GOTO 2050
2005 IF N>0 GOTO 2035
2020 GOTO 1465
2035 IF N>12 GOTO 1465
2050 A5=C*2+A5
2065 GOTO 1360
2080 END
```

SANTA

DESCRIPTION

Thats right! It's Santa Claus, and he is still on the go. Doens't he ever slow down? Santa will store in 4K Bytes of memory and will run in 5K Bytes of available memory, thats in addition to the space taken up by your compiler.

```
SANTA


001 PRINT
002PRINT
003PRINT
004PRINT
005PRINT
006PRINT
007PRINT
008PRINT
009PRINT
010PRINT
011PRINT
012PRINT
013PRINT"                    XXXXXXXXXXXXXXXXX     "
014PRINT"                  XXXXXX                .  "
015PRINT"                 XXXXXXXXX                 "
016PRINT"                XXXXXXXXX                  "
017PRINT"               XXXXXXXXXX..........        "
018PRINT"               XXXXXXXXXX  000000000.....  "
019PRINT"                XXXXXXXXX  000000000000./  "
020PRINT"               $$$XXXXXX  0000.0000000  ////"
021PRINT"              $$$$$$      000000000000.0 /////"
022PRINT"              $$$$$$      000000000000000 /////  ......"
023PRINT"              $$$$          000000000000   /*****  ..  "
024PRINT"               XXX               *****0    *********  "
025PRINT"             XXXXXX                       **********   "
026PRINT"            XXXXXXX          **           **********   "
027PRINT"           XXXXXXXXX      *******        X *********   "
028PRINT"         XXXXXXXXXXX  *******             XXXX//////////XXX"
029PRINT"       XXXXXXXX       *******             XXXX//////////XXX  "
030PRINT"     XXXXXXXX         ***********         XXX///////  ////XXX"
```

```
031PRINT"
032PRINT"
033PRINT"
034PRINT"
035PRINT"
036PRINT"
037PRINT"
038PRINT"
039PRINT"
040PRINT"
041PRINT"
042PRINT"
043 PRINT
044PRINT"
045PRINT"
046PRINT"
047PRINT"
048PRINT"
049PRINT"
050PRINT"
051PRINT"
052PRINT"
060PRINT"
061PRINT"
062PRINT"
063PRINT"
064PRINT"
065PRINT"
066PRINT"
067PRINT"
068PRINT"
069PRINT"
070PRINT"
071PRINT"
072PRINT"
073PRINT"
074PRINT"
075PRINT"
076PRINT"
077PRINT"
078PRINT
079 PRINT
080 PRINT
081 PRINT
082 PRINT
083 PRINT
084 PRINT
085 END
```

STAT 10:

DESCRIPTION

This statistical program calculates, for 2 groups of paired data; having equal variances, the mean, standard error, difference, difference error and T ratios.

USERS

This program will be most useful to statisticians and engineers having a need to compare data groups.

INSTRUCTIONS

Your data must be entered in data statements, starting in line 900, before the program is run. Use the following format for entering your data:

```
900  DATA  G1,G2,G3.....
950  DATA  H1,H2,H3.....
```

where

G1 - is the first data point of group one
G2 - is the second data point of group one
G3 - is the third data point of group one

and so forth until all the points are entered. Then:

H1 - is the first data point for group two
H2 - is the second data point - and so forth

After the data is entered type RUN. List the program for additional information.

LIMITATIONS

Line 92 contains a Restore statement and the TAB( ) statements is used in lines 260, 330, and 400. The source code is 2K Bytes in length and Stat 10 will execute in 3K of memory.

STAT 10

```
10   REM  ***DESCRIPTION:  COMPUTES THE MEANS, STANDARD ERROR OF MEANS,
20   REM  MEAN DIFFERENCE, STANDARD ERROR OF DIFFERENCE, AND
30   REM  T-RATIO FOR TWO GROUPS OF DATA, PAIRED.  THIS PROGRAM ASSUMES
40   REM  THAT THE TWO GROUPS HAVE AN EQUAL VARIANCE.
60   REM  ***INSTRUCTIONS FOR USE:  PUT IN DATA AS ORDERED
70   REM  PAIRS IN LINE 900 AND FOLLOWING, MAKING SURE THAT YOUR DATA
80   REM  LINE NUMBERS DO NOT EXCEED 997.
90   READ T
92   RESTORE
94   IF T <> 999999 THEN 100
96   PRINT "LIST LINES 10 TO 80 FOR INSTRUCTIONS"
98   STOP
100  READ X
110  IF X = 999999 THEN 210
120  READ Y
130  LET N = N + 1
140  LET S1 = S1 + X
150  LET S2 = S2 + X*X
160  LET U1 = U1 + Y
170  LET U2 = U2 + Y*Y
180  LET D1 = D1 + X - Y
190  LET D2 = D2 + (X -Y)↑2
200  GO TO 100
210  LET S3 = S1/N
220  LET S4 = (N*S2 - S1*S1)/N/(N-1)
230  LET S5 = SQR(S4)
240  LET S6 = SQR(S4/N)
250  PRINT "GROUP","NUMBER","MEAN        STD DEVIATION    STD ERROR MEAN"
260 PRINT TAB(0);1;TAB(14);N;TAB(26);S3;TAB(43);S5;TAB(60);S6
270  PRINT
280  LET U3 = U1/N
290  LET U4 = (N*U2 - U1*U1)/N/(N-1)
300  LET U5 = SQR(U4)
310  LET U6 = SQR(U4/N)
320  PRINT "GROUP","NUMBER","MEAN        STD DEVIATION    STD ERROR MEAN"
330 PRINT TAB(0);2;TAB(14);N;TAB(26);U3;TAB(43);U5;TAB(60);U6
340  PRINT
350  LET D3 = D1/N
360  LET D4 = (N*D2 - D1*D1)/N/(N-1)
370  LET D5 = SQR(D4)
380  LET D6 = SQR(D4/N)
390 PRINT " MEAN DIFFERENCE        VARIANCE OF DIFF,","STD ERROR OF DIFF,"
400 PRINT D3;TAB(28);D4;TAB(55);D6
410  PRINT
420  LET T = D3/D6
430  PRINT "T-RATIO =";T; "ON";N-1;"DEGREES OF FREEDOM."
998 DATA 999999,999999,999999
999  END
```

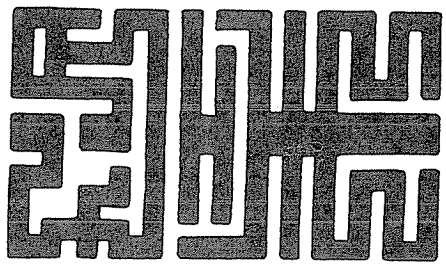
EXAMPLE:

PROBLEM

TO ANALYZE TWO GROUPS OF DATA: (1,2,3,4,5,6) AND (2,3,5,5,6,7)

SAMPLE RUN

```
900 DATA  1,2,  2,3,  3,5,  4,5,  5,6,  6,7
RUN

GROUP      NUMBER       MEAN       STD. DEVIATION      STD. ERROR MEAN
  1           6          3.5         1.870829             .7637626

GROUP      NUMBER       MEAN       STD. DEVIATION      STD. ERROR MEAN
  1           6        4.666667      1.861899             .7601169

MEAN DIFFERENCE      VARIANCE OF DIFF.      STD. ERROR OF DIFF.
  -1.166667              .1666667               .1666667

T-RATIO =  -7 ON  5 DEGREES OF FREEDOM
```



STAT 11:

DESCRIPTION

Stat 11 determines probabilities of an unknown population mean using sample statistics.

USERS

Statisticians would find this program to be useful for rapidly determining limits on various populations.

INSTRUCTIONS

The program is self prompting and should be listed for additional instructions. After Stat 11 has been loaded into memory type RUN. The program will ask for inputs for five variables N,M,S,W and X. Where:

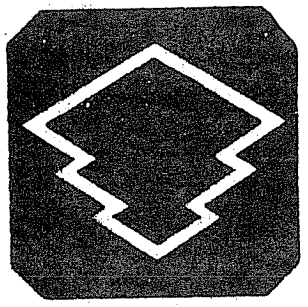
N - is the sample size
M - is the sample mean
S - is the standard deviation
W - is the population size (Ø is infinite)
X - is the population mean to be tested

LIMITATIONS

Line 190 contains a DEF FN__ statement and is used throughout this program. Line 820 contains an ABS( ) statement. The source code requires 3K Bytes for storage and will execute in 5K.

```
110 DATA 5000000,5398278,5792597,6179114,6554217,6914625,7257469
120 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
130 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
140 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
150 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
160 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
170 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
180 DIM X(49)
190 DEF FND(U)=X(U)-X(U-1)
200 DEF FNT(U)=1-((U↑2)+1)/(4*D)+(13*(U↑2)↑2+8*(U↑2)+3)/(96*D↑2)
210 FOR I = 1 TO 49
220 READ X(I)
230 NEXTI
240 PRINT "INSTRUCTIONS ? (1=YES, 0=NO) ..... WHICH ";
250 INPUT O0
260 IF O0=0 THEN 420
270 IF O0=1 THEN 320
280 LET O1=O1+1
290 IF O1>2 THEN 9999
300 PRINT "--940. ILLEGAL INPUT CHARACTER IN BIPPY. RETYPE IT.";
310 GO TO 250
320 PRINT
330 PRINT "THIS PROGRAM TESTS AN UNKNOWN POPULATION MEAN USING SAMPLE ";
340 PRINT "STATISTICS."
350 PRINT
360 PRINT "IT ASKS YOU FOR N, M, S, W, AND  X"
370 PRINT "WHERE  N = SAMPLE SIZE,"
380 PRINT "       M = SAMPLE MEAN,"
390 PRINT "       S = SAMPLE STANDARD DEVIATION,"
400 PRINT "       W = POPULATION SIZE [ZERO, IF INFINITE],"
410 PRINT "       X = POPULATION MEAN TO BE TESTED."
420 PRINT
430 PRINT "N, M, S, W, X = ";
440 INPUT N,M,S,W,X
450 LET D=N-1
460 PRINT
470 PRINT"BASED ON STUDENT'S T-DISTRIBUTION WITH";D;" DEGREES OF FREEDOM,"
480 PRINT"THE PROBABILITY OF FINDING A SAMPLE MEAN THIS MUCH ";
490 IF M<X THEN 520
500 PRINT "GREATER";
510 GOTO 530
520 PRINT "LESS";
530 PRINT " THAN THE"
540 PRINT "POPULATION MEAN IS";
550 IF W>0 THEN 570
560 LET W=1E25
```

```
570 LET S=S*SQR((W-1)/(W*D))
580 LET B1=(M-X)/S
590 LET B1=B1*FNT(B1)
600 GOSUB 680
610 IF B2<.5 THEN 630
620 LET B2=1-B2
630 IF B2<1E-5 THEN 660
640 PRINT 1E-5*INT(.5+1E5*B2)
650 STOP
660 PRINT "LESS THAN 1 IN 100,000."
670 STOP
680 IF B1<-4.5 THEN 800
690 IF B1<0 THEN 770
700 IF B1<4.5 THEN 740
710 LET B2=1
720 GOTO 810
730 IF 00=1 THEN 320
740 GOSUB 820
750 LET B2=Q
760 GOTO 810
770 GOSUB 820
780 LET B2=1-Q
790 GOTO 810
800 LET B2=0
810 RETURN
820 LET Z=10*ABS(B1)
830 LET K=INT(Z)
840 LET D1=Z-K
850 LET Q = X(K+1) + D1*FND(K+2) + (D1*(D1-1)/2)*(FND(K+3)-FND(K+1))
860 LET Q=1E-6*INT(.5+.1*Q)
870 RETURN
9999 END
```



STEEL:

DESCRIPTION

Steel calculates a number of quantities for steel sections used as beams and supports. It performs its' calculations for any "I" or "WF" steel section. For beams it can compute stresses not included in charts or tables in the AISC Handbook.

USERS

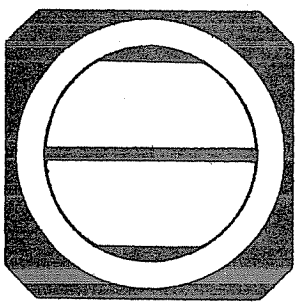
Homeowners, construction companies, engineers or anyone who is planning to design any structure requiring some form of steel supports will find this program both useful and helpful.

INSTRUCTIONS

The program is self instructing and features a full prompting mode. List the program for instructions or type RUN and it will ask you if you would like to know how to enter your data. Before you can solve your problem your data must be entered in data statements starting in line 1400.

LIMITATIONS

The source code for Steel requires 5K Bytes for storage and the program should execute in most Basic systems with 7K of available memory.



STEEL

```
10 DIM F(10),T(10),S(10),C(10),L(10)
070 PRINT"IF YOU WISH METHOD FOR ENTERING DATA TYPE 1,"
075 PRINT"IF NOT TYPE 2."
080 INPUT F9
090 IF F9>1 THEN 112
100 PRINT"THIS PROGRAM COMPUTES FOR WF AND I SECTIONS RES. MOM.CAP."
101 PRINT"AXIAL L'D CAP., AND SOLVES FORMULAS 6,7A/7B FOR COMBINED"
102 PRINT"DIRECT STRESS AND BENDING FOR ANY YP STRESS."
103 PRINT"ENTER DATA AS FOLLOWS:"
104 PRINT"1400 DATA M (AB'T X AXIS), M(AB'T Y AXIS), AXIAL L'D, YP"
105 PRINT"IN PSI, C SUB M. IF M(X)>0."
106 PRINT"1410 DATA B(FL'GE), T(WEB),T(FL'GE),D,S(X),L IN INCHES."
107 PRINT"IF P>0, 1420 DATA A,L(X),L(Y),R(X),R(Y),K(X),K(Y),0(0 FOR"
108 PRINT"SECONDARY MEMBER, 1 FOR MAIN MEMBER)"
109 PRINT"ENTER MOM. IN FF AND AXIAL L'D IN PDS."
110 PRINT"PROGRAM CURRENTLY NOT SET TO CHECK FOR BENDING AB'T"
111 PRINT"Y AXIS. SET M(AB'T Y AXIS)=0."
112 LET Y=0
113 LET W=1
114 LET F=0
115 LET X=0
116 LET U=0
120 READ M4,M5,P1,F(Y),C3
122 IF M4=0 THEN 130
123 READ B,T(W),T(F),D,S(X),L
125 IF P1=0 THEN 150
130 READ A2,L1,L2,R1,R2,K1,K2,Q
150 LET C=SQR(2*3.1416↑2*29E+6/F(Y))
160 LET C(1)=INT(10*C+.5)/10
170 PRINT"C SUB C ="C(1)
200 IF M4=0 THEN 790
210 LET A=(B-T(W))/(2*T(F))
220 IF A>1600/SQR(F(Y)) THEN 430
230 LET J=D/T(W)
240 IF J>13300/SQR(F(Y)) THEN 430
250 LET G=2400*B/SQR(F(Y))
260 LET H=2E+7*B*T(F)/(D*F(Y))
270 IF G>H THEN 320
280 LET G1=INT(10*G+.5)/10
290 PRINT"L(C) IN INCHES="G1
300 IF L>G THEN 430
310 GO TO 350
320 LET H1=INT(10*H+.5)/10
330 PRINT"L(C) IN INCHES="H1
340 IF L>H THEN 430
350 LET F3=.66*F(Y)/1000
360 LET F4=INT(F3+.5)
```

```
370 PRINT"F(B) IN KSI="F4
375 LET F5=F4
380 LET G3=F4
390 IF P1>0 THEN 790
400 LET M=S(X)*F4/12
410 PRINT"M IN KF="M
420 GO TO 710
430 LET F1=.6*F(Y)/1000
440 LET F5=INT(F1+.5)
450 PRINT"F(B) IN PSI="F5
460 LET G3=F5
480 LET K=12E+3*(B*T(F))/(D*F5)
490 LET L(U)=INT(K)
500 PRINT"L(U) IN INCHES="L(U)
510 IF L(U)<L THEN 550
515 IF P1>0 THEN 790
520 LET M2=S(X)*F5/12
530 PRINT"M2 IN KF="M2
540 GO TO 9999
550 LET P=SQR((T(F)*B↑3)/(12*B*T(F)+T(W)*(D-2*T(F)/6)))
560 LET R=INT(100*P+.5)/100
570 PRINT"R="R
580 LET N=L/R
590 IF N<40 THEN 630
600 PRINT"VALUE OF C(B) IN FORMULA 4 IS";
610 INPUT C2
620 LET I=(1-(L/R)↑2/(2*C(1)↑2*C2))*.0006*F(Y)
630 LET Z=12E+3*B*T(F)/(L*D)
640 IF Z>I THEN 720
650 LET F6=INT(10*I+.5)/10
660 PRINT"F(B) FROM FORMULA 4 IN KSI="F6
670 LET G3=F6
680 IF P1>0 THEN 790
690 LET M3=S(X)*F6/12
700 PRINT " M3 IN KF= "M3
710  GO TO 9999
720 LET F(2)=INT(Z)
730 PRINT"F(2) FROM FORMULA 5 IN KSI="F(2)
740 LET G3=F(2)
750 IF P1>0 THEN 790
760 LET M1=S(X)*F(2)/12
770 PRINT"M1 IN KF="M1
780 GO TO 9999
790 IF Q>0 THEN 980
830 LET A3=L1/R1
840 PRINT"LX/RX="A3
850 LET A4=L2/R2
860 PRINT"LY/RY="A4
870 IF A3>=A4 THEN 900
880 LET A5=A3
890 IF A3<A4 THEN 910
```

```
900 LET A5=A4
910 IF A5<=120 THEN 980
920 IF A5>C(1) THEN 1120
930 LET S=1.67+3*A5/(8*C(1))-A5↑3/(8*C(1)↑3)
933 PRINT"F.S.="S
935 LET F9=(1-A5↑2/(2*C(1)↑2))*F(Y)/(S*(1.6-A5/200))
940 GO TO 1175
980 LET A3=K1*L1/R1
990 PRINT"K1*L1/R1="A3
1000 LET A4=K2*L2/R2
1010 PRINT"K2*L2/R2="A4
1020 IF A3<=A4 THEN 1050
1030 LET A5=A3
1040 IF A3<A4 THEN 1060
1050 LET A5=A4
1060 IF A5>C(1) THEN 1170
1065 LET S=1.67+3*A5/(8*C(1))-A5↑3/(8*C(1)↑3)
1067 PRINT"F.S.="S
1070 LET F9=(1-A5↑2/(2*C(1)↑2))*F(Y)/S
1075 GO TO 1175
1120 LET F9=(149E+6/A5↑2)/(1.6-A5/200)
1125 GO TO 1175
1170 LET F9=149E+6/A5↑2
1175 PRINT"F(A)="F9
1180 IF M4>0 THEN 1260
1190 LET P2=F9*A2
1200 PRINT"P="P2
1210 IF P2<P1 THEN 1240
1220 LET Q1=(P2-P1)/P1*100
1230 PRINT"(FA-P)/P="Q1"PERCENT UNDERSTRESS"
1235 GO TO 9999
1240 LET Q1=(P1-P2)/P1*100
1250 PRINT"(P-FA)/P="Q1"PERCENT OVERSTRESS"
1255 GO TO 9999
1260 LET G2=P1/A2
1270 LET G4=G2/F9
1271 IF J>13300*(1-1.43*G4)/SQR(F(Y)) THEN 1390
1275 PRINT"SMALL FA/CAP FA="G4
1290 LET G5=M4*12/S(X)
1295 IF G4>.15 THEN 1330
1300 LET G6=G4+G5/(G3*1000)
1310 PRINT"FORMULA(6)="G6
1320 GO TO 9999
1330 LET G7=149E+6/A3↑2
1340 LET G8=G2/G7
1350 LET G9=G4+C3*G5/((1-G8)*G3*1000)
1360 PRINT"FORMULA 7A="G9
1363 LET F1=.6*F(Y)/1000
1365 LET F5=INT(F1+.5)
1370 LET H2=G2/(F5*1000)+G5/(G3*1000)
1380 PRINT"FORMULA 7B="H2
1385 GO TO 9999
1390 PRINT"D/T(WEB) TOO GREAT"
1400 DATA 40000,0,20000,36000,.6
1410 DATA 6.5,.24,.4,11.96,34.1,144
1420 DATA 7.97,144,144,5.06,1.44,1,1,1
9998 REM
9999 END
```

TOP:

DESCRIPTION

Top calculates the cost of materials required to pave a road or other similar surface. This surface could be a driveway, shopping center parking lot, or even a highway. Top will also compute the number of tons of material needed to do the paving.

USERS

Homeowners, construction companies and state highway departments will all be able to put this program to good use.

INSTRUCTIONS

The program is self prompting and will prompt for all required inputs. Line 110 sets the amount of material; in tons, required for a one inch covering per square yard. This quantity is set at 0.055 but may be changed to suit your needs; it is material dependent. After Top has been loaded into memory type RUN.

LIMITATIONS

Lines 790 and 870 contain TAB( ) statements and lines 810, 830, 870, and 880 contain Print Using statements. The source code for Top will store in 5K Bytes of memory and will require 9K Bytes of memory for execution.

TOP

```
10REM
20REM THIS BASIC PROGRAM CALCULATES THE COST AND NUMBER OF TONS
30REM OF PAVEING MATERIAL THAT WILL BE NEEDED TO PAVE A ROAD.
40REM
50REM*************************************************************************
60REM
70REM INSTRUCTIONS--TYPE RUN.  THE PROGRAM WILL ASK FOR THE NEEDED
71REM INFORMATION.  AFTER THE DATA IS ENTERED THE OUTPUT WILL BE PRINTED.
72REM
73REM
74REM
80REM
90 REM
95REM
96REM********************* MAIN PROGRAM ****************************
99REM
100 DIM W(50),L(50),T(50),M(50),S(50),V(50)
105REM          TONS/SQ.-YD.,FOR 1 IN. COVERING
110 LET C1=.055
120 PRINT"  DO YOU HAVE PRICES AVAILABLE FOR MIX";
130 INPUT A$
140 IF A$="NO" THEN 240
150 LET J8=1
160 PRINT
170 PRINT"  ENTER PRICE/TON OF 1/2 INCH STONE MIX";
180 INPUT P1
190 PRINT
200 PRINT"  ENTER PRICE/TON OF 3/4 INCH STONE MIX";
210 INPUT P2
220 PRINT
230 GOTO 260
240 LET J8=2
250 GOTO 260
260 PRINT"  ENTER THE TOTAL NUMBER OF SEGMENTS"
270 PRINT"TO BE PAVED";
280 INPUT N
290 LET A=1
300 PRINT
310 FOR I=1 TO N
320 IF I>1 THEN 530
330 PRINT "
            SEGMENT NO. 1
                          "
340 PRINT"  ENTER WIDTH OF SEGMENT NO.";I;
350 INPUT W(I)
360 GO TO 380
```

```
370 LET W(I)=W
380 PRINT
390 PRINT"  ENTER THE LENGTH OF SEGMENT NO.";I;
400 INPUT L(I)
410 PRINT
420 PRINT"  ENTER COVERING THICKNESS IN INCHES THAT IS DESIRED"
430 PRINT "FOR SEGMENT NO.";I;".USE DECIMAL NOTATION FOR "
440 PRINT "FRACTIONS OF AN INCH  ";
450 INPUT T(I)
460 PRINT
470 PRINT"FOR SEGMENT NO. ";I;", SPECIFY STONE MIX BY TYPING"
480 PRINT" '1' FOR 1/2 IN. MIX, OR '2' FOR 3/4 IN. MIX. ";
490 INPUT S(I)
500 PRINT
510 GOTO 590
520 PRINT
530 IF I>2 THEN 570
540 PRINT"FOR SEGMENTS 2 -";N;"ENTER FOUR PIECES OF DATA.  THE WIDTH"
550 PRINT"THE LENGTH, THE THICKNESS, AND THE STONE MIX."
560 PRINT
570 PRINT"SEGMENT NO.";I;
580 INPUT W(I),L(I),T(I),S(I)
585 REM            SQ. YDS. OF SURFACE
590 LET V(I)=(L(I)*W(I))/9
600 LET N1=N1+V(I)
605REM           TONS FOR THIS SEGMENT
610 LET M(I)=V(I)*C1*T(I)
620 IF S(I)=2 THEN 650
625REM         'CUM. TOTAL OF 1/2 IN. MIX
630 LET S1=S1+M(I)
640 GOTO 660
645REM           'CUM. TOTAL OF 3/4 IN. MIX
650 LET S2=S2+M(I)
660 REM
670 NEXT I
680 GOSUB 700
690 GOTO 940
700 PRINT
710 PRINT
720 PRINT
730 PRINT " ","TONS","TONS"
740 PRINT "SEGMENT","1/2 INCH","3/4 INCH","SQ.YDS. TO"
750 PRINT"NUMBER","STONE MIX","STONE MIX","PAVE"
760 PRINT"=============================================================="
770 PRINT
780 FOR I=1 TO N
790 PRINT TAB(1);I;
800 IF S(I)=2 THEN 830
810 PRINT USING 835,TAB(13);M(I);TAB(43);V(I)
820 GO TO 840
830 PRINT USING 835,TAB(27);M(I);TAB(43);V(I)
```

```
835:######.###
840 PRINT
850 NEXT I
860 PRINT" ","---------","---------","---------"
865 PRINT "TOTALS";
870 PRINT USING 835,TAB(13);S1;TAB(27);S2;TAB(43);N1
880 PRINT USING 890,(P1*S1)+(P2*S2)
890:TOTAL COSTS--$#########.##
900 PRINT
910 PRINT
920 PRINT
930 RETURN
940 PRINT"  DO YOU WISH TO MAKE ANY CHANGES IN"
950 PRINT"COVERING THICKNESS OR STONE SIZE MIX"
960 PRINT"FOR ANY SEGMENT";
970 INPUT A$
980 IF A$="NO" THEN 1710
990 PRINT
1000 PRINT"  IN HOW MANY SEGMENTS DO YOU WISH TO MAKE CHANGES";
1010 INPUT R
1020 PRINT
1030 PRINT"  FOR EACH CHANGE, ENTER SEGMENT NUMBER TO BE"
1040 PRINT"CHANGED, THE NEW THICKNESS IN INCHES (DECIMAL"
1050 PRINT"FOR FRACTIONS OF AN INCH), AND THE NEW STONE"
1060 PRINT"SIZE MIX (1 FOR 1/2 IN.,2 FOR 3/4 IN.). IF YOU"
1070 PRINT"WISH TO CHANGE ONLY ONE OF THESE, TYPE '0' FOR "
1080 PRINT"VARIABLE NOT TO BE CHANGED"
1090 LET B=1
1100 PRINT
1110 PRINT
1120 FOR I=1 TO R
1130 PRINT" FOR CHANGE NO. ";I;
1140 INPUT X,Y,Z
1150 IF Y=0 THEN 1290
1160 IF Z=0 THEN 1390
1170 IF Z=2 THEN 1230
1180 LET S2=S2-M(X)
1190 LET S(X)=1
1200 LET M(X)=V(X)*C1*Y
1210 LET S1=S1-M(X)
1220 GO TO 1270
1230 LET S1=S1-M(X)
1240 LET S(X)=2
1250 LET M(X)=V(X)*C1*Y
1260 LET S2=S2+M(X)
1270 LET T(X)=Y
1280 GOTO 1530
1290 IF Z=0 THEN 1500
1300 IF Z=2 THEN 1350
1310 LET S2=S2-M(X)
1320 LET S1=S1+M(X)
```

```
1330 LET S(X)=1
1340 GOTO 1530
1350 LET S1=S1-M(X)
1360 LET S2=S2+M(X)
1370 LET S(X)=2
1380 GOTO 1530
1390 IF S(X)=2 THEN 1450
1400 LET S1=S1-M(X)
1410 LET M(X)=U(X)*C1*Y
1420 LET T(X)=Y
1430 LET S1=S1+M(X)
1440 GOTO 1530
1450 LET S2=S2-M(X)
1460 LET M(X)=U(X)*C1*Y
1470 LET T(X)=Y
1480 LET S2=S2+M(X)
1490 GOTO 1530
1500 PRINT"  YOU HAVE NOT REQUESTED A CHANGE FOR THIS"
1510 PRINT"SEGMENT - RETYPE IT."
1520 GOTO 1130
1530 NEXT I
1540 PRINT
1550 PRINT
1560 PRINT
1570 PRINT" RESULTS FOR ALTERATION NO. ";B
1580 PRINT
1590 GOSUB 700
1600 PRINT"  DO YOU WISH TO MAKE FURTHER CHANGES";
1610 INPUT A$
1620 IF A$="NO" THEN 1710
1630 LET B=B+1
1640 PRINT
1650 PRINT
1660 PRINT
1670 PRINT"  IN HOW MANY SEGMENTS DO YOU WISH TO MAKE CHANGES";
1680 INPUT R
1690 PRINT
1700 GOTO 1110
1710 END
```

VARY:

DESCRIPTION

This program analyzes a variance table of one-way random design. Identical values or observations may be grouped together, followed by the number of times that value occured; when entering data.

USERS

Statisticians and engineers will be able to put this program to good use as well as individuals whose studies produce data requiring reduction.

INSTRUCTIONS

Before the program is run the data must be entered into data statements, starting in line 900. The data is entered in the following sequence:

```
900  DATA  A,M
910  DATA  N1,N2,.....NM
920  DATA  T1,T2,.....TM
```

where

A - is the total number of observations
M - is the number of different runs
N - is the number of observations per run
T - is the number of different observations per run

Then enter the data starting with the first point of the first run, then the second point, and so on, as:

```
930  DATA  R1(1),R1(2),R1(3).....
940  DATA  R2(1),R2(2),..........
9..  DATA  RM(1),RM(2),..........
```

If there are more then 10 runs and/or if there are more then 20 observations per run, change the DIM statements in line 100 to accomodate your data size. REMEMBER increasing the DIM statements increases the amount of memory required for execution.

LIMITATIONS

This program uses two dimensional arrays, starting in line 100. Lines 115 and 117 contain MAT READ statements. If your system does not have these statements refer to Appendix A for a definition of the statements and/or Appendix B for conversion listings. The source code requires 3K Bytes for storage and 9K for execution, with the DIM statements set as they presently are.

VARY

```
10  REM  DESCRIPTION: COMPUTES THE ANALYSIS OF VARIANCE TABLE
20  REM  FOR A ONE-WAY COMPLETELY RANDOMIZED DESIGN.
30 REM THIS IS EXACTLY THE SAME AS Variance 2 EXCEPT THAT THE OBSERVATIONS
31 REM INSTEAD OF CONSISTING OF A SIMPLE LIST, CONSIST OF A LIST OF
32 REM PAIRS OF NUMBERS, THE 1ST OF WHICH IS AN OBSERVATION VALUE,
33 REM AND THE 2ND OF WHICH IS THE NUMBER OF OBSERVATIONS TAKING ON
34 REM THAT VALUE.
40  REM  INSTRUCTIONS: ENTER DATA IN LINE 900 AND FOLLOWING.
50  REM  ENTER DATA IN THE FOLLOWING ORDER:
60  REM  1) A, THE TOTAL NUMBER OF OBSERVATIONS
70  REM  2) M, THE NUMBER OF DIFFERENT TREATMENTS
80  REM  3) N(1),...,N(M), WHERE N(J) IS THE NUMBER OF OBSERVATIONS
81  REM      IN TREATMENT J
82 REM 4) V(1),...,V(M), WHERE V(J) IS THE NUMBER OF DIFFERENT
83 REM    VALUES TAKEN ON BY THE OBSERVATIONS IN TREATMENT J
84 REM 5) THE OBSERVATIONS THEMSELVES, 1ST FOR TREATMENT 1,THEN FOR
85 REM    TREATMENT 2,ETC.  FOR EACH TREATMENT THESE WILL BE IN THE
86 REM    FORM OF A LIST OF PAIRS OF NUMBERS, THE 1ST OF WHICH IS AN
87 REM    ACTUAL OBSERVATION VALUE TAKEN ON B9 AT LEAST ONE OBSERVATION
88 REM    IN THAT TREATEMENT, AND THE 2ND OF WHICH IS THE NUMBER
89 REM    OF OBSERVATIONS IN THAT TREATMENT TAKING ON THAT VALUE.
90 REM IF ANY N(J) >20 CHANGE THE DIMS IN LINE 100
91 REM IF M >10,CHANGE THE DIMS IN  LINE 100
100 DIM X(20,10),N(10),T(10),S(10),Y(20,10),V(10)
110 READ A, M
115    MAT READ N(M)
117 MAT READ V(M)
120 FOR J = 1 TO M
130 FOR I = 1 TO V(J)
140 READ X(I,J),Y(I,J)
150 NEXT I
160 NEXT J
170 FOR J = 1 TO M
180 FOR I = 1 TO V(J)
190 LET T(J) = T(J) + X(I,J)*Y(I,J)
200 LET S(J) = S(J) + X(I,J)*X(I,J)*Y(I,J)
210 NEXT I
220 LET U=U+T(J)
230 LET R=R+S(J)
240 LET V=V+T(J)*T(J)/N(J)
250 NEXT J
260 LET C = U*U/A
270 LET W = V - C
280 LET E = R - V
290 PRINT "ANOVA TABLE:"
300 PRINT
310 PRINT "ITEM","         SS","          DF","      MS"
320 PRINT
```

```
330 PRINT "GRAND TOTAL";R;A
340  PRINT "GRAND MEAN";C;1
350 PRINT "TREATMENTS"; W; M-1; W/(M-1)
360 PRINT "ERROR";E;A-M;E/(A-M)
370 PRINT
380 PRINT
390 LET F = (W/(M-1))/(E/(A-M))
400 PRINT "F ="F"ON"M-1"AND"A-M"DEGREES OF FREEDOM."
402 LET G=F
403 LET N=A-M
404 LET M=M-1
405 GOSUB 800
410 STOP
800 REM
802 LET P=1
803 IF G<1 THEN 808
804 LET A=M
805 LET B=N
806 LET F=G
807 GO TO 811
808 LET A=N
809 LET B=M
810 LET F=1/G
811 LET A1=2/(9*A)
812 LET0B1=2/(9*B)
813 LET Z=ABS((1-B1)*F↑(.333333)-1+A1)
814 LET Z=Z/SQR(B1*F↑(.666667)+A1)
815 IF B<4 THEN 819
816 LET P=(1+Z*(.196854+Z*(.115194+Z*(.000344+Z*.019527))))↑4
817 LET P=.5/P
818 GO TO 821
819 LET Z=Z*(1+.08*Z↑4/B↑3)
820 GO TO 816
821 IF G<1 THEN 823
822 GO TO 825
823 LET P=1-P
824 GO TO 825
825 PRINT
826 LET P = INT(100000*P)/100000
827 PRINT "EXACT PROB. OF F=";G;"WITH ( "M;", "N;" ) D.F. IS ";P
828 PRINT
829 RETURN
99999END
```

SAMPLE RUN

```
900  DATA  25,5
905  DATA  2,6,11,4,2
907  DATA  2,4,5,3,2
910  DATA  83,1, 85,1
920  DATA  84,1, 85,2, 86,2, 87,1
930  DATA  86,1, 87,3, 88,5, 89,1, 90,1
940  DATA  89,1, 90,2, 91,1
950  DATA  90,1, 92,1
RUN
```

ANOVA TABLE:

ITEM	SS	DF	MS
GRAND TOTAL	191791	25	
GRAND MEAN	191668.8	1	
TREATMENTS	99.02344	4	24.75586
ERROR	23.13672	20	1.156836

F = 21.39963 ON 4 AND 20 DEGREES OF FREEDOM.

EXACT PROB. OF F = 21.39963 WITH (4, 20) D.F. IS 1.000000E-05

XMAS:

DESCRIPTION

This is a sing-a-long Christmas card. It not only prints the lyrics to the song but it also draws small pictures between each verse and includes a special ending.

INSTRUCTIONS

The program is ready to run upon being loaded into memory, just type RUN.

LIMITATIONS

This program should execute without problem in most 4K Basic systems. The source code requires 5K to store and will execute in 6K Bytes of available memory. Immediately following the source code listing is a partial executed run of XMAS. Only a part of the sample run is included as the program generates a very unusual ending. Thus it will be necessary for you to run the program in order to find out what happens at the end.

XMAS

```
100REM  XMAS    BASIC PROGRAM BEGINS AT LINE 210
110REM
120REM  A SING-ALONG-PROGRAM
130REM
140REM DESCRIPTION: THIS BASIC PROGRAM PRINTS OUT A COPY OF
150REM               "THE TWELVE DAYS OF CHRISTMAS" ADORNED
155REM               WITH APPROPRIATE HOLIDIDAY SYMBOLS. IT IS
160REM               SUITABLE AS AN UNUSUAL CHRISTMAS CARD OR AS
170REM             SHEET MUSIC FOR A SING-ALONG.
175REM-----------------------------------------------------------------
180REM INSTRUCTIONS--TYPE "RUN" AND SING-ALONG
190REM
195REM-----------------------------------------------------------------
200REM
210 PRINT"    E V E R Y B O D Y     S I N G"
220 PRINT
230 LET C=1
240 PRINT "ON THE";
250 IF C=1 THEN 290
260 IF C=2 THEN 310
270 IF C=3 THEN 330
280 IF C>3 THEN 350
290 PRINT C;"ST";
300 GO TO 360
310 PRINT C;"ND";
320 GO TO 360
330 PRINT C;"RD";
340 GO TO 360
350 PRINT C;"TH";
360 PRINT " DAY OF CHRISTMAS"
370 PRINT "MY TRUE LOVE SENT TO ME"
380 IF C=1 THEN 600
390 IF C=2 THEN 590
400 IF C=3 THEN 580
410 IF C=4 THEN 570
420 IF C=5 THEN 560
430 IF C=6 THEN 550
440 IF C=7 THEN 540
450 IF C=8 THEN 530
460 IF C=9 THEN 520
470 IF C=10 THEN 510
480 IF C=11 THEN 500
490 PRINT "TWELVE LORDS A-LEAPING,"
500 PRINT "ELEVEN LADIES DANCING,"
510 PRINT "TEN PIPERS PIPING,"
520 PRINT "NINE DRUMMERS DRUMMING,"
```

```
530 PRINT "EIGHT MAIDS A-MILKING,"
540 PRINT "SEVEN SWANS A-SWIMMING,"
550 PRINT "SIX GEESE A-LAYING,"
560 PRINT "FIVE GO-OLD RINGS,"
570 PRINT "FOUR COLLY BIRDS,"
580 PRINT "THREE FRENCH HENS,"
590 PRINT "TWO TURTLEDOVES AND"
600 PRINT "A PARTRIDGE IN A PEAR TREE."
610 PRINT
620 GOSUB 660
630 IF C=12 THEN750
640 LET C = C+1
650 GO TO 240
660 PRINT
670 PRINT"          0"
680 PRINT      "          *"
690 PRINT     "         ***"
700 PRINT                   "        *****"
710 PRINT"          I"
720 PRINT
730 RETURN
740 LET X=0
750 PRINT
760 PRINT
770 LET N=0
780 LET X=X+1
790 PRINT " "," ","          0"
800 PRINT"  "," ","          *"
810 PRINT " "," ",      "        ***"
820 PRINT " "," ",   "       *0***"
830 PRINT " "," ",         "     *******0*"
840 LET N=N+1
850 IF N=1 THEN 820
860 PRINT " "," ",    **0*******"
870 PRINT " "," ",      "   **0**0****0*"
880 PRINT " ", " ",         " *0*************"
890 LET N=N+1
900 IF N=3 THEN 860
910 PRINT " "," ",        "***0**0****0***0*"
920 PRINT " ",      "                *0*****0******0*****"
930 PRINT " ","                  *****0****0********0****"
940 IF N=4 THEN 880
950 PRINT " ","                 *0**********0****0***0**"
960 PRINT " ",      "              *****0**0**********0********0"
970 PRINT " ", "              *0****0*0****0**0**0*0****0***"
980 PRINT " ",    "             *****0****0**0****0******0******"
990 PRINT " "," ","0  0  III 0   0"
1000 FOR S=1TO3
1010 PRINT " "," ",         "        III"
1020 NEXT S
1030 PRINT
```

```
1040 PRINT
1050 IF X=2 THEN 1520
1060 PRINT"*      *   *****       *       *   *   *****   *   *       *     *   ***   *   *"
1070 PRINT "*   *   *   *           *   *   *   *   *       *   *         *  *   *   *  *   *"
1080PRINT"* * * *   ***         *  *  *  *   *     ***     *****           *     *   *  *   *"
1090 PRINT "**    **   *           **     **   *       *   *   *           *     *   *  *   *"
1100 PRINT "*      *   *****       *       *   *   *****   *   *           *      ***    ***"
1110 FOR M=1 TO 4
1120 PRINT
1130 NEXT M
1140 PRINT"               *         *       *   *****   ****    ****     *     *"
1150 PRINT "              * *       **     **   *       *   *   *   *     * *"
1160 PRINT  "             *   *      * * * * *   ***     ****    ****        *"
1170 PRINT"             ******      *   *   *   *       * *     * *       *"
1180 PRINT "             *   *      *       *   *****   *   *   *   *     *"
1190 FOR M=1 TO 4
1200 PRINT
1210 NEXT M
1220 PRINT"     ****   *   *   ****     *   *****   ******   *       *      *      *****"
1230 PRINT  "    *       *   *   *   *   *   *           *      **     **     * *    *"
1240 PRINT "    *       *****   ****     *    **       *      * * * * *    *     **"
1250 PRINT"    *       *   *   * *      *       *      *      *   *   *   ******       *"
1260 PRINT  "     ****   *   *   *   *   *   *****      *      *       *   *    *   *****"
1270 FOR M=1 TO 4
1280 PRINT
1290 NEXT M
1300 PRINT "                              *      *    *   ****"
1310 PRINT     "                            * *    **   *   *   *"
1320 PRINT"                           *    *   * * *   *   *"
1330 PRINT  "                           ******   *   **   *   *"
1340 PRINT"                           *    *   *    *   ****"
1350 FOR M=1 TO 4
1360 PRINT
1370 NEXT M
1380 PRINT"               *         *    *      *      ****    ****     *     *"
1390 PRINT  "               * *       *    *     * *     *   *   *   *    * *"
1400 PRINT "             *    *      ******   *    *   ****    ****        *"
1410 PRINT"             ******      *    *   ******   *       *          *"
1420 PRINT  "             *    *      *    *   *    *   *   *       *          *"
1430 FOR M=1 TO 4
1440 PRINT
1450 NEXT M
1460 PRINT  "        *    *   *****   *       *       *    *   ******      *      ****"
1470 PRINT"       **   *   *       *   *   *         * *    *         * *    *   *"
1480 PRINT "       * * *   ****    * * * * *           *      ****    *    *   ****"
1490 PRINT"       *   **   *       **     **          *      *       *******   * *"
1500 PRINT  "        *    *   *****   *       *          *      *****   *    *   *   *"
1510 GO TO 750
1520END
```

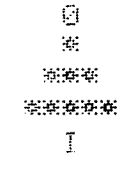
THE FOLLOWING IS THE PRINT OUT

FROM THE PROGRAM "XMAS" HO! HO! HO!

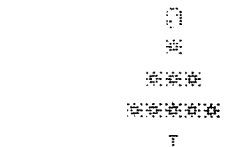
E V E R Y B O D Y S I N G

ON THE 1 ST DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
A PARTRIDGE IN A PEAR TREE.

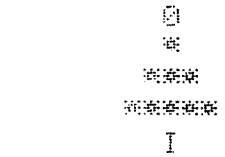
ON THE 2 ND DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
TWO TURTLEDOVES AND
A PARTRIDGE IN A PEAR TREE.



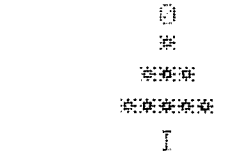
ON THE 3 RD DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
THREE FRENCH HENS,
TWO TURTLEDOVES AND
A PARTRIDGE IN A PEAR TREE.



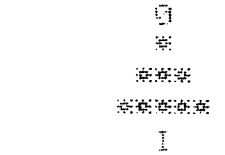
ON THE 4 TH DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
FOUR COLLY BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES AND
A PARTRIDGE IN A PEAR TREE.



ON THE 5 TH DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
FIVE GO-OLD RINGS,
FOUR COLLY BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES AND
A PARTRIDGE IN A PEAR TREE.



ON THE 6 TH DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME
SIX GEESE A-LAYING,
FIVE GO-OLD RINGS,
FOUR COLLY BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES AND
A PARTRIDGE IN A PEAR TREE.



APPENDIX B

STATEMENT CONVERSION ALGORITHMS

SEE APPENDIX A FOR STATEMENT DEFINITIONS

B A S I C C O N V E R S I O N S

This appendix is intended to aid the users of this Library in correcting any syntax errors they may encounter when running any of the programs in this Library. This appendix is divided into two sections - Direct Changes and Indirect Changes. A Direct Change is one that only requires changing of an alphanumeric character to another one or involves only the correction of a single line. These changes are fast and easy to make. Indirect Changes are those that require the statement in question to be deleted from the program and several lines of code substituted.

The conversions are not separately listed for the various machines and manufacturers as there are a number of similarities between the various systems. To use this listing, first isolate the statements from the program(s) that are not recognized by your compiler and then look up these statements in the Direct Change section. Now substitute the appropriate conversion for the program statement. If there is more then one conversion shown for the statement be sure the one you choose is referenced in the Basic Manual supporting your system. For statements not found in the Direct Change section go to the Indirect Change section and follow the same procedure.

The conversions are especially designed to allow full compatibility between these programs and a number of alien Basic compilers. If your system or compiler is not specifically mentioned it does not mean the conversions listed here are not applicable; on the contrary they probably are in most cases, it only means that at the time of this writing an operators manual for your system was not available. This listing will be enlarged with each additional printing to incorporate as many different systems as possible. The blank pages at the end of this section are reserved for future expansions. The following is a list of systems and/or Basic compilers that have been included in this appendix:

Polymorphics 8K
DEC RSTS - 11
BASIC PLUS
8K MITS
4K MITS
8K SWTPC
4K SWTPC
8K Processor Tech
4K Processor Tech
G.E. 635
Sigma 9
IBM 370
Honeywell 6000
Intercolor 8K
8K IMSAI
4K IMSAI
8K ZAPPLE
2100 Hewlett Packard
IBM 5100

DIRECT CHANGES*

As Used In Library	May Have To Change To:
1. RND(-X)	RND(X) or RND(Ø)
2. **	↑
3. A$ & B$	A$ + B$
4. SPC(X)	POS or POS(X) or see Indirect Changes
5. CLK$	(Requires real time clock) (Processor Dependent)
6. DAT$	(Requires real time clock) (Processor dependent)
7. TIM(X)	(Requires real time clock) (Processor dependent)
8. PRINT USING _ _ _ _	PRINT (and Remove the # # Lines)
9. PRINT "....."	PRINT '.....'
10. Line Numbers > 9999	Resequence numbers < 9999
11. SST(X$, Y, Z)	MID(X$,Y,Z) or MID$(X$,Y,Z) or STR(X$,Y,Z)
12. FNEND	(Remove if not in your Basic)
13. IF.....GOTO	IF.....THEN
14. A5$	A$(5) or A$[5]
15. MAT	See Indirect Changes
16. X(Y)	X[Y] Where Y is any integer
17. SQR( )	SQRT( )
18. RND( )	FRAND( )

* If your Basic does not have string functions omit their references.

INDIRECT CHANGES

1.) ON....GOTO 11Ø,125,135,...

Remove the above statement and insert the following routine:

```
if.... = 1 GOTO  11Ø
if.... = 2 GOTO  125
if.... = 3 GOTO  135
```

2.) CLG(X)

Remove the above statement and replace it with:

```
LOG(X)/(2.3025851)
```

3.) NUM(X)

This statement is used to count the number of data points that are input during a MAT READ statement. If you know the total number of data points that are read into the matrix, then this number represents NUM(X). If this statement is used with a MAT statement and your system doesn't have MAT statements place the following routine in the FOR....NEXT loop used to evaluate the MAT statement:

```
N9 = N9 + 1
```

Be sure N9 is set to zero before the FOR loop is entered. After the loop is done, N9 represents the value of the NUM( ) statement for that location. This procedure must be followed each time the NUM( ) statement is used.

4.) DEF FN__ = ...

In place of the above statement substitute the following:

```
XYZ  F__ = .....
```

where

```
F__ = FØ to F9
```

and XYZ is the line number where the DEF FN__ statement appears.

```
XYZ+1  IF  N = Ø  GOTO  XYZ+3
XYZ+2  RETURN
XYZ+3  N = 1
```

Then everytime FN__ is called substitute a GOSUB to XYZ. After the

return use F__ in place of the FN__ statement. If you have more then 10 different FN__, ie: FNA, FNB, ..., FNM, then use F(1) for FNA and F(2) for FNB, etc. Remember F( ) must be dimensioned for the maximum size needed, if it is greater then 10.

5.) .....\.....

The \ is a line separator and separates two statements placed on the same line. Leave the first portion of the line up to the \ mark as it is, then delete the portion of the line that appears after the \ . Add another line immediately below the first, number it one number larger then the first line and type in the portion of the line that you previously deleted, leaving out the \ mark completely.

6.) COT(X)

Remove the above statement and substitute it with:

```
1/TAN(X)
```

7.) SPC(X)

For most purposes this statement can be replaced by TAB(X). However, there are a few fine subtleties that distinguish the two statements. While these differences are very slight and should not come into play in any of the programs within this Library, if it should be necessary to generate the SPC(X) statement function it can be done by using the following algorithm:

```
      FOR I = 1 to X
    * PRINT  "Ø"
      NEXT  I
* or PRINT  'Ø'
```

where Ø represents a Space character.

8.) :###.##

This is a Print Using control line. There is no equivalent if your system does not offer a Print Using statement. If you do not have this statement, then delete this line and change the Print Using_ _ _ _ statement to a Print statement.

9.) Change A to A$

and A(Ø) = X where A is a table

Remove the above statement and insert the following routine:

```
FOR J = 1 to X
I = A(J)
B$(J) = STR$(I)
A$ = B$(J-1) + B$(J)
NEXT J
```

10.) Change A$ to A

Remove the above statement and insert the following routine:

```
   I = LEN(A$)
   FOR J = 1 to I
   A(Ø) = I
*  B$ = MID(A$,J,1)
   A(J) = VAL(B$)
   NEXT J
* or B$ = MID$(A$,J,1)
```

11.) ABS(X)

Remove the above statement and replace with the following:

```
XYZ  1F  X>Ø  GOTO  XYZ+2
XYZ+1  X = -X
XYZ+2  REM  ABS(X)
```

12.) MAT READ A

Where A is dimensioned as A(X,Y)

```
FOR I = 1 to X
FOR J = 1 to Y
READ  A(I,J)
NEXT  J
NEXT  I
```

Replace the MAT READ statement with the above algorithm. Be sure to enter the numeric values of the dimensions for X and Y in the above routine.

13.) MAT INPUT A

Where A is dimensioned as A(X,Y)

```
FOR I = 1 to X
FOR J = 1 to Y
INPUT A(I,J)
NEXT J
NEXT I
```

Replace the MAT INPUT statement with the above algorithm. Be sure to enter the numeric values of the dimensions for X and Y in the above routine.

14.) MAT A = ZER

Where A is dimensioned as A(X,Y)

```
FOR I = 1 to X
FOR J = 1 to Y
A(I,J) = Ø
NEXT J
NEXT I
```

Replace the MAT = ZER statement with the above algorithm. Be sure to enter the numeric values of the dimensions for X and Y in the above routine.

15.) MAT A = CON

Where A is dimensioned as A(X,Y)

```
FOR I = 1 to X
FOR J = 1 to Y
A(I,J) = 1
NEXT J
NEXT I
```

Replace the MAT = CON statement with the above algorithm. Be sure to enter the numeric values of the dimensions for X and Y in the above routine.

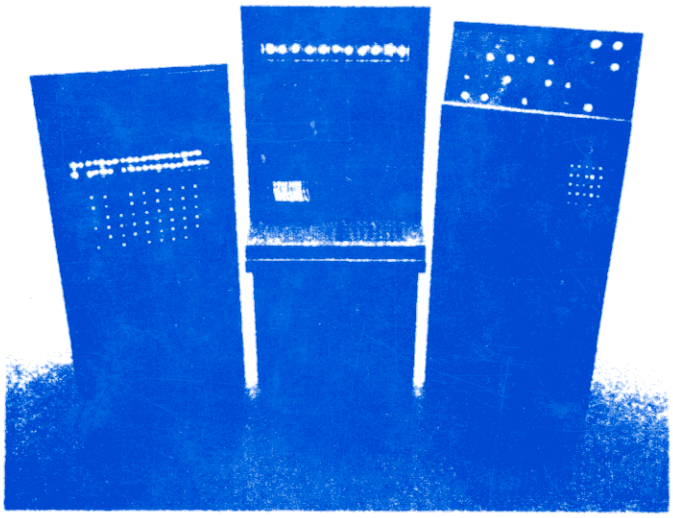
16.) MAT READ A,B,C

Where A,B and C are dimensioned as: A(X,Y), B(R,S) and C(V,W)

```
FOR I = 1 to X
FOR J = 1 to Y
READ  A(I,J)
NEXT  J
NEXT  I

FOR I = 1 to R
FOR J = 1 to S
READ  B(I,J)
NEXT  J
NEXT  I

FOR I = 1 to V
FOR J = 1 to W
READ  C(I,J)
NEXT  J
NEXT  I
```

Substitute the above routine for each MAT READ A,B,C statement. Be sure to enter the numeric values of the dimensions in these steps.

RELIABLE COMPUTER SOFTWARE



FOR YOUR DOWN TO EARTH TASKS

SCIENTIFIC RESEARCH INST. Crofton, Maryland BASIC SOFTWARE LIBRA