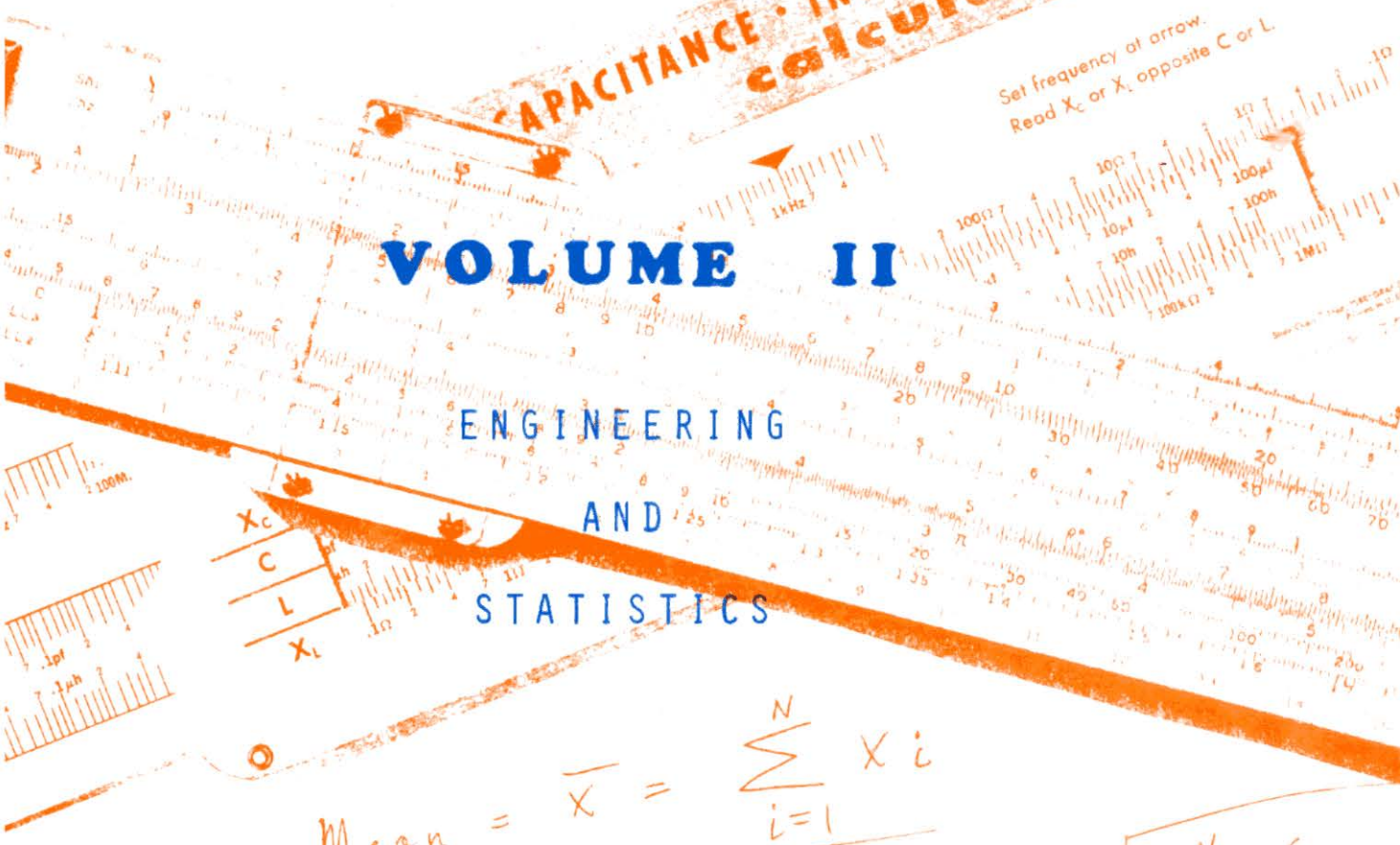
$24.95

BASIC SOFTWARE LIBRARY



VOLUME II

ENGINEERING AND STATISTICS

by R. W. BROWN

BASIC SOFTWARE LIBRARY

VOLUME II

ENGINEERING

AND

STATISTICS

THIS BASIC SOFTWARE LIBRARY IS MADE AVAIL
ABLE, FREE OF RESTRICTIONS AND ROYALTIES
TO SCHOOLS, COLLEGES, UNIVERSITIES, INDI-
VIDUALS, HOBBYIST & BUSINESS CONCERNS FOR
USE ON THEIR OWN COMPUTERS AND OR COMPUT-
ING SYSTEMS. REPRODUCTION IN ANY PART OR
FORM OF THIS ENTIRE LIBRARY IS STRICTLY
FORBIDDEN. USE OF ANY PART OR FORM OF THIS
ENTIRE LIBRARY FOR COMMERCIAL USE OF ANY
KIND IS STRICTLY FORBIDDEN WITHOUT THE EX
PRESSED WRITTEN PERMISSION OF SCIENTIFIC
RESEARCH.

1ST. PRINTING - - OCTOBER 1976
2ND. PRINTING - - DECEMBER 1976

COPYRIGHT UNDER UCC 1976 BY:

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

Preface

Part 1 - Business & Personal Bookkeeping Programs

NAME	DESCRIPTION
Bond	Computes price and interest for bond purchases.
Building	Analyzes the cost of building design proposals.
Compound	Computes effective compound interest rates.
Cyclic	Determines seasonal coefficients for two cycles.
Decision 1	Makes a lease/buy decision for you.
Decision 2	Makes a decision on whether to buy a component or make it.
Depreciation	Calculates depreciation by 4 different methods.
Efficient	Cal. the most efficient assignment of resources and/or personnel.
Flow	Predicts your yearly cash flow.
Installment	Performs monthly installment accounting.
Interest	Computes interest accruals, monthly.
Investments	Computes annual rates of return on investments.
Mortgage	Makes a comparison of mortgage terms.
Optimize	Optimizes the layout for a plant, shop, office, etc.
Order	Determines your economic order quantity for inventory items.
Pert Tree	Performs an analysis of a pert network.
Rate	Computes true annual interest rates.
Return 1	Computes lessor's rate of return for uncertain assets.
Return 2	Computes a lessor's rate of return after taxes.
Schedule 1	Schedules N jobs in a shop with M machines.

Part 2 - Games & Pictures

NAME	DESCRIPTION
Animals Four	Teach the computer all about animals.
Astronaut	Land your spaceship on another planet.
Bagel	Advanced number game, numbers may be algebraic, few clues.
Bio Cycle	Calculate your Bio-Life Cycle and plan your days.
Cannons	An advanced war game with big guns.
Checkers	Plays a regulation game of checkers.
Craps	A dice game with hard way odds.
Dogfight	Air fight w/missiles; betweeen a phantom and a mig.
Golf	Plays any number of holes; inc. obstacle course.
Judy	Have a rap session with Judy via your computer.
Line Up	Simple number game, all you have to do is unscramble them.
Pony	Authentic horse race, any number of players.
Roulette	Gamblers delight, plays Las Vegas rules.
Sky Diver	Sky dive on another planet
Tank	A war game between two tanks.
Teach Me	Teach the computer to learn new things.

TABLE OF CONTENTS

VOLUME ONE (CONT.)

PICTURES

NAME	DESCRIPTION
	Introduction
A. Newman	He's absolutely MAD! MAD! MAD!
J.F.K.	Our 35th. president.
Linus	Loveable "Peanuts" character, w/blanket.
Ms. Santa	A modern miss to put a twinkle in your eye.
Nixon	Former "United States" president.
Noel Noel	Christmas or anytime this is a beautiful creation.
Nude	A true work of art for anyone's gallery.
Peace	A message for all seasons.
Policeman	True and blue, he's the law.
Santa's Sleigh	In banner form, perfect for decorating the mantle.
Snoopy	That paragon of Dogdom even plays football.
Virgin	A picture you can read as well as see.

TABLE OF CONTENTS

VOLUME TWO

Part 3 - Math & Engineering Programs

NAME	DESCRIPTION	PAGE
Beam	Evaluates and selects steel beam sizes.	294
Conv.	Calculates convolutions.	300
Filter	Calculates low pass filter components.	304
Fit	Performs interpolations by spline fits.	308
Integration 1	Uses Gaussion Quadrature to do integration.	312
Integration 2	Integrates a function by spline fits.	315
Intensity	Calc. and plots RF or Acoustic intensities.	319
Lola	Calc. Long. and Lat. from interstellar fix or distance.	332
Macro	Simulates a language compiler.	338
Max. Min.	Calc. the max. & min. values of funct. over a spec. interval.	341
Navaid	Calc. position from altitude and azimuth of celestial bodies.	346
Optical	Calculates Blackbody energies, w/filter look-up tables.	350
Planet	Calculates Sun and Moon positions, hourly.	370
PSD	Calculates Power Spectral Densities and FFT's.	376
Rand 1	Generates random numbers between 0 and 1.	388
Rand 2	Generates random integers between (X) and (Y).	390
Solve	Solves polynomials by "Bairstows Method".	392
Sphere Trian	Solves any spherical triangle.	398
Stars	Locates 50 stars (celestial).	405
Track	Calc. course and distance and incremental vectors.	413
Triangle	Solves for all parts of any triangle.	418
Variable	Finds all variables in Basic programs.	423
Vector	Calc. final position; given start and motion vectors	426

TABLE OF CONTENTS

VOLUME TWO (CONT.)

Part 4 - Plotting & Statistics Programs

NAME	DESCRIPTION	PAGE
Binomial	Calculates binomial probability distributions.	430
Chi-Sq.	Applies the Chi-Square test to samples.	433
Coeff	Calc. coefficients of fourier series to apprx. a function.	437
Confidence 1	Calculates confidence limits on linear regressions.	444
Confidence 2	Calculates confidence limits for a sample mean.	448
Correlations	Performs auto and cross correlations with plots.	452
Curve	Fits 6 different curves by the least squares method.	461
Differences	Calculates difference of means in non-equal variances.	467
Dual Plot	Plots two functions on the same sheet.	472
Exp-Distri	Calculates exponential distributions for a sample.	476
Least Squares	Performs least squares fit by linear, exp., or power function.	479
Paired	Compares 2 groups of data using the rank test.	483
Plot	Plots 6 equations on the same sheet.	486
Plotpts	Plots data points on standard teletypes.	490
Polynomial Fit	Performs least squares polynomial fit.	493
Regression	Performs multiple linear fit with or without transformations.	499
Stat 1	Finds the mean, variance and standard deviation.	504
Stat 2	Computes various stat. measures for a variable.	509
T-Distribution	Calculates normal and T-distributions.	515
Unpaired	Compares 2 groups of unpaired data.	521
Variance 1	Performs one way analysis of variances.	524
Variance 2	Analyzes a variance table of one way random design.	528
XY	Plots functions of X and Y.	531

APPENDIX A - BASIC STATEMENT DEFINITIONS 535

TABLE OF CONTENTS

VOLUME THREE

Part 5 - Advanced Business Programs

NAME	DESCRIPTION
Billing	Performs posting and billing of accounts.
Inventory	Maintains data for inventory records.
Payroll	Computes payrolls with full set of deductions.
Risk	Performs a risk analysis on capital investments.
Schedule 2	Performs the most effi. scheduling of men or resources to loca.
Shipping	Solves the problem of scheduling and assignments.
Stocks	Computes the value of stocks.
Switch	Calculates the effects of a bond switch.

TABLE OF CONTENTS

VOLUME FOUR

General Purpose Programs

NAME	DESCRIPTION
Bingo	An age old favorite. "B9, C23, D4, E13, F21, BINGO!
Bonds	Computes the yields for a bond for different periods.
Bull	If you ever dreamed of being a Matador, here's your chance.
Enterprise	Take charge of the Enterprise while Capt. Kirk is on leave.
Football	Authentic NFL version of this well known sport.
Funds 1	Calculates long-term predictions of funds.
Funds 2	Plots the results of Funds 1.
Go-Moku	Ancient Chinese game of chance.
Jack	Plays Blackjack, Las Vegas style.
Life	Life is truly a battle for survival, a real challenger!
Loans	Calculates annuities, loans and mortgages.
Mazes	Generates unique maze puzzles for you to solve.
Poker	Five card draw - for up to 5 players.
Popul	Performs population projections for defined areas.
Profits	Determines the profitability of a firms various depts.
Qubic	3-Dimensional Tic-Tac-Toe.
Rates	Calc. the effective annual interest rate for stated interest.
Retire	Calculates your Civil Service Retirement benefits.
Savings	Computes savings plan profiles.
SBA	Calculates repayment schedules for SBA loans.
Tic-Tac-Toe	An all time favorite for young and old alike.

TABLE OF CONTENTS

VOLUME FIVE

Experimenter's Programs

NAME	DESCRIPTION
Andy Cap	Draws this famous cartoon character.
Baseball	Plays a full 9 innings of baseball.
Compare	Compares two groups of data.
Confid 10	Determines the confidence limits for a normal population.
Descrip	Provides a description of uni-variant data.
Differ	Computes the diff. of the means for data of equal variance.
Engine	Calculates the otto cycle of engines.
Fourier	This program evaluates fourier series.
Horse	Draws a picture of a horse.
Integers	Computes integers as the sum of other integers.
Logic	Determines conclusions from logic statements.
Playboy	Draws the playboy symbol.
Primes	Factors numbers into their primes.
Probal	Calc. Chi-Sq. and probabilities from 2X2 data sets.

TABLE OF CONTENTS

VOLUME FIVE (CONT.)

Experimenter's Programs

NAME	DESCRIPTION
Quadrac	Solves quadratic equations
Red Baron	Draws a picture of the infamous Red Baron.
Regression 2	Calculates linear regressions.
Road Runner	"Beep! Beep!" Draws a picture of the Road Runner.
Roulette	Computerized "Wheel of Fortune", plays roulette.
Santa	Old Saint Nick appears as jolly as ever.
Stat 10	Calculates quantities for two groups of paired data.
Stat 11	Computes sample statistics.
Steel	Calculates steel beam capacities.
Top	Computes cost for surfacing a road or driveway, etc.
Vary	Performs an analysis of a vari. table; one-way random design.
Xmas	Generates a "SINGING" Christmas card.

APPENDIX B - STATEMENT CONVERSION ALGORITHMS

TABLE OF CONTENTS

VOLUME SIX

A Complete Business System

NAME	DESCRIPTION
Ledger	Maintains ALL Company accounts and generates ALL financial reports. Includes routines for: Pyrl, Inv, Depr, A/R, A/P, Balance Sheets and Profit & Loss statements, etc.

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 tot al 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 sys tems 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 pro grams 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

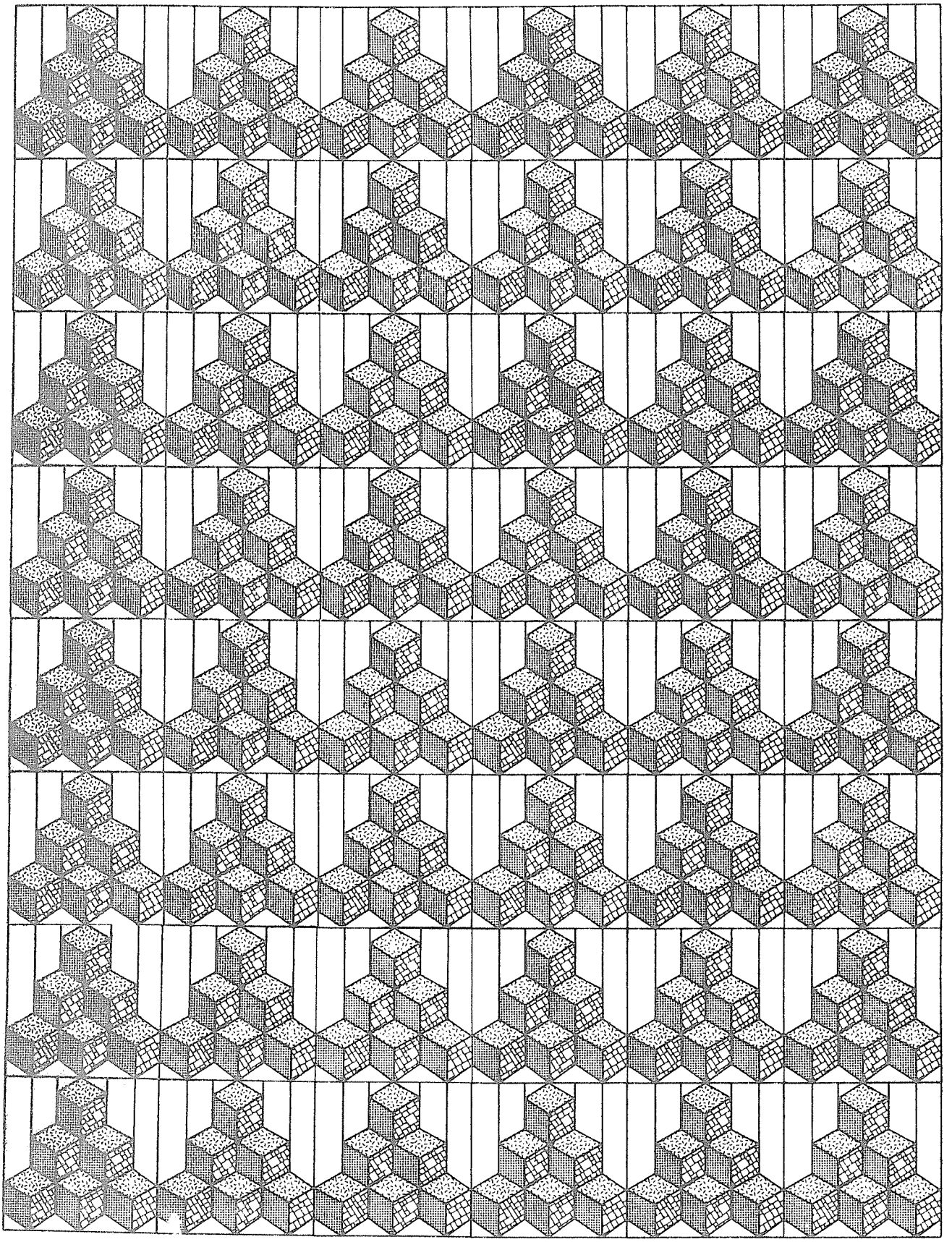
GE TIMESHARING

BILL JONES

GEORGE LONG

TOM ROSE

ARTWORK COURTESY OF MELISSA



PART 3

MATH

AND

ENGINEERING

BEAM:

DESCRIPTION

This program calculates the proper steel beam to be used for a particular application. The support may be subjected to the following conditions:

(L1.) Uniformly distributed load
(L2.) Midpoint load
(L3.) Two symmetric loads
(B1.) Support at both ends
(B2.) One end supported, other end fixed
(B3.) Both ends fixed
(B4.) Cantilever

USERS

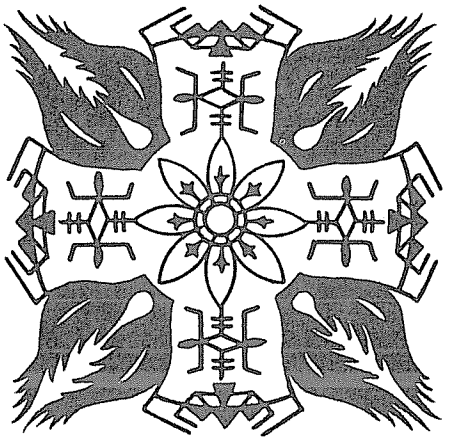
This program lends itself to construction type projects; such as building an addition to your house or garage, dams, bridges, transmission towers, etc.

INSTRUCTIONS

This program will ask if you want instructions when it is run. The output from the running of the program will be the standard nomenclature designation for the proper beam best suiting your needs. The weight of the beam is also taken into account when the calculations are performed.

LIMITATIONS

The program will store in 5K Bytes of memory. It should execute in any standard Basic speaking computer.



BEAM

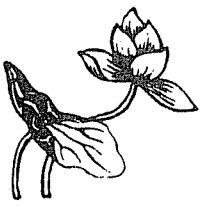
```
80 DIM Q(4,4), S(55), D(55), K(55), W(55)
90 PRINT "DO YOU WANT INSTRUCTIONS (1=YES, 0=NO)";
100 INPUT Y7
110 IF Y7=1 THEN 1670
120 GOTO 140
130 DATA 1,2,3,4,5,6,7,8,10,11,12,13,9,0,0,0
140 FOR I=1 TO 4
150 FOR J=1 TO 4
160 READ Q(I,J)
170 NEXT J
180 NEXT I
190 FOR I=1 TO 53
200 READ S(I),D(I),K(I),W(I)
210 NEXT I
220 PRINT
230 PRINT "WHAT ARE L,B,S,W,P,A";
240 INPUT C1,C2,L,W,P,A
250 LET C=Q(C1,C2)
260 IF C>0 THEN 310
270 PRINT
280 PRINT "NOT PROGRAMMED TO SOLVE THAT CASE.  FOR L=4,"
290 PRINT "CAN COMPUTE RECOMMENDED BEAM ONLY FOR B=1."
300 GOTO 1940
310 PRINT
320 IF C<3 THEN 440
330 IF C=3 THEN 460
340 IF C=4 THEN 480
350 IF C=5 THEN 500
360 IF C=6 THEN 520
370 IF C=7 THEN 540
380 IF C=8 THEN 560
390 IF C=9 THEN 580
400 IF C= 10 THEN 600
410 IF C=11 THEN 620
420 IF C=12 THEN 640
430 IF C=13 THEN 660
440 LET M=W*L*L/8
450 GOTO 670
460 LET M=W*L*L/12
470 GOTO 670
480 LET M=W*L*L/2
490 GOTO 670
500 LET M=P*L/4
510 GOTO 670
520 LET M=3*P*L/16
```

```
530 GOTO 670
540 LET M=P*L/8
550 GOTO 670
560 LET M=P*L/2
570 GOTO 670
580 LET M=P*A
590 GOTO 670
600 LET M=W*L*L/8+P*L/4
610 GOTO 670
620 LET M=W*L*L/8+3*P*L/16
630 GOTO 670
640 LET M=W*L*L/12+P*L/8
650 GOTO 670
660 LET M=W*L*L/2+P*L/2
670 LET S=12*M/20000
680 FOR I=1 TO 53
690 IF S<S(I) THEN 730
700 NEXT I
710 PRINT "LARGE ENOUGH BEAM DOES NOT EXIST--REDEFINE PROBLEM"
720 GOTO 1940
730 IF C<2 THEN 850
740 IF C=3 THEN 870
750 IF C=4 THEN 890
760 IF C=5 THEN 850
770 IF C=6 THEN 850
780 IF C=7 THEN 870
790 IF C=8 THEN 890
800 IF C=9 THEN 850
810 IF C=10 THEN 850
820 IF C=11 THEN 850
830 IF C=12 THEN 870
840 IF C=13 THEN 890
850 LET M1=M+W(I)*L*L/8
860 GOTO 900
870 LET M1=M+W(I)*L*L/12
880 GOTO 900
890 LET M1=M+W(I)*L*L/2
900 LET S2=12*M1/20000
910 IF S(I)>=S2 THEN 940
920 LET S=S2
930 GOTO 690
940 PRINT "   RECOMMENDED BEAM IS A  ";D(I);
950 IF K(I)=1 THEN 1000
960 IF K(I)=2 THEN 1020
970 IF K(I)=3 THEN 1040
980 IF K(I)=4 THEN 1060
990 IF K(I)=5 THEN 1080
1000 PRINT "U";
1010 GOTO 1090
1020 PRINT "JR";
1030 GOTO 1090
```

```
1040 PRINT "JRU";
1050 GOTO 1090
1060 PRINT "B";
1070 GOTO 1090
1080 PRINT "WF";
1090 PRINT W(I)
1100 PRINT "
              MORE DATA (1=YES,0=NO)";
1110 INPUT Y7
1120 IF Y7=0 THEN 1940
1130 GO TO220
1140 DATA 1.1,3,1,4.1
1150 DATA 1.2,3,1,5
1160 DATA 2.4,6,2,4.4
1170 DATA 3.5,7,2,5.5
1180 DATA 4.7,8,2,6.5
1190 DATA 6.5,10,3,8.4
1200 DATA 7.8,10,2,9
1210 DATA 9.3,12,3,10.6
1220 DATA 10.5,10,4,11.5
1230 DATA 12,12,2,11.8
1240 DATA 14.8,12,4,14
1250 DATA 17.5,12,4,16.5
1260 DATA 18.8,10,4,19
1270 DATA 21.4,12,4,19
1280 DATA 21.5,10,5,21
1290 DATA25.3,12,4,22
1300 DATA 26.4,10,5,25
1310 DATA 34.1,12,5,27
1320 DATA41.8,14,5,30
1330 DATA 48.5,14,5,34
1340 DATA 56.3,16,5,36
1350 DATA 64.4,16,5,40
1360 DATA 72.4,16,5,45
1370 DATA 80.7,16,5,50
1380 DATA 89,18,5,50
1390 DATA 98.2,18,5,55
1400 DATA 107.8,18,5,60
1410 DATA 126.4,21,5,62
1420 DATA 139.9,21,5,68
1430 DATA 150.7,21,5,73
1440 DATA 175.4,24,5,76
1450 DATA 196.3,24,5,84
1460 DATA 220.9,24,5,94
1470 DATA 242.8,27,5,94
1480 DATA 248.9,24,5,100
1490 DATA 266.3,27,5,102
1500 DATA 299.2,30,5,108
1510 DATA 327.9,30,5,116
1520 DATA 354.6,30,5,124
1530 DATA 404.8,33,5,130
```

```
1540 DATA 446.8,33,5,141
1550 DATA 502.9,36,5,150
1560 DATA 541,36,5,160
1570 DATA 579.1,36,5,170
1580 DATA 621.2,36,5,182
1590 DATA 663.6,36,5,194
1600 DATA 669.6,33,5,200
1610 DATA 740.6,33,5,220
1620 DATA 835.5,36,5,230
1630 DATA 892.5,36,5,245
1640 DATA 951.1,36,5,260
1650 DATA 1031.2,36,5,280
1660 DATA 1105.1,36,5,300
1670 PRINT
1680 PRINT "THIS PROGRAM WILL RECOMMEND THE CORRECT STEEL BEAM"
1690 PRINT "TO USE FOR A NUMBER OF COMMON APPLICATIONS.  TO USE,"
1700 PRINT "RESPOND TO THE QUESTION 'WHAT ARE L,B,S,W,P,A?'"
1710 PRINT "ACCORDING TO THE FOLLOWING CODE:"
1720 PRINT
1730 PRINT "    L = 1 FOR UNIFORMLY DISTRIBUTED LOAD"
1740 PRINT "      = 2 FOR SINGLE MIDPOINT LOAD"
1750 PRINT "      = 3 FOR UNIFORM LOAD + SINGLE MIDPOINT LOAD"
1760 PRINT "      = 4 FOR TWO EQUAL SYMMETRIC LOADS"
1770 PRINT
1780 PRINT "    B = 1 FOR BEAM SUPPORTED AT BOTH ENDS"
1790 PRINT "      = 2 FOR ONE END FIXED, OTHER END SUPPORTED"
1800 PRINT "      = 3 FOR BEAM FIXED AT BOTH ENDS"
1810 PRINT "      = 4 FOR ONE END FIXED (CANTILEVER)"
1820 PRINT
1830 PRINT "    S = LENGTH OF THE SPAN IN FEET"
1840 PRINT "    W = DISTRIBUTED LOAD IN POUNDS PER FOOT"
1850 PRINT "          (SET = 0 IF NOT APPLICABLE)"
1860 PRINT "    P = EACH CONCENTRATED LOAD IN POUNDS"
1870 PRINT "          (SET = 0 IF NOT APPLICABLE)"
1880 PRINT "    A = LOCATION OF LOAD(S) IN FEET FROM END"
1890 PRINT "          (SET = 0 IF NOT APPLICABLE)"
1900 PRINT
1910 PRINT
1920 GOTO 120
1930 DATA 1,1,1,1,1,1
1940 END
```

```
BEAM
RUN

WHAT ARE L,B,S,W,P,A ?1,1,23,1000,0,0
   RECOMMENDED BEAM IS A      14 WF      30
MORE DATA (1=YES,0=NO) ?1
WHAT ARE L,B,S,W,P,A ?1,2,23,100,0,0
   RECOMMENDED BEAM IS A       8  JR    6.5
MORE DATA (1=YES,0=NO) ?1
WHAT ARE L,B,S,W,P,A ?4,1,23,0,10000,9
   RECOMMENDED BEAM IS A      16  WF    36
MORE DATA (1=YES,0=NO) ?0
```



CONV:

DESCRIPTION

Conv calculates convolutions in the time domain. The calculations may be done for data points or for equations. The calculations are performed by integrating the function over the interval, with a very large number of increments (1000 or more). Due to this, the answer will only be approximate and the running time may be excessive.

USERS

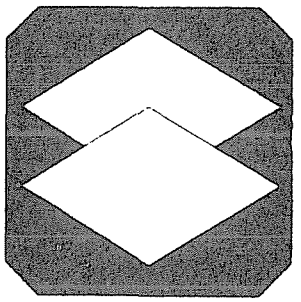
Mathematicians, Physicists, and Electrical Engineers will have use for this program. This program is a very powerful tool for determining impulse responses, antenna responses, and system transforms.

INSTRUCTIONS

The program will list all necessary instructions. If the convolution is for equations they must be entered between lines 1680 and 1700 before the program is run; upon running type a 1 to the first response and a Ø to the second response. If the input consists of data, type in the # of data points you have to the first response and then enter your data points at the second response.

LIMITATIONS

Lines 1080 and 1090 contain MAT = ZER statements. A3 in line 1370 may be made larger for greater accuracy, however this will increase the running time. I(MAX) * I(INCREMENT) $\geq$ 1/2 for proper program operation. The source code is 3K Bytes long and execution space requirements will be a function of the DIM statement on line 1020



CONV

```
1020 DIM X(4100),W(4100)
1030 REM THIS PROGRAM WILL DO CONVOLUTIONS IN THE TIME DOMAIN
1040 REM ON DATA POINTS OR EQUATIONS.
1050 PRINT "THIS PROGRAM WILL CONVOLVE TIME DOMAIN DATA OR EQ."
1060 PRINT
1070 D7=0
1080 MAT X=ZER
1090 MAT W=ZER
1100 X(0)=1
1110 PRINT "IF THE CALCULATIONS ARE FOR DATA TYPE A (1), ELSE (2)."
1120 INPUT D
1130 IF D=1 GOTO 1200
1140 PRINT "ENTER YOUR EQ.'S BETWEEN LINES 1680 AND 1700 BEFORE"
1150 PRINT "EXECUTING THIS PROGRAM. LET Y1=(EQ.1 * EQ.2)"
1160 PRINT "WHERE EQ.1=FUN(T) & EQ.2=FUN(I-T)."
1170 PRINT "UPON RERUNNING TYPE A (1) TO THE FIRST QUESTION."
1180 PRINT "AND ENTER A (0) FOR THE NUMBER OF DATA POINTS."
1190 GOTO 1830
1200 PRINT
1210 E=2.718282
1220 PRINT "  HOW MANY DATA POINTS DO YOU HAVE? "
1230 INPUT N
1240 IF N=0 GOTO 1360
1250 PRINT
1260 PRINT "INPUT YOUR X DATA"
1270 FOR I=1 TO N
1280 INPUT X(I)
1290 NEXT I
1300 PRINT
1310 PRINT "ENTER YOUR W DATA."
1320 FOR I=1 TO N
1330 INPUT W(I)
1340 NEXT I
1350 GOTO 1370
1360 D7=1
1370 A3=2000
1380 B=1
1390 IF D7=0 GOTO 1430
1400 REM A3 IS THE INCREMENT STEP AND SHOULD BE VERY LARGE
1410 PRINT "INPUT I(MAX) AND THE I(INCREMENT) DESIRED."
1420 INPUT N,B
1430 D2=Y2=0
1440 PRINT
```

```
1450 PRINT
1460 PRINT "   I   "," CONVOLUTION "
1470 PRINT "       "," VALUE "
1480 PRINT
1490 A=B
1500 FOR I7=A TO N STEP B
1510 IF D7=1 GOTO 1570
1520 FOR K=1 TO (I7*A3)
1530 I9=-K/I7
1540 REM THIS LOOP IS FOR GENERATING X() & W() SERIES IF DESIRED.
1550 NEXT K
1560 IF D7=0 GOTO 1590
1570 C=I7/(A3*INT(I7+1.1))
1580 GOTO 1610
1590 C=1
1600 D2=-1
1610 REM IF THIS IS FOR DATA AND NOT AN EQ. LET A=C=1
1620 IF D7=0 GOTO 1660
1630 I=I7
1640 GOTO 1670
1650 REM IF N>2, X() & W() MAY HAVE TO BE REDIMENSIONED.
1660 I=I7*A3
1670 FOR T=0 TO I STEP C
1680 REM ENTER YOUR EQ.1 & EQ.2 HERE
1690 Y1=X(T)*W(I-T)
1700 REM LET YI=(EQ.1*EQ.2)
1710 Y2=Y2+Y1
1720 D2=D2+1
1730 NEXT T
1740 Y=Y2
1750 IF D7=0 GOTO 1780
1760 Y=(Y*I)/D2
1770 GOTO 1790
1780 Y=(Y*I)/(D2*I7)
1790 I2=I7
1800 PRINT I2,Y
1810 D2=Y2=0
1820 NEXT I7
1830 PRINT
1840 PRINT "IF YOU WOULD LIKE TO RUN MORE DATA TYPE A (1)."
1850 INPUT D
1860 IF D=1 GOTO 1060
1870 END
```

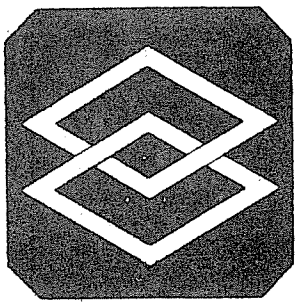
Example: Find the response of a circuit with an impulse response of e^{-t} to a forcing function of u(t), for $0 \leq t \leq \infty$.

and $y(t) = \int u(\tau)e(t-\tau)\, d\tau$

and $e(t-\tau) = e^{-(t-\tau)} = e^{(\tau-t)}$

```
LET    EQ.1 = U(T) =1   (THE UNIT STEP RESPONCE)
LET    EQ.2 = EXP(T-I)


AND LET

    I(MAX) =10
    I(MIN) = 0.05


1680  X(T)=1
1681  W(I-T)=EXP(-(I-T))
```

I	CONV VALUE
1	0.6321
2	0.8646
3	0.9502



FILTER:

DESCRIPTION

This program designs and calculates components for low pass filters. The filter is designed using constant K T sections and M - derived L sections. As many as 9 sections may be combined to give specific notch rejections a-above the rolloff frequency.

USERS

Audio enthusiasts can use this program to design special tone circuits. The ham radio operator who wishes to suppress harmonics from his transmitter is still another user. Anyone who designs AF or RF circuits will be able to benefit from use of this program.

INSTRUCTIONS

Before the program is run your data must be entered in lines 10 thru 29. The format is as follows:

```
     10 DATA  R,C,N,F(1),F(2),.................
```

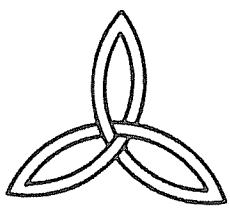
where

R - is the input and output impedance
C - is the rolloff frequency in Hertz
N - is the number of notch frequencies
F(1) - is the frequency for the first notch
F(2) - is the frequency for the second notch, etc.

Then type RUN. Additional instructions may be obtained by listing the program.

LIMITATIONS

This program should store and execute in 4K Bytes of memory.



FILTER

```
30 DIM L(10),F(10),S(10),C(10)
40 DIM M(10)
50 DATA 40E4,10E6,1,100
60 FOR I=0TO 10
70 LET F(I)=0
80 NEXTI
90 READ R,F(1),A
100 FOR I=2 TO A+1
110 READ F(I)
120 NEXT I
130 LET L=1000*R/(3.14159265*F(1))
140 LET C=1000000/(3.14159265*F(1)*R)
150 LET L(1)=.6*L/2
160 LET S(1)=.64*L/1.2
170 LET C(1)=.6*C/2
180 FOR I=2 TO 10
190 IF F(I)=0 THEN 250
200 LET M(I)=SQR(1-F(1)/F(I)*F(1)/F(I))
210 LET L(I)=M(I)*L/2
220 LET S(I)=L*((1-M(I)*M(I))/(4*M(I)))
230 LET C(I)=C*M(I)
240 NEXT I
250 PRINT
260 PRINT "DESIGN FOR DESIRED LOW PASS FILTER:"
270 PRINT
280 PRINT
290 GOSUB 660
300 GOSUB 620
310 GOSUB 640
320 GOSUB 620
330 PRINT">",L/2+L(1),"MH","                I"
340 GOSUB 620
350 PRINT "+---------------";C;"   MFD    ----------------------+"
360 GOSUB 620
370 LET I=2
380 IF F(I)=0 THEN 560
390 PRINT">",L/2+L(I),"MH","                I"
400 GOSUB 620
410 PRINT"+---------";S(2);"MH +",C(2),"MFD ------------+"
420 GOSUB 620
430 FOR I=3 TO 10
440 IF F(I)=0 THEN 500
450 PRINT">",L(I)+L(I-1),"MH","                  I"
460 GOSUB 620
470 PRINT"+---------";S(I);"MH +",C(I),"MFD ----------+"
```

```
480 GOSUB 620
490 NEXT I
500 PRINT">";L(I-1)+L(1);"MH";"                  I"
510 GOSUB 620
520 GOSUB 640
530 GOSUB 620
540 GOSUB 660
550 GOTO 680
560 PRINT">";L/2+L(1);"MH";"                  I"
570 GOSUB 620
580 GOSUB 640
590 GOSUB 620
600 GOSUB 660
610 GOTO 680
620 PRINT"I";" ";" ";"                  I"
630 RETURN
640 PRINT"+--------";S(1);"MH +";C(1);"MFD ----------+"
650 RETURN
660 PRINT "0<--------- ";R;"         OHM LINE";"                  --------->0"
670 RETURN
680 PRINT
690 PRINT
700 PRINT "TERMINATING SECTIONS GIVE MAXIMUM ATTENUATION AT";1.25*F(1)
710 IF F(2)<>0 THEN 740
720 PRINT "CYCLES PER SECOND."
730 GOTO 750
740 PRINT "CPS IN ADDITION TO THE SPECIFIED ATTENUATOR FREQUENCIES."
750 STOP
760 PRINT
770 PRINT "THIS PROGRAM DESIGNS LOW PASS FILTERS USING CONSTANT K"
780 PRINT "PROTOTYPE T SECTION AND M DERIVED (M=0.6) TERMINATION"
790 PRINT "L SECTIONS.  UP TO NINE ADDITIONAL M DERIVED T SECTIONS"
800 PRINT "MAY BE INCLUDED TO GIVE HIGH ATTENUATION AT SPECIFIED"
810 PRINT "FREQUENCIES IN THE STOP BAND.  TO USE, ENTER DATA AS:"
820 PRINT
830 PRINT "    10 DATA  R, C, N, F(1), F(2),..., F(N)"
840 PRINT
850 PRINT "WHERE R = DESIRED CHARACTERISTIC IMPEDANCE IN OHMS"
860 PRINT "       C = DESIRED CUTOFF FREQUENCY IN CYCLES/SECOND"
870 PRINT "       N = NUMBER OF ATTENUATORS DESIRED IN STOP BAND"
880 PRINT "    F(I) = FREQUENCY FOR ATTENUATOR I"
890 PRINT " TYPICAL INPUT FOR ONE  ATTENUATOR:500,2000,1,2050"
891 PRINT " INPUT TAKEN FROM: THEORY OF NETWORKS & LINES, "
892 PRINT "BY POTTER & FICH  P.P 111"
900 PRINT "THEN TYPE 'RUN'."
910 DATA 0
920 END
```

```
FILTER


10 DATA 50,2E4,2,455000,91E3

RUN


DESIGN FOR DESIRED LOW PASS FILTER:

0-----------          50          OHM LINE     -------------0
I                                                           I
+----------  0.4244132 MH  +   0.095493  MFD    ------------+
I                                                           I
>                 0.6366198  MH                            I
I                                                           I
+---------------  0.3183099   MFD    -----------------------+
I                                                           I
>                 0.7953901  MH                            I
I                                                           I
+--------     0.0003848  MH  +  0.3180022    MFD   ---------+
I                                                           I
>                 0.7856615   MH                            I
I                                                           I
+-------      0.0098505   MH   +   0.310527   MFD  ---------+
I                                                           I
>                 0.6268912  MH                            I
I                                                           I
+-------      0.4244132  MH   +    0.095493   MFD  ---------+
I                                                           I
0-----------          50           OHM LINE     -------------0


TERMINATING SECTIONS GIVE MAXIMUM ATTENUATION AT    25000
CPS IN ADDITION TO THE SPECIFIED ATTENUATOR FREQUENCIES.
```

FIT:

DESCRIPTION

This program performs interpolations using spline fits. The interpolation may be on data points, or integration limits, or anything else that needs to be interpolated.

USERS

This program is oriented toward individuals who analyze data or do interpolations for other reasons.

INSTRUCTIONS

Enter your data before the program is run; in the following format:

```
     1900  DATA    X,I
     1901  DATA    X(1),Y(1),X(2),Y(2),.....
     19__ DATA     ..........................
     1990  DATA    I(1),I(2),................
```

where

X = number of X and Y data points
Y = number of interpolations wanted

and

X(1) = first X point
Y(1) = corresponding Y point
I(1) = first point where interpolation is desired

After the data has been entered type RUN.

LIMITATIONS

Line 15 contains a Restore statement. The source code is 3K Bytes long and execution space is a function of the size of the DIM statements in lines 1100, 1110, 1121, 1122 and 1123. The program is set for 50 data sets in the DIM statements; this will require about 8K Bytes just for storage of the variable tables. With the DIM set at 50, the program will require 11K Bytes of memory for execution. The sample problem designs a low filter with 50ohm impedance, 20 KHz rolloff, and 2 antenuators in the stop band; 455 KHz and 91 KHz.

FIT

```
10 READ Q3
15 RESTORE
20 IF Q3 = 6E36 THEN 30
25 GOTO 1100
30 PRINT "DO YOU NEED INSTRUCTIONS? TYPE 1 FOR YES, 0 FOR NO.";
35 INPUT U6
40 IF U6 = 1 THEN 1999
45 READ Q3
50 RESTORE
55 IF Q3 = 6E36 THEN 2075
1100 DIM X(50),Y(50),S(50),A(50),B(50),C(50),F(50),G(50)
1110 DIM R(50),E(50),O(50)
1121 DIM Z(50)
1122 DIMT(50)
1123 DIM W(50)
1130 READ N,M
1140 FOR I=1 TO N
1150 READ X(I),Y(I)
1160 NEXT I
1170 FOR I = 1 TO M
1180 READ Z(I)
1190 NEXT I
1200 FOR I =2 TO N
1210 LET S(I)=X(I)-X(I-1)
1220 NEXT I
1230 FOR I = 2 TO N-1
1240 LET A(I)=S(I)/6
1250 LET B(I)=(S(I)+S(I+1))/3
1260 LET C(I)=S(I+1)/6
1270 LET F(I)=(Y(I+1)-Y(I))/S(I+1) - (Y(I)-Y(I-1))/S(I)
1280 NEXT I
1290 LET B(1)=1.0
1300 LET B(N)=1.0
1310 LET C(1)=-.5
1320 LET F(1)=0.
1330 LET F(N)=0.
1340 LET W(1)=B(1)
1350 LET R(1)=C(1)/W(1)
1360 LET G(1)=0.
1370 FOR I = 2 TO N
1380 LET W(I)=B(I)-A(I)*R(I-1)
1390 LET R(I)=C(I)/W(I)
1400 LET G(I)=(F(I)-A(I)*G(I-1))/W(I)
1410 NEXT I
```

```
1420 LET E(N)=G(N)
1430 FOR I = 2 TO N
1440 LET K=N+1-I
1450 LET  E(K)=G(K)-R(K)*E(K+1)
1460 NEXT I
1470 FOR I = 1 TO M
1480 LET K = 2
1490 IF Z(I)-X(1)>0 THEN 1600
1500 IF Z(I)<X(1) THEN 1530
1510 LET T(I)=Y(1)
1520 GOTO 1700
1530 IF Z(I)>= (1.1*X(1)-.1*X(2)) THEN 1670
1540 PRINT "OUT OF RANGE  Z= "Z(I)
1550 GOTO 1670
1560 IF Z(I)<=(1.1*X(N)-.1*X(N-1)) THEN 1580
1570 PRINT "OUT OF RANGE  Z= "Z(I)
1580 LET K = N
1590 GOTO 1670
1600 IF Z(I)>X(K) THEN 1640
1610 IF Z(I)<X(K) THEN 1670
1620 LET T(I)=Y(K)
1630 GOTO 1700
1640 LET K = K + 1
1650 IF K<= N THEN 1600
1660 GOTO 1560
1670 LET T(I)=E(K-1)*(X(K)-Z(I))↑3/6/S(K)+E(K)*(Z(I)-X(K-1))↑3/6/S(K)
1680 LET T(I)=T(I)+(Y(K)/S(K)-E(K)*S(K)/6)*(Z(I)-X(K-1))
1690 LET T(I)=T(I)+(Y(K-1)/S(K)-E(K-1)*S(K)/6)*(X(K)-Z(I))
1700 NEXT I
1710 PRINT "NO OF POINTS GIVEN = "N
1720 PRINT "NO OF INTERPOLATED POINTS = "M
1730 PRINT
1740 PRINT "INTERP X","INTERP Y"
1750 PRINT
1760 FOR I = 1 TO M
1770 PRINT Z(I),T(I)
1780 NEXT I
1790 PRINT
1800 PRINT "ORIGINAL X","ORIGINAL Y"
1810 PRINT
1820 FOR I = 1 TO N
1830 PRINT X(I),Y(I)
1840 NEXT I
1899 STOP
1900 DATA 6E36
1999 PRINT
2000 PRINT "   THIS PROGRAM COMPUTES SPLINE INTERPOLATION."
2010 PRINT "STARTING IN LINE 1900 THE FIRST DATA STATEMENT"
2020 PRINT "MUST BE THE NUMBER OF PAIRS OF X AND Y, FOLLOWED"
2022 PRINT "BY THE NUMBER OF INTERPOLATION POINTS DESIRED."
2030 PRINT "THE FOLLOWING DATA STATEMENTS MUST CONTAIN THE"
```

```
2040 PRINT "VALUES OF X AND Y, FOLLOWED BY THE X VALUES AT"
2042 PRINT "WHICH INTERPOLATIONS ARE TO TAKE PLACE."
2050 PRINT "ENTER DATA ACCORDING TO INSTRUCTIONS."
2060 PRINT "AFTER DATA IS ENTERED TYPE   RUN   "
2070 STOP
2075 PRINT
2080 PRINT "PLEASE ENTER DATA AND THEN TYPE    RUN   "
9999 END
```

```
FIT
```

```
1900 DATA 11,3
1901 DATA 0,0
1902 DATA .1,.0398,.2,.0793,.3,.1179,.4,.1554,.5,.1915
1903 DATA .6,.2257,.7,.2580,.8,.2881,.9,.3159,1,.3413
1904 DATA .23,.57,.65
```

```
RUN

NO. OF POINTS GIVEN =   11
NO. OF INTERPOLATED POINTS =   3

INTERP  X       INTERP  Y

      .23         .0909903
      .57         .2156386
      .65         .2421052
```

INTEGRATION 1:

DESCRIPTION

This math program evaluates definite integrals. The integrals are calculated using the Gaussion quadrature techniques. For this, ten values of the integrand are used. This program is only used to evaluate discrete equations; for data sets see Integration 2.

USERS

This program will find many users among all types of engineers, physicists, and mathematicians. The program is useful anywhere anyone needs integration.

INSTRUCTIONS

Enter your function in line 100. The format is as follows:

```
100  LET Y = f(x)
```

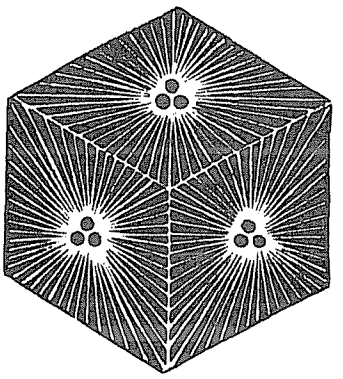
Then type RUN. The program will ask for L, U, and N, where:

L = lower limit
U = upper limit
N = number of intervals to take

To stop the program type Ø,Ø,1 to the L,U,N question.

LIMITATIONS

This program will require about 2K Bytes for storage and 3K Bytes of memory for execution.



INTEGRATION 1

```
2 DIM U(10),R(10)
3 LET Y=1E44
4 FOR I=1 TO 5
5 READ U(I),R(I)
6 NEXT I
7 LET X=LOG(99)
10 LET X=X
240 IF Y=1E44 THEN 690
250 IF J=1 THEN 510
260 IF J=2 THEN 550
270 IF N0 = 1 THEN 380
280 PRINT
290 PRINT "L, U, N = ";
300 INPUT A0, B0, N0
310 IF A0 = B0 THEN 860
320 IF N0 < 1  THEN  780
330 LET A = A0
340 IF N0 = 1  THEN  410
350 LET Z0 = 0
360 LET D = (B0-A0)/N0
370 LET B = A
380 LET B = B + D
390 IF B <= B0  THEN  420
400 IF Z0 > 0  THEN  620
410 LET B = B0
420 LET C1 = .5*(B+A)
430 LET C2=B-A
440 LET S=0
450 LET I=0
460 LET I=I+1
470 LET W=C2*U(I)
480 LET X=C1+W
490 LET J=1
500 GO TO 10
510 LET S=S+R(I)*Y
520 LET X=C1-W
530 LET J=2
540 GO TO 10
550 LET S=S+R(I)*Y
560 IF I<5 THEN 460
570 IF N0 = 1  THEN  610
580 LET Z0 = Z0 + S*C2
590 LET A = B
```

```
600 GOTO 380
610 LET Z0 = S*C2
620 PRINT
630 PRINT "     THE INTEGRAL FROM "; A0; " TO "; B0; " = "; Z0
640 PRINT
650 GOTO 280
660 DATA 744371695E-10,147762112E-9,216697697E-9,134633360E-9
670 DATA 339704784E-9,109543181E-9,432531683E-9,747256746E-10
680 DATA 486953264E-9,333356722E-10
690 PRINT
700 PRINT "THIS PROGRAM EVALUATES DEFINITE INTEGRALS USING GAUSS' RULE ";
710 PRINT "WITH 10 PTS."
720 PRINT "TO USE, ENTER THE FUNCTION TO BE INTEGRATED AS :"
730 PRINT
740 PRINT "    100  LET Y = < A FUNCTION OF X >"
750 PRINT
760 PRINT "YOU CAN USE LINE 10-129 TO EXPRESS THE FUNCTION."
770 PRINT "THEN TYPE 'RUN' ."
780 PRINT
790 PRINT "AS IT RUNS, THE PROGRAM ASKS FOR :"
800 PRINT "LOWER LIMIT (L),  UPPER LIMIT (U),  # OF INTERVALS (N) ."
810 PRINT "[ONE INTERVAL IS USUALLY GOOD ENOUGH FOR A SMOOTH FUNCTION.]"
820 PRINT "IF MORE THAN ONE INTEGRAL IS TO BE EVALUATED,";
830 PRINT "JUST KEEP SUPPLYING LIMITS."
840 PRINT "( 'P' ALREADY HAS THE VALUE  PI, IN CASE YOU NEED IT.)"
850 PRINT "TO STOP THE PROGRAM, MAKE THE TWO LIMITS EQUAL."
860 END
```

```
INTEGRATION 1



100 LET Y=X**2/(LOG(3*X)/2.302585093)

RUN

L, U, N = ?0,7.65,1

      THE INTEGRAL FROM      0  TO  7.65  =  125.5347

L, U, N = ?7.65, 1567,1

      THE INTEGRAL FROM     7.65  TO  1567  =  3.64278 E08

L, U, N = ?0,0,1
```

INTEGRATION 2:

DESCRIPTION

This integration program performs integration by spline fits. The function may be defined by unequally spaced data points instead of an equation, as in Integration 1. Integration 2 is used to integrate a function described by data points and not an equation as in the previous program.

USERS

This program would have more appeal and use to individuals who gather data then would Integration 1.

INSTRUCTIONS

Your data must be entered before the program is run. The format for the data entry is as follows:

```
700  DATA    X
701  DATA    X(1),Y(1),X(2),Y(2),.....
```

where

X - is the number of data sets to be used
X( ) and Y( ) are the corresponding data points

Then type RUN.

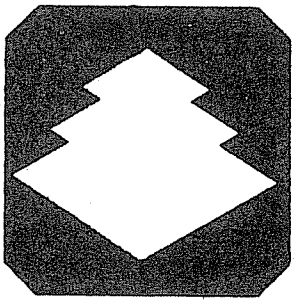
LIMITATIONS

Line 19 contains a Restore statement. The source code will store in 3K Bytes of memory. The execution space is about 11K Bytes when set for 50 data sets. The number of data sets the program will handle is set in the DIM statements on lines 170, 180, and 190.

INTEGRATION 2

```
6 REM:          [DO NOT RESEQUENCE]
7 REM:    INTEGRATION OF A FUNCTION FROM DATA POINTS, BY SPLINE FITS.
8 REM:    (POSSIBLY  UNEQUALLY SPACED POINTS)
9 REM:               ----------
10REM:    FROM "MATHEMATICAL METHODS FOR DIGITAL COMPUTERS", VOL.II,
11REM:    RALSTON & WILF, 1967,  PAGES 156 - 168
18 READ Z
19 RESTORE
20 IF Z = 6E36 THEN 25
21 GO TO 170
25 PRINT "INSTRUCTIONS ?  (TYPE 1 FOR YES,  0 FOR NO)"
30 INPUT Z8
40 IF Z8 = 1 THEN 810
50 READ Z
60 RESTORE
70 IF Z = 6E36 THEN 950
170 DIM X(50),Y(50),S(50),A(50),B(50)
180 DIM C(50),F(50),R(50),G(50),E(50)
190 DIM W(50),O(50),T(50)
200 READ N
210 FOR I=1 TO N
220 READ X(I),Y(I)
230 LET O(I)=0
240 LET E(I)=0
250 NEXT I
260 FOR I =2 TO N
270 LET S(I)=X(I)-X(I-1)
280 NEXT I
290 FOR I = 2 TO N-1
300 LET A(I)=S(I)/6
310 LET B(I)=(S(I)+S(I+1))/3
320 LET C(I)=S(I+1)/6
330 LET F(I)=(Y(I+1)-Y(I))/S(I+1) - (Y(I)-Y(I-1))/S(I)
340 NEXT I
350 LET A(N)=-.5
360 LET B(1)=1.
380 LET C(1)=-.5
390 LET F(1)=0.
400 LET F(N)=0.
410 LET W(1)=B(1)
420 LET R(1)=C(1)/W(1)
430 LET G(1)=0.
440 FOR I=2 TO N
```

```
450 LET W(I)=B(I)-A(I)*R(I-1)
460 LET R(I)=C(I)/W(I)
470 LET G(I)=(F(I)-A(I)*G(I-1))/W(I)
480 NEXT I
490 LET E(N)=G(N)
500 FOR I=2 TO N
510 LET K=N+1-I
520 LET E(K)=G(K)-R(K)*E(K+1)
530 NEXT I
540 LET T(1)=0.0
550 FOR I =2 TO N
560 LET T(I)=T(I-1)+S(I)*(Y(I)+Y(I-1))/2. -S(I)↑3*(E(I)+E(I-1))/24.
570 NEXT I
580 PRINT
590 PRINT "    X","   Y","INTEGRAL"
600 PRINT
610 FOR I=1 TO N
620 PRINT X(I),Y(I),T(I)
630 NEXT I
650 STOP
700 DATA 6E36
810 PRINT
815 PRINT "INTEGRATION OF A FUNCTION BY SPLINE FITS, WITH THE FUNCTION"
820 PRINT "DEFINED BY (POSSIBLY) UNEQUALLY SPACED DATA POINTS."
825 PRINT "                              ---------"
830 PRINT " STARTING WITH STATEMENT NUMBER 700 THE FIRST DATA STATEMENT"
840 PRINT " MUST BE THE NUMBER OF PAIRS OF X AND Y."
850 PRINT " THE FOLLOWING DATA STATEMENTS MUST CONTAIN THE VALUES"
860 PRINT " OF X AND Y."
900 PRINT "ENTER DATA ACCORDING TO INSTRUCTIONS."
910 PRINT "AFTER DATA IS ENTERED TYPE  'RUN'    "
920 STOP
950 PRINT " PLEASE ENTER DATA AND THEN TYPE  RUN "
9999 END
```



INTEGRATION 2

```
700 DATA 11
701 DATA 0,0
702 DATA .1,.0398, .2,.0793,.3,.1179
703 DATA .4,.1554,.5,.1915,.6,.2257
704 DATA .7,.258,.8,.2881,.9,.3159
705 DATA 1,.3413
```

RUN

X	Y	INTEGRAL
0	0	0
.1	.0398	.0019911
.2	.0793	.0079511
.3	.1179	.0178199
.4	.1554	.0314948
.5	.1915	.0488539
.6	.2257	.06973
.7	.258	.0939314
.8	.2881	.1212573
.9	.3159	.1514695
1	.3413	.1843773

INTENSITY:

DESCRIPTION

This program will compute wave interactions. The wave may be acoustical - such as sonar or hydrosonics, RF - such as VHF television, or it may be optical - such as infared light. Given the array size and spacings in wavelengths of the wave the program will calculate and plot the directivity of the array, antenna, or transducer.

USERS

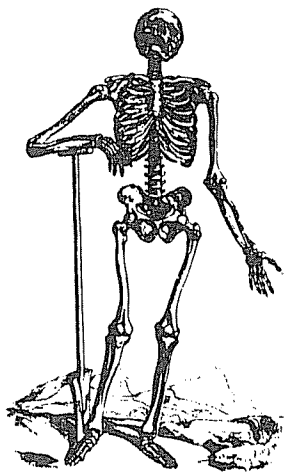
This program is useful to anyone who is interested in directive emitters. The marine engineer who is designing side scan sonar systems, the ham radio operator who wants to build a highly directional beam, or the engineer who is working with the new IR and optical military weapon systems.

INSTRUCTIONS

List the program for all necessary information. The plotting is in 9 degree increments from Ø to 90 degrees. If other limits or resolutions are desired modify lines 1750 and 1760.

LIMITATIONS

This program will store in 7K Bytes of memory and execute in 9K Bytes of memory. The plotting can be done on any standard teletype terminal.



```
10 PRINT "(ANTENNA) OR (WAVE) EQUATION PLOTTING PROGRAM        "
20 REM BY ROGER BROWN JANUARY 1975
30 PRINT "SIN(X) PEAKS AT (0) DEGREES"
40 PRINT
50 PRINT
60 PRINT "TYPE A (1) FOR SIN(X) DEPENDENCY"
70 PRINT "TYPE A (2) FOR COS(X) DEPENDENCY"
80 PRINT "TYPE A (3) FOR JUST (X) DEPENDENCY"
90 PRINT "TYPE A (1) FOR Z1=SIN(N*DEP.*S)/N*SIN(DEP.*S)"
100 PRINT "TYPE A (2) FOR Z2=SIN(DEP.*L)/(DEP.*L)"
110 PRINT "TYPE A (3) FOR Z3=Z1*Z2"
120 PRINT "TYPE A (0) FOR ABS. VALUE OF Y,(1) AMP., (2) INTEN."
130 PRINT "(DEP.) = (C*P*X) WHERE C IS SET=1/P, LINE 240"
140 PRINT
150 LET C1=0
160 IF C1=2 GO TO 180
170 INPUT A, B, D1
180 REM ##########################################
190 LET J=0
200 LET P7=0
210 IF C1=3 GO TO 330
220 REM (J) AND (P7) MUST BE MOVED FROM PLOT PROG. TO HERE TO WORK
230 REM X GOES FROM Q2 TO Q3
240 PRINT
250 PRINT
260 IF C1=1 GO TO 350
270 IF C1=2 GO TO 330
280 REM THE LOOP STARTS AT 230
290 PRINT "INPUT N THE NUMBER OF ELEMENTS"
300 PRINT "INPUT S THE SPACING IN WAVELENGTHS"
310 PRINT "INPUT L THE WIDTH OF THE ARRAY IN WAVELENGTHS"
320 PRINT "IF NOT KNOWN LET N=10, S=1, AND L=1"
330 PRINT
340 INPUT N, S, L
350 LET P=3.14159
360 LET C=1/P
370 REM THIS C IS USED IN LINES 280 AND 300 AND WHEN=1/P CANCELS
380 REM C VARIES THE PATTERN IF USED. IT IS IN WAVELENGTHS
390 LET X9=ABS(X)
400 IF X9=0 GO TO 470
410 IF X9=180 GO TO 470
```

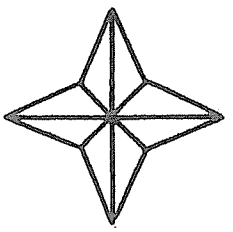
```
420 IF X9=360 GO TO 470
430 IF X9=540 GO TO 470
440 IF X9=720 GO TO 470
450 IF X9=900 GO TO 470
460 IF X9<>0 GO TO 480
470 LET X=X+.000001
480 LET X1=(X*P)/180
490 LET X2=(C*P*SIN(X1))
500 IF A=1 GO TO 550
510 LET X2=(C*P*COS(X1))
520 IF A=2 GO TO 550
530 LET X2=(C*P*(X1))
540 IF A<>3 GO TO 2350
550 LET N=N
560 REM N IS THE NUMBER OF ELEMENTS IN THE TRANS. ANTENNA
570 REM ALSO EACH ELEMENT IS TO BE DRIVEN
580 LET S=S
590 REM S IS THE ELEMENT SPACING IN WAVELENGTHS
600 LET Z1=SIN(N*S*X2)/(N*SIN(S*X2))
610 LET L=L
620 REM L IS THE WIDTH OF THE ARRAY IN WAVELENGTHS
630 LET Z2=SIN(L*X2)/(L*X2)
640 LET Z3=(Z1*Z2)
650 LET W=Z1
660 IF B=1 GO TO 730
670 LET W=Z2
680 IF B=2 GO TO 730
690 LET W=Z3
700 REM THIS IS TPLOT1
710 REM THIS IS A GENERAL PLOTTING PROGRAM IT WILL PLOT UP TO
720 REM 2 FUNCTIONS AS DESCRIBED IN LINES 2625 TO 2675
730 LET Y=ABS(W)
740 IF D1=0 GO TO 780
750 LET Y=W
760 IF D1=1 GO TO 780
770 LET Y=W*W
780 LET Z=X2
790 LET Y1=Z
800 REM ##### LET Z=01 IN LINE 2050 TO PUT A RIGHT BORDER ON PLOT
810 LET T=0
820 LET T1=0
830 IF J=0 THEN 880
840 RETURN
850 REM ######$$$$$$###### Q0 AND Q1 ARE LOWER AND UPPER LIMITS
860 REM ######$$$$$$###### FOR THE Y AXIS
870 REM ######$$$$$$###### Q2 AND Q3 ARE MIN. AND MAX. FOR X
880 LET Q0=-1
890 LET Q1=1.5
900 LET Q2=0
910 LET Q3=90
920 LET Q5=(Q1-Q0)/50
```

```
930 LET Q6=0
940 LET B0=10
950 LET Q4=(Q3-Q2)/B0
960 LET J=2
970 LET Q9=(Q3+Q4)
980 LET K=0
990 REM ********** X ONLY PLOTS TO Q9-Q4 FOR SOME REASON ?
1000 FOR X=Q2 TO Q9 STEP Q4
1010 REM ########################################
1020 REM THE SUB ROUTINE MUST GO TO THE START OF THE EQUATIONS
1030 GOSUB 350
1040 IF Y1<>0 GO TO 1060
1050 LET Y1=Q1
1060 IF Q6=0 THEN 1470
1070 IF Q6 =20 THEN 1200
1080 LET K=K+1
1090 IF K=5 GO TO 1140
1100 IF K=7 GO TO 1160
1110 IF K=6 GO TO 1180
1120 PRINT "                ";"I";
1130 GO TO 1220
1140 PRINT " DELTA X       ";"I";
1150 GOTO 1220
1160 PRINT "(";Q4;")";"       I";
1170 GO TO 1220
1180 PRINT "(DEGREES)      ";"I";
1190 GO TO 1220
1200 PRINT "                ";"X";
1210 LET Q6 = 10
1220 IF Y > Q1 THEN 1450
1230 IF Y1<Q1 GO TO 1250
1240 LET Y1=Q1
1250 IF Y < Q0 THEN 1450
1260 IF Y1<Q0 THEN 1450
1270 LET Q7 = Q0 + 2*Q5
1280 LET Z=Q7+(Q5/2)
1290 IF T>0 THEN 1830
1300 IF Z<Y THEN 1820
1310 IF T1>0 THEN 1340
1320 IF Z<Y1 THEN 1340
1330 GOTO 1940
1340 IF Z-Y>=2*Q5 THEN 1400
1350 IF Z-Y>=Q5 THEN 1380
1360 PRINT "  *";
1370 GOTO 1610
1380 PRINT " * ";
1390 GOTO 1610
1400 PRINT "*  ";
1410 GOTO 1610
1420 PRINT "   ";
1430 LET Q7= Q7+ 3*Q5
```

```
1440 GOTO 1280
1450 PRINT "OFF SCALE.  (X,Y,Z) =";X;",";Y;",";Y1
1460 GOTO 1630
1470 PRINT
1480 IF P7=99 THEN 1530
1490 PRINT
1500 REM Y IS PLOTTED '*' AND Z IS PLOTTED '+' AND  O IS COMMON
1510 PRINT
1520 PRINT
1530 PRINT "Y IS (*)"
1540 PRINT "Z=(DEP.)  (+)"
1550 PRINT "(O) IS COMMON DATA POINT"
1560 PRINT
1570 PRINT "    ";00,"    ( Y INCREMENT =";05;")";"                ";01
1580 PRINT
1590 PRINT 02,"I----+----I----+----I----+----I----+----I----+----I"
1600 GOTO 1200
1610 LET T= 1
1620 IF T1=0 THEN 1430
1630 LET 06=06+1
1640 IF INT(Z)>01 THEN 1660
1650 PRINT
1660 NEXT X
1670 LET P7=99
1680 PRINT 03,"I----+----I----+----I----+----I----+----I----+----I"
1690 PRINT
1700 PRINT
1710 PRINT
1720 PRINT
1730 PRINT "TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES"
1740 PRINT "TYPE (1) FOR NEW EQUATIONS ONLY"
1750 PRINT " TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN  0900, 0910)"
1760 PRINT "TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE  0940)"
1770 INPUT C1
1780 IF C1=1 GO TO 160
1790 IF C1=2 GO TO 160
1800 IF C1<>3 GO TO 2350
1810 GO TO 160
1820 IF T1>0 THEN 1420
1830 IF Z<Y1 THEN 1420
1840 IF Z-Y1>=2*05 THEN 1900
1850 IF Z-Y1>=05 THEN 1880
1860 PRINT "  +";
1870 GOTO 1910
1880 PRINT " + ";
1890 GOTO 1910
1900 PRINT "+  ";
1910 LET T1=1
1920 IF T>0 THEN 1630
1930 GOTO 1430
1940 IF Z-Y>=2*05 THEN 2120
```

```
1950 IF Z-Y>=05 THEN 2040
1960 IF Z-Y1>=2*05 THEN 2020
1970 IF Z-Y1>=05 THEN 2000
1980 PRINT "  0";
1990 GOTO 1630
2000 PRINT " +*";
2010 GOTO 1630
2020 PRINT "+ *";
2030 GOTO 1630
2040 IF Z-Y1>=2*05 THEN 2100
2050 IF Z-Y1>=05 THEN 2080
2060 PRINT " *+";
2070 GOTO 1630
2080 PRINT " 0";
2090 GOTO 1630
2100 PRINT "+*";
2110 GOTO 1630
2120 IF Z-Y1>=2*05 THEN 2180
2130 IF Z-Y1>=05 THEN 2160
2140 PRINT "* +";
2150 GOTO 1630
2160 PRINT "*+";
2170 GOTO 1630
2180 PRINT "0";
2190 GOTO 1630
2200 PRINT
2350 END
```

The first data entry prompted for by the program is the values for A, B and D1. The possible values for these variables are listed just prior to their input. A and B may be 1, 2 or 3 and D1 may be Ø, 1 or 2. The value of 1 is normally used for A, the equations have sin (X) dependency. Sin (X) dependent equations have a peak response at Ø degrees or on axis. B is usually 3, which yields normal antenna response. The particular equation Z1, Z2 or Z3 that satisfies your system response will ultimately determine the value of B for your problem. D1 is a bit simpler. If D1 is set equal to Ø, the graph plots the absolute value of the function. If equal to 1 the function itself is plotted, the function will have values on both sides of zero. If D1 is set equal to 2, then the intensity function is plotted.



Sample Run of Intensity

Example #1

Determining the effect on directivity by varying the N,S, and L ratios. Where

N = Total number of elements
S = Element spacing in wavelengths
L = Average array width in wavelengths

For this array we chose sin (X) dependency and Z3 = Z1 * Z2. Then plot the amplitude of the results between Ø degrees and 90 degrees. The plot is symmetrical from 270 degrees to Ø to 90 degrees. Where Ø degrees is the direction the array or antenna is pointing. The first plot is done for 9 degree increments as we are looking for the gross effects in this example. This plot is to be used as a reference which will be compared against the other examples. In this plot N = 10, S = 1 wavelength and L = 1 wavelength. It can be seen that the half power points are about 20 degrees wide, twice 10 degrees. The small curve represented by "+" represents the dependent function, sin (X) as set initially. This second curve is used to indicate proper operation.

For the second plot N is left at 10, S is changed to 5 and L is left at 1. This array now has half power points at about 10 degrees. This means that by taking the same array as above and only changing the spacing between the elements the directivity has been doubled. However the side bands have also been doubled, this is normally not a desireable effect.

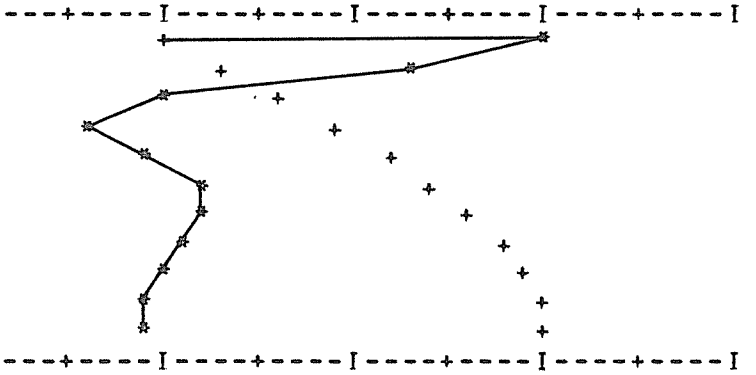
For the third plot N = 10, S is reset to 1 and L is widened to 5 wavelengths. Here the half power points are 20 degrees but the side lobes are considerably less then in the first plot.

For the fourth plot N has been increased to 50, S and L both set equal to 1 wavelength. Now the half power points for this combination is again 10 degrees but the side lobes are almost completely eliminated.

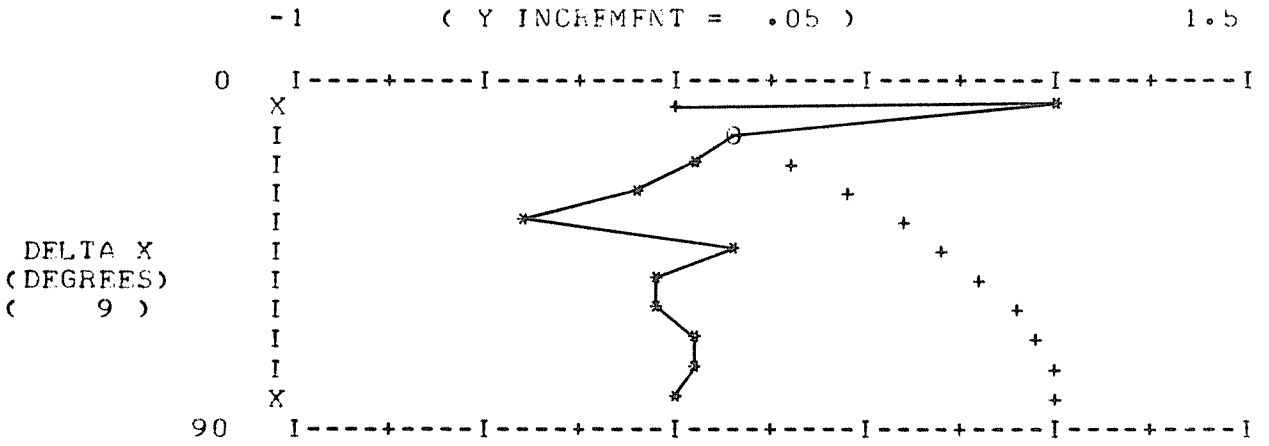
Example #2

This example examines the effects of infared light that is emitted from a #47 light bulb. The plot is contained on two pages. The outer envelope "+" represents the diffraction effects when focused by a perfect lense. The inner envelope "*" represents the intensity of the radiation very close to the axis of orientation.

```
RUN


(ANTENNA) OR (WAVE) EQUATION PLOTTING PROGRAM
SIN(X) PEAKS AT (0) DEGREES


TYPE A (1) FOR SIN(X) DEPENDENCY
TYPE A (2) FOR COS(X) DEPENDENCY
TYPE A (3) FOR JUST (X) DEPENDENCY
TYPE A (1) FOR Z1=SIN(N*DEP.*S)/N*SIN(DEP.*S)
TYPE A (2) FOR Z2=SIN(DEP.*L)/(DEP.*L)
TYPE A (3) FOR Z3=Z1*Z2
TYPE A (0) FOR ABS. VALUE OF Y,(1) AMP., (2) INTEN.
(DEP.) = (C*P*X) WHERE C IS SET=1/P, LINE 240

?1,3,1


INPUT N THE NUMBER OF ELEMENTS
INPUT S THE SPACING IN WAVELENGTHS
INPUT L THE WIDTH OF THE ARRAY IN WAVELENGTHS
IF NOT KNOWN LET N=10, S=1, AND L=1

?10,1,1        (Reference plot)



Y IS (*)
Z=(DEP.)  (+)
(O) IS COMMON DATA POINT

              -1        ( Y INCREMENT =  .05 )                    1.5

          0   I----+----I----+----I----+----I----+----I----+----I
              X
              I
              I
              I
              I
 DELTA X      I
(DEGREES)     I
(    9 )      I
              I
              I
              I
              X
         90   I----+----I----+----I----+----I----+----I----+----I
```

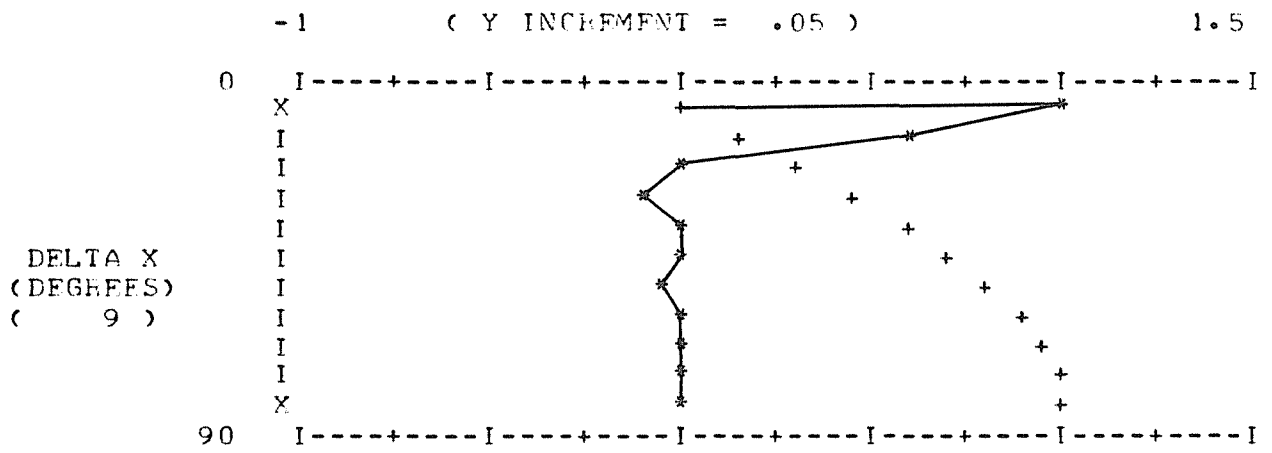


```
TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES
TYPE (1) FOR NEW EQUATIONS ONLY
 TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN 0900, 0910)
TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE 0940)
?2


?10,5,1


Y IS (*)
Z=(DEP.)  (+)
(O) IS COMMON DATA POINT
```



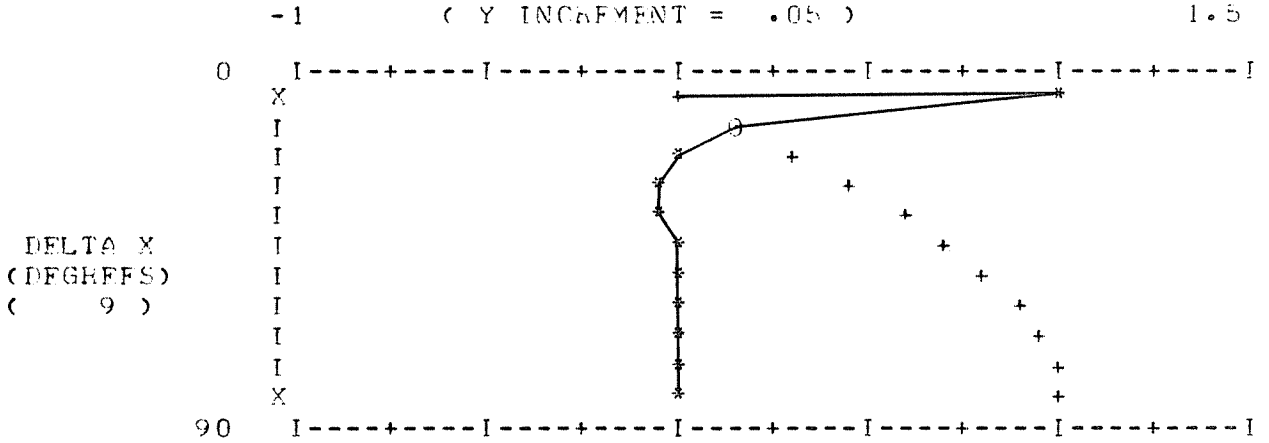

```
TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES
TYPE (1) FOR NEW EQUATIONS ONLY
 TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN 0900, 0910)
TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE 0940)
?2
```

```
?10,1,5

Y IS (*)
Z=(DEP.)  (+)
(O) IS COMMON DATA POINT
```



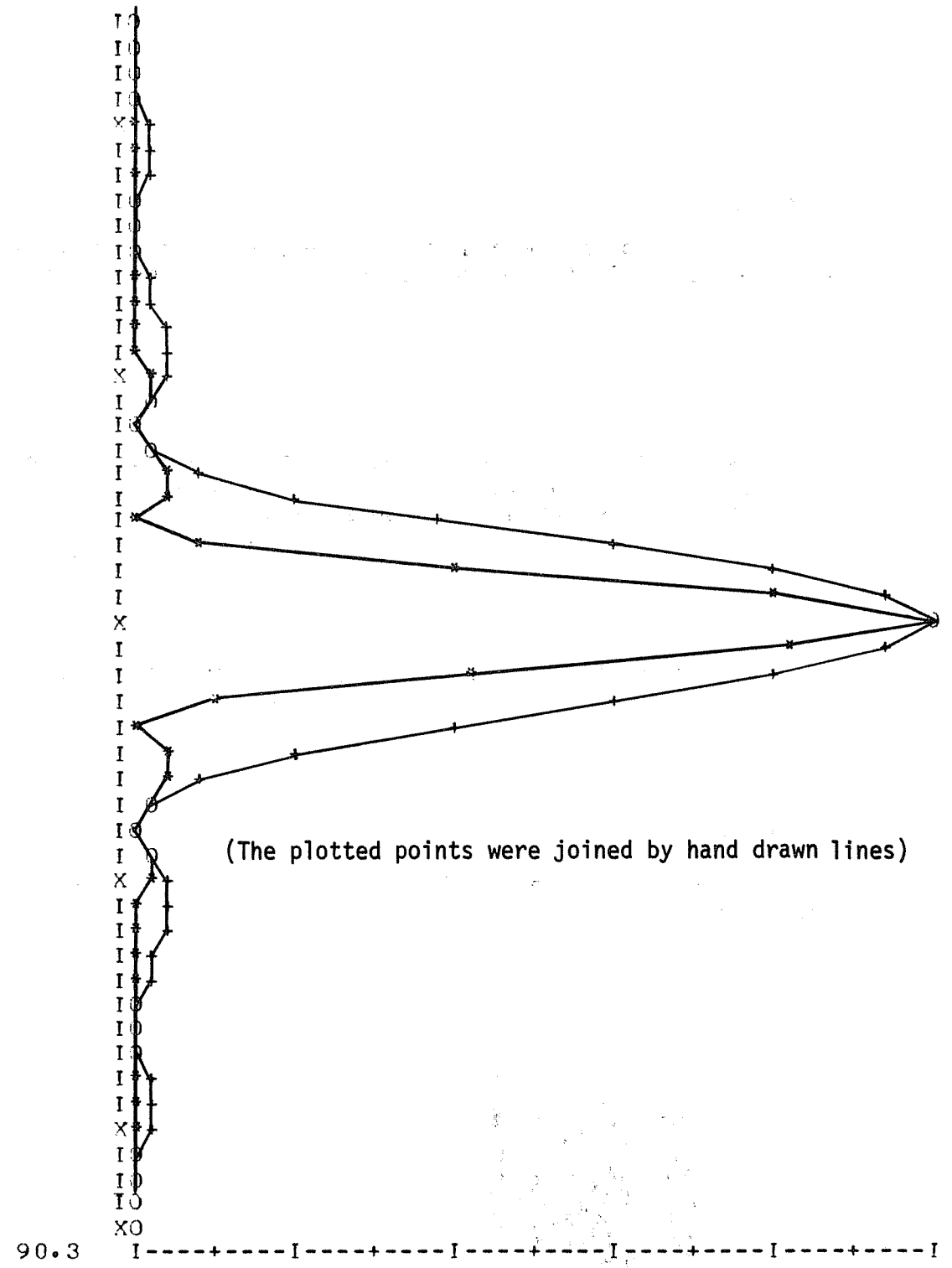

```
TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES
TYPE (1) FOR NEW EQUATIONS ONLY
 TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN  0900, 0910)
TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE  0940)
?2
```

```
?50,1,1


Y IS (*)
Z=(DEP.)  (+)
(O) IS COMMON DATA POINT
```




```
TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES
TYPE (1) FOR NEW EQUATIONS ONLY
 TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN 0900, 0910)
TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE  0940)
?S

READY
*
```

```
 0940  LET R0=120
*RUN


(ANTENNA) OR (WAVE) EQUATION PLOTTING PROGRAM
SIN(X) PEAKS AT (0) DEGREES


TYPE A (1) FOR SIN(X) DEPENDENCY
TYPE A (2) FOR COS(X) DEPENDENCY
TYPE A (3) FOR JUST (X) DEPENDENCY
TYPE A (1) FOR Z1=SIN(N*DEP.*S)/N*SIN(DEP.*S)
TYPE A (2) FOR Z2=SIN(DEP.*L)/(DEP.*L)
TYPE A (3) FOR Z3=Z1*Z2
TYPE A (0) FOR ABS. VALUE OF Y,(1) AMP., (2) INTEN.
(DEP.) = (C*P*X) WHERE C IS SET=1/P, LINE 240

?2,1,2


INPUT N THE NUMBER OF ELEMENTS
INPUT S THE SPACING IN WAVELENGTHS
INPUT L THE WIDTH OF THE ARRAY IN WAVELENGTHS
IF NOT KNOWN LET N=10, S=1, AND L=1

?60,48,1467



Y IS (*)
Z=         (+)  (Z2)^2
(O) IS COMMON DATA POINT

                 0        ( Y INCREMENT =   .02 )                      1

         89.7    I----+----I----+----I----+----I----+----I----+----I----+----I
                XO
                IO
                IO
                IO
                IO
 DELTA X        IO
(DEGREES)       IO
( .005 )        IO
                IO
                IO
                IO
                XO
                IO
                IO
                IO
                IO
                IO
```



(The plotted points were joined by hand drawn lines)

```
TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES
TYPE (1) FOR NEW EQUATIONS ONLY
 TYPE (2) FOR NEW DATA ONLY. (X LIMITS IN 0900, 0910)
TYPE (3) FOR ALL NEW VALUES. (X SCALE IN LINE  0940)
?2
```

LOLA:

DESCRIPTION

This is a navigation program and will enable the user to determine position in latitude and longitude from data supplied from sextant readings and a nautical almanac.

USERS

Anyone who wants to determine his location with a sextant and doesn't want to go through the laborious hand calculations would find LOLA very useful. This would include most small vessel owners and crewmen.

INSTRUCTIONS

List the program for instructions. All inputs are entered while the program is running.

LIMITATIONS

Execution and storage of Lola will require about 6K Bytes of memory in most computer systems.

LOLA

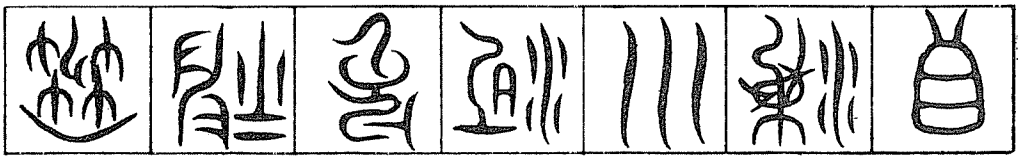
```
100 PRINT "TYPE 1 IF FIX    2 IF INTERSTELLAR DISTANCE"
110 PRINT " MAKE UNNEEDED VALUES 0"
120 PRINT"LIST LINES 140 TO 240 FOR INPUT DESCRIPTIONS"
130 INPUT X9
140 REM ALL INPUTS ARE IN DEGREES AND MINUTES
150 REM NORTH (+)  SOUTH (-)
160 REM A1&A2 CORRECTED DECLINATION OF 1ST BODY
170 REM A3&A4 SIDERIAL HOUR ANGLE 1 ST BODY
180 REM A5&A6 CORRECTED GHA(ARIES FOR STAR SIGHT)
190 REM H5&H3 CORRECTED ALTITUDE (HO) 1 ST BODY
200 REM V1&V2 CORRECTED DECLINATION 2ND BODY
210 REM V3&V4 SIDERIAL HOUR ANGLE 2ND BODY
220 REM V5&V6 CORRECTED GHA (ARIES FOR STAR SIGHT)
230 REM H4&H9 CORRECTED ALTITUDE 2ND BODY
240 PRINT"A1,A2,A3,A4,A5,A6,H5,H3"
250 INPUT A1,A2,A3,A4,A5,A6,H5,H3
260 PRINT"V1,V2,V3,V4,V5,V6,H4,H9"
270 INPUT V1,V2,V3,V4,V5,V6,H4,H9
280  Y=57.2957795
290  Y1=1.57079632
300 PRINT
310  L5=A3+(A4/60)+A5+(A6/60)
320  T1=L5
330 GOSUB 2500
340  T3=T2
350  L7=V3+(V4/60)+V5+(V6/60)
360  T1=L7
370 GOSUB 2500
380  T4=T2
390  X4=ABS(T4-T3)
400 IF T4>T3 THEN 420
410 GOTO 450
420 IF X4<180 GOTO 510
430  X4=360-X4
440 GOTO 560
450 IF X4<180 THEN 560
460  X4=360-X4
510  B3=(A1+(A2/60))/Y
520  B1=(V1+(V2/60))/Y
530  B8=(90-(H5+(H3/60)))/Y
540  B7=(90-(H4+(H9/60)))/Y
550 GOTO 620
```

```
560  B3=(U1+(U2/60))/Y
570  B1=(A1+(A2/60))/Y
580  B8=(90-(H4+(H9/60)))/Y
590  B7=(90-(H5+(H3/60)))/Y
600  L5=U3+(U4/60)+U5+(U6/60)
610  L7=A3+(A4/60)+A5+(A6/60)
620  X5=X4/Y
630 IF X5>Y1 THEN 650
640 GOTO 670
650 B=(2*Y1)-X5
660 GOTO 680
670  B=X5
680 GOSUB 1010
690  B9=H2/Y
700  F1=COS(B7)
710  F2=COS(B8)*COS(B9)
720  F3=SIN(B8)*SIN(B9)
730  F4=(F1-F2)/F3
740  F6=ATN(((1-(F4↑2))↑.5)/F4)
750 IF F6>0THEN 800
770  F9=2*Y1+F6
780  F5=ABS(F9)
790 GOTO 810
800  F5=ABS(F6)
810  F=F5*Y
830  Z9=Q1
840  T8=180-Q1
850  X5=(Q1/Y1)-F5
860  B=X5
870  B1=Y1-B8
880 GOSUB 1010
890  Z5=Z9
900 GOSUB 2020
910 PRINT "FOR SECOND FIX TYPE 1 RETURN TYPE 2"
920 INPUT P9
930 IF P9=1 THEN 950
940 GOTO 240
950  X5=(Z9/Y)+F5
960  B=X5
970 GOSUB 1010
980  Z5=T8
990 GOSUB 2020
1000 GOTO 240
1010  P=ATN((SIN(B)*COS(B1))/(1-(SIN(B)*COS(B1))↑2)↑.5)
1020  K=ATN((SIN(B1)/COS(P))/(1-(SIN(B1)/COS(P))↑2)↑.5)
1030  K1=ABS(K)
1040 IF X5>Y1 THEN 1060
1050 GOTO 1120
1060 IF K1>Y1 THEN 1080
1070 GOTO 1100
1080  K2=K1
```

```
1090 GOTO 1170
1100  K2=2*Y1-K1
1110 GOTO1170
1120 IF K1>Y1 THEN 1140
1130 GOTO 1160
1140  K2=(2*Y1)-K1
1150 GOTO 1170
1160  K2=K1
1170 IF B1>0THEN 1190
1180 GOTO 1210
1190  K3=ABS(K2)
1200 GOTO 1260
1210 IF K>0 THEN 1230
1220 GOTO 1250
1230  K3=K2
1240 GOTO 1260
1250  K3=-K2
1260  K4=K3*Y
1270  K5=ABS(K3)
1280  B4=-B3
1290  G1=K3+B4
1300  G3=ABS (G1)
1310  G4=G3*Y
1320 IF G4>180 THEN 1350
1330 IF G4>90 THEN 1370
1340 IF G4<90 THEN 1390
1350  G5=G4-180
1360 GOTO 1400
1370  G5=180-G4
1380 GOTO 1400
1390  G5=G4
1400 GOTO 1410
1410  G2=G5/Y
1420  H=ATN(((1-(COS(P)*COS(G2))↑2)↑.5)/(COS(P)*COS(G2)))
1430  R3=ABS(H)
1440  H1=R3*Y
1450  G6=B3/B1
1460  B4=-B3
1470  G1=K3+B4
1480  G3=ABS(G1)
1490 IF G6>0THEN 1510
1500 GOTO 1550
1510 IF X5>Y1 THEN 1530
1520 GOTO 1650
1530 GOTO 1590
1540 IF G3<Y1 THEN 1650
1550 IF X5<Y1 THEN 1570
1560 GOTO 1590
1570 IF G3<Y1 THEN 1650
1590  H2=180-H1
1640 GOTO 1710
```

```
1650  H2=H1
1710  H8=ABS(H2)
1730 IF X9=1 THEN 1750
1740 IF X9=2 THEN 2400
1750  H7=H8/Y
1760  Z=ATN((SIN(P)/SIN(H7))/(1-(SIN(P)/SIN(H7))↑2)↑.5)
1770  Z6=ABS(Z)
1780  Z1=Z6*Y
1790 IF G6>0 THEN 1805
1800 GOTO 1830
1805  A=ABS(B3)
1810 IF K5>A THEN 1850
1820 GOTO 1910
1830 IF G3>2*Y1 THEN 1850
1840 GOTO 1910
1850  Q1=Z1
1900 GOTO 1960
1910  Q1=180-Z1
1960 IF B3>0 THEN 2010
1980  Q1=180-Q1
2010 RETURN
2020  U=90-H2
2030 IF U>0THEN 2050
2040 IF U<0THEN 2080
2050  U1=U
2060 PRINT"NORTH LATITUDE"
2070 GOTO 2100
2080  U1=-U
2090 PRINT"SOUTH LATITUDE"
2100  U8=ABS(U1)
2110 PRINT U8
2120  U2=INT(U8)
2130  U3=(U1-U2)*60
2140 PRINT     "DEGREES      MINUTES"
2150 PRINT U2,U3
2160 IF Z5>F THEN 2200
2180  U4=L5-Q1
2190 GOTO 2210
2200  U4=L5+Q1
2210 IF U4>360THEN 2260
2220 IF U4>180 THEN 2280
2230 IF U4<180 THEN 2240
2240 IF U4>0THEN 2300
2250 IF U4<0 THEN 2320
2260  U5=U4-360
2270 GOTO 2340
2280  U5=360-U4
2290 GOTO 2360
2300  U5=U4
2310 GOTO 2340
2320  U5=ABS(U4)
```

```
2330 GOTO 2360
2340 PRINT"WEST LONGITUDE"
2350 GOTO 2370
2360 PRINT"EAST LONGITUDE"
2370 GOSUB 2420
2380 PRINT
2390 RETURN
2400 PRINT "INTERSTELLAR DIST"
2410  U5=H8
2420 PRINT U5
2430  U6=INT(U5)
2440  U7=(U5-U6)*60
2450 PRINT     "DEGREES      MINUTES"
2460 PRINT U6,U7
2470 IF X9=1 THEN 2490
2480 IF X9=2 THEN 240
2490 RETURN
2500 IF T1>720 THEN 2530
2510 IF T1>360 THEN 2550
2520 IF T1<360 THEN 2570
2530  T2=T1-720
2540 GOTO 2580
2550  T2=T1-360
2560 GOTO 2580
2570  T2=T1
2580 RETURN
2590 END
```



MACRO:

DESCRIPTION

This program simulates a high level language assembler. The program may also be used to study emulator functioning. It can also be used to amuse yourself while trying to determine what Macro is doing.

USERS

The cypher enthusiast, advanced programmer, and computer system analyst will find this program useful and enjoyable. Not recommended for the novice.

INSTRUCTIONS

Macro requires an Input use file and an Output use file. The Output file is not used in this version and may be deleted from line 170 if desired. To use this program you must pick an available file, preferably one with something in it. Then name it in the File Input Statement and type RUN. The teletype will then display the compiling function. The Input use file is not altered.

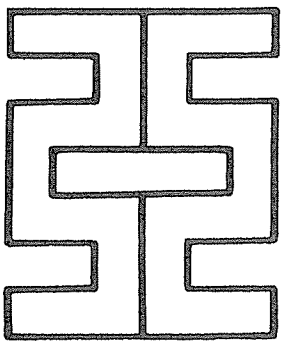
LIMITATIONS

In line 170 there is a FILES statement. The SST( ) statement is used extensively throughout this program. This program requires file manipulations. This program stores in 2K Bytes of memory. Execution length is a function of the length of the designated Input file.

MACRO

```
1 REM THIS PROGRAM REQUIRES 2 FILES - INPUT - OUTPUT. LINE 170
10 REM MACRO
20 REM\MACRO
30 REMMACRO
40 REMMACRO
50 PRINT TIM(x) ;" SECONDS ";CLK$;"  ";DAT$
60 PRINT " TYPE CHAR AND CR "
70 LET T = TIM(x)
80 INPUT A$
90 REM DIMENSIONS
100 DIM B(80)
110 DIM C(80)
120 DIM D(80)
130 REM CONSTANTS
140 LET C = ASC( )
150 LET D$ = "LET"
160 REM FILEDEF
170 FILES INPUT;OUTPUT
180 DELIMIT #1,(CR)
190 REM ACTIVE CODE
200  GOTO 600
210 PRINT "LINE DONE"
220 REM GET NEXT LINE
230 LET J = 0
240 LET F = 0
250 INPUT #1,A$
260 CHANGE A$ TO B
270 LET K = 1
280 REM GO TO SCANNER
290 GOTO 400
300 REM CLEAR LINE
310 FOR I = 1 TO 80
320 LET B(I) = C(I)
330 LET D(I) = C(I)
340 NEXT I
350 GOTO 250
360 LET T1 = TIM(x)
370 LET D = T1 - T
380 PRINT "TIME = ";D
390 REM SCANNER
400 LET X = B(K)
410 LET K = K + 1
```

```
420 IF x = 0 GOTO 200
430 IF x = 32 GOTO 530
440 IF x < 64  GOTO  500
450 LET F = 80
460 LET J = J+1
470 LET D(J) =x
480 LET D(0) = J
490 GOTO 510
500 IF K = 80 GOTO 200
510 GOTO 400
520 REM FINDS SECOND SPACE AFTER DATA
530 LET F = F + 1
540 IF F > 80 GOTO570
550 GOTO 400
560 GOTO 430
570 LET x = x
580 CHANGE D TO D$
590  GOTO 200
600 IF D$ = "REMMACRO" GOTO 640
610 IF D$ = "REMMACRO" GOTO640
620 PRINT D$
630 GOTO230
640 PRINT "MADE IT "
650 PRINT D$
660 PRINT "NORMAL"
670 GOTO 210
680 END
```



MAX. MIN.:

DESCRIPTION

This program calculates the maximum, minimum, and zero points, if they exist, of any function over any fixed interval.

USERS

People solving equations for roots or extreme values could put this program to very good use, such as: high school students, engineers, etc.

INSTRUCTIONS

Enter all of your data into the program before execution, in the following format:

```
100 DATA  MIN,MAX,ACC,INC
200 Let Y = f(x)
```

where

MIN and MAX - define the interval
ACC - is the accuracy number of significant figures desired
INC - number of increments the interval is divided (try 100)

Then type RUN.

LIMITATIONS

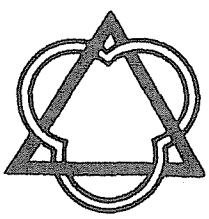
Max. Min. will store in 3K Bytes of memory and execute in 4K Bytes.

MAX. MIN.

```
30 LET Z0 = 0
100 GO TO 1490
104 LET Y = 1E38
108 READ  I, F, Q, W
112 IF I<>F THEN 140
140 PRINT
144 PRINT"POINT-TYPE", TAB(23), "F(X)", "
148 PRINT
152 LET K=(F-I)/W
156 LET K1=K
160 LET Q5=INT(0.4343*LOG(0.5*(ABS(I)+ABS(F))))
164 LET M=0.5*(10↑(Q5-Q))
168 LET F1=F+K
172 LET R=0
176 LET X=I-K
180 LET T=2
200 LET Y=1E38
705 IF Y<>1E38 THEN 710
706 IF I=F THEN 1490
710 IF T<2 THEN 940
720 IF T=3 THEN 750
730 LET T=3
740 GOTO 1150
750 IF Y=Y1 THEN 870
760 IF Y<Y1 THEN 830
770 LET D=1
780 IF Y1<0 THEN 810
790 LET T=-1
800 GOTO 1070
810 LET T=1
820 GOTO 960
830 LET D=-1
840 IF Y1>0 THEN 810
850 LET T=0
860 GOTO 1100
870 IF K1<M THEN 910
880 LET K1=K1/2
890 LET X=X1+K1
900 GOTO 1180
910 PRINT
920 PRINT "NO CHANGE...."
930 STOP
940 IF T<1 THEN 1050
950 IF Y=0 THEN 1300
```

```
960 IF D=-1 THEN 1010
970 IF Y>0 THEN 1200
980 IF ABS(Y)<ABS(Y1) THEN 1150
990 LET T=-1
1000 GOTO 1070
1010 IF Y<0 THEN 1200
1020 IF ABS(Y)<ABS(Y1) THEN 1150
1030 LET T=0
1040 GOTO 1100
1050 IF R=1 THEN 1470
1060 IF T=0 THEN 1100
1070 IF Y>Y1 THEN 1150
1080 IF K1<M THEN 1330
1090 GOTO 1120
1100 IF Y<Y1 THEN 1150
1110 IF K1<M THEN 1410
1120 LET X1=X1-K1
1130 LET R=1
1140 GOTO 880
1150 LET Y1=Y
1160 LET X1=X
1170 LET X=X+K1
1180 IF X<F1 THEN 200
1190 STOP
1200 IF K1<M THEN 1220
1210 GOTO 880
1220 IF ABS(Y)<ABS(Y1) THEN 1300
1230 PRINT
1235 PRINT " ZERO", Z0, X1
1240 LET K1=K
1250 IF D=1 THEN 1280
1260 LET T=0
1270 GOTO 1150
1280 LET T=-1
1290 GOTO 1150
1300 LET Y1=Y
1310 LET X1=X
1320 GOTO 1230
1330 PRINT "  MAX", Y1, X1
1340 LET K1=K
1350 LET D=-1
1360 IF Y1<0 THEN 1390
1370 LET T=1
1380 GOTO 1150
1390 LET T=0
1400 GOTO 1150
1410 PRINT "  MIN", Y1, X1
1420 LET K1=K
1430 LET D=1
1440 IF Y1<0 THEN 1370
1450 LET T=-1
```

```
1460 GOTO 1150
1470 LET R=0
1480 GOTO 1150
1490 PRINT
1500 PRINT "THIS PROGRAM WILL LOCATE 'INTERESTING' VALUES"
1510 PRINT "OF X FOR ANY FUNCTION OF X.  SPECIFICALLY, IT"
1520 PRINT "WILL LOCATE THE VALUES OF X AT WHICH RELATIVE"
1530 PRINT "MAXIMUMS AND MINIMUMS OF F(X) OCCUR, AND THE"
1540 PRINT "VALUES OF X FOR WHICH F(X) IS ZERO (I.E., THE"
1550 PRINT "ZEROES OR ROOTS OF THE FUNCTION)."
1560 PRINT
1570 PRINT " TO USE THE PROGRAM, TYPE THE FOLLOWING :"
1580 PRINT
1590 PRINT "       100  DATA <XMIN>,<XMAX>,<ACC>,<INCR>"
1600 PRINT "       200  LET Y = <A FUNCTION OF X>"
1610 PRINT "       RUN"
1620 PRINT
1630 IF 00 = 1 THEN 9999
1640 PRINT "WHERE 'XMIN' AND 'XMAX' DEFINE THE INTERVAL IN"
1650 PRINT "WHICH INTERESTING VALUES OF X ARE TO BE SOUGHT,"
1660 PRINT "'ACC' IS THE ACCURACY (IN NUMBER OF SIGNIFICANT"
1670 PRINT "FIGURES) TO WHICH THE ZEROES OF X AND THE MAX"
1680 PRINT "AND MIN VALUES OF F(X) ARE TO BE ESTIMATED,"
1690 PRINT "'INCR' IS THE NUMBER OF INCREMENTS INTO WHICH"
1700 PRINT"TOTAL INTERVAL IS TO BE DIVIDED FOR SEARCH (TRY 50 TO START)"
1710 PRINT "AND THE FUNCTION IS ANY LEGITIMATE BASIC-LANGUAGE EXPRESSION"
1720 PRINT "INVOLVING THE VARIABLE X."
1730 PRINT "YOU CAN USE LINES 200 THRU 700 FOR THE FUNCTION."
9999 END
```



MAX. MIN.

```
100 DATA 0,725,3,200
200 LET Y=SIN(X*3.1415965/180)

RUN

POINT-TYPE          F(X)              X

ZERO                 0                0
 MAX                 1         90.00195

ZERO                 0         180.0039
 MIN                -1         270.0059

ZERO                 0         360.0078
 MAX                 1         450.0098

ZERO                 0         540.0117
 MIN                -1         630.0137

ZERO                 0         719.9873
```

NAVAID:

DESCRIPTION

Navaid is used to determine the Altitude and Azimuth of any known celestial body. This can be used to determine what celestial body you may be looking at or to determine where to look for a particular celestial body.

USERS

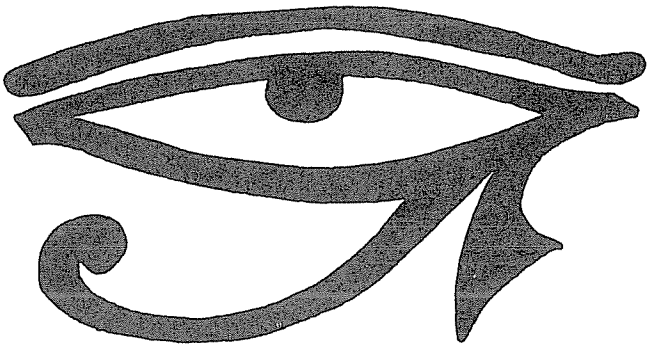
Amateur astronomers and people interested in navigation would find this program useful. It would also be of use to mariners.

INSTRUCTIONS

For detailed program information list the program. All necessary information will be asked for when the program is run.

LIMITATIONS

Navaid should execute without problem in any Basic speaking computer with 5K Bytes of program space available.



NAVAID

```
100 PRINT "ALTITUDE & AZIMUTH OF CELESTIAL BODY"
110 PRINT "LIST LINES 130 TO 300 FOR INPUT DESCRIPTIONS"
120 REM ALL VALUES IN DEGREES AND MINUTES
130 REM NORTH VALUES ARE +
140 REM SOUTH VALUES ARE -
150 REM EAST VALUES ARE +
160 REM WEST VALUES ARE -
170 REM A,A1 SIDERIAL HOUR ANGLE DEG       MIN
180 REM A2,A3  GHA EVEN HOUR       "          "
190 REM A4,A5 GHA  MIN SEC         "        "
200 REM  A6  V CORRECTION                     "
210 REM A7,A8  DR OR AP LONGITUDE "         "
220 REM D,D1 DECLINATION OF BODY  "         "
230 REM D2  D CORRECTION                    "
240 REM FOR SOUTH DEC  CHANGE SIGN OF D COR
250 REM L,L1  DR OR AP LATITUDE    "          "
260 REM W,W1, SEXTANT ALTITUDE     "          "
270 REM W2 INDEX CORRECTION                    "
280 REM W3     DIP                            "
290 REM A4/ ALTITUDE  CORRECTION              "
300 REM  THESE VALUES ARE PUT IN EXACTLY AS DONE ON  HO 211 "
310 LET Y=57.29577951
320 LET Y1=1.57079632
330 LET Y2=3.14159265
340 PRINT "THERE ARE 19 INPUTS AS GIVEN ON PROGRAM SHEET"
350 PRINT "A,A1,A2,A3,A4,A5,A6,A7,A8"
360 INPUT A,A1,A2,A3,A4,A5,A6,A7,A8
370 PRINT "D,D1,D2,L,L1,W,W1,W2,W3,W4"
380 INPUT D,D1,D2,L,L1,W,W1,W2,W3,W4
390 LET B6=A+(A1/60)+A2+(A3/60)+A4
400 LET C2=B6+(A5/60)+(A6/60)+A7+(A8/60)
410 LET B1=(D+(D1/60)+(D2/60))/Y
420 IF C2>720 THEN 480
430 IF C2>540 THEN 500
440 IF C2>360 THEN 520
450 IF C2>180 THEN 540
460 IF C2<180 THEN 560
470 IF C2<0 THEN 580
480 LET X2=720-C2
490 GOTO 590
500 LET X2=720-C2
```

```
510 GOTO 590
520 LET X2=360-C2
530 GOTO 590
540 LET X2=360-C2
550 GOTO 590
560 LET X2=-C2
570 GOTO 590
580 LET X2=-C2
590 LET X4=ABS(X2)/Y
600 LET A=ABS(X2)
610 GOSUB 1550
620 PRINT"T DEGREES";A1;"MINUTES";A2
630 IF X4>Y1 THEN 650
640 GOTO 670
650 LET B=Y2 -X4
660 GOTO 680
670 LET B=X4
680 LET B=ABS (B)
690 LET B3=(L+(L1/60))/Y
700 LET P=ATN((SIN(B)*COS(B1))/(1-(SIN(B)*COS(B1))↑2)↑.5)
710 LET K=ATN((SIN(B1)/COS(P))/(1-(SIN(B1)/COS(P))↑2)↑.5)
720 LET K1=ABS(K)*Y
730 IF X4>Y1 THEN 800
740 IF K1>90 THEN 760
750 GOTO 780
760 LET K2=180-K1
770 GOTO 850
780 LET K2=K1
790 GOTO 850
800 IF K1>90 THEN 820
810 GOTO 840
820 LET K2=K1
830 GOTO 850
840 LET K2=180-K1
850 IF K>0 THEN 870
860 GOTO 890
870 LET K3=K2
880 GOTO 900
890 LET K3=-K2
900 LET A=ABS(K3)
910 GOSUB 1550
920 PRINT "K DEGREES";A1;"MINUTES";A2
930 LET K4=K3/Y
940 LET B4=-B3
950 LET G1=K4+B4
960 LET G2=ABS(G1)
970 LET A=ABS(G2*Y)
980 GOSUB 1550
990 PRINT"K#L DEGRC⊏⊐S";A1;"MINUTES";A2
1000 LETH=ATN((COS(P)*COS(G2))/(1-(COS(P)*COS(G2))↑2)↑.5)
1010 LET H1=H*Y
```

```
1020 LET A=H1
1030 GOSUB 1550
1040 LET H3=A1
1050 LET H5=A2
1060 LET Z=ATN((SIN(P)/COS(H))/(1-(SIN(P)/COS(H))↑2)↑.5)
1070 LET Z1=Z*Y
1080 LET D3=K/B3
1090 LET B7=ABS(B3)
1100 LET K8=K2/Y
1110 IF D3>0 THEN 1130
1120 GOTO 1140
1130 IF K8>B7 GOTO 1160
1140 IF Z1>90 THEN 1180
1150 GOTO 1200
1160 IF Z1>90 THEN 1200
1170 GOTO 1180
1180 LET Q1=Z1
1190 GOTO 1210
1200 LET Q1=180-Z1
1210 IF B3>0THEN 1230
1220 GOTO 1290
1230 IF X2>0 THEN1250
1240 GOTO 1270
1250 LET D5=Q1
1260 GOTO 1340
1270 LET D5=360-Q1
1280 GOTO 1340
1290 IF X2>0THEN 1310
1300 GOTO 1330
1310 LET D5=180-Q1
1320 GOTO 1340
1330 LET D5=180+Q1
1340 PRINT
1350 PRINT
1360 PRINT
1370 PRINT"HC DEGREES";H3;"HC MINUTES";H5
1380 PRINT
1390 PRINT"Z";Q1;"ZN";D5
1400 PRINT
1410 LET D7=W+(W1/60)+(W2/60)+(W3/60)+(W4/60)
1420 LET A=D7
1430 GOSUB 1550
1440 PRINT "HO DEGREES";A1;"HO MINUTES";A2
1450 LET D8=D7-H1
1460 LET D9=ABS(D8*60)
1470 IF D7>H1 GOTO 1500
1480 GOTO 1530
1490 PRINT
1500 PRINT "A";D9;"TOWARD"
1510 PRINT
1520 GOTO 1580
1530 PRINT "A";D9;"AWAY"
1535 PRINT
1540 GOTO 1580
1550 LET A1=INT(A)
1560 LET A2=(A-A1)*60
1570 RETURN
1580 GOTO 350
1590 END
```

OPTICAL:

DESCRIPTION

Optical performs blackbody radiation calculations and conversions. It is programmed with look-up tables for various optical elements. The program's calculations are accurate from about 100 Anstroms to microwave wavelengths. The output may be listed and plotted on standard terminals.

USERS

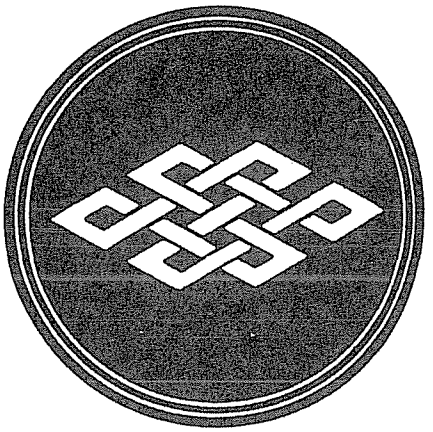
This program would be of considerable use to individuals who measure heat, optical transmissions, or temperatures. Laser engineers, optical military weapon system specialists, S.H.F. communications engineers, etc. Infared specialists will have considerable use for this program in determining complex energy transmissions.

INSTRUCTIONS

The program will prompt for all necessary data while the program is executing. This program will work for any type of Thermal Radiators. The program may be listed for additional information.

LIMITATIONS

This program is 16K Bytes long and may be restrictive due to it's length. It will require an additional 8K Bytes for execution space. It should execute in any Basic system having enough available memory.



OPTICAL

```
70 REM THIS PROGRAM CALCULATES BLACKBODY PROBLEMS
90 PRINT "THIS PROGRAM CALCULATES BLACKBODY PROBLEMS"
110 PRINT "IT IS VERY FLEXIBLE AND MAKES NO ASSUMPTIONS"
130 PRINT "BE CAREFUL TO ENTER ALL UNITS AS INDICATED"
140 DIM T1(50),R5$(50),V1(50),I1(50),R3(50),R3$(50),P5(50)
142 DIM P5$(50),R1(50),R1$(50)
150 PRINT
170 PRINT"#### REMEMBER USE WITH THERMAL RADIATORS ONLY ####"
190 PRINT
210 PRINT
230 PRINT
270 LET A1=(1.0E-04)*(1.0E-03)
290 REM THIS CONSTANT IS USED IN THE CALC. OF W ONLY
310 LET C5=0.4342945
330 REM C5 IS .4342945 AND CONVERTS LOG E TO LOG 10
350 LET E2=0.25
370 REM THIS IS THE EMISSIVITY GENERATOR FOR TUNGSTEN
390 LET D3=1
410 REM D3 IS USED IN THE LUMENS/STR CONVERSION
430 LET A2=5.6686E-05
450 LET A3=1.38054E-16
470 LET A4=2897.9
490 LET K2=2404
510 LET D4=3670
530 LET C=2.99793E10
550 LET A5=273.18
570 LET A6=1.4388
590 REM A6=C*H/A3
610 LET P=3.14159
630 LET H=6.6252E-27
650 REM THESE CONSTANTS ARE IN THE (CGS) SYSTEM
670 LET A7=(0.0065)
690 REM A7 IS THE APERATURE IN CM**2. OR (0.0065)
710 REM THIS IS 1/2 THE FILAMENT SURFACE AREA OF A #44 BULB IN CM**2
730 LET S=0.001
750 REM S IS THE STEP SIZE IN MICRONS
770 LET B7=0.1
790 REM THIS IS THE ERROR IN MEASURING A7 IN %
810 LET B1=0.06
830 REM THIS IS THE ERROR IN MEASURING T1
850 REM ** THESE BLKBDY CALC ARE FOR A LAMBERTIAN HEMISPHERE
870 PRINT
```

```
890 PRINT"DO YOU WANT CONVERSIONS (1), OR BLACKBODY CALC. (2)"
910 PRINT
930 INPUT Q
950 IF Q=1 GOTO 8270
970 PRINT
990 LET R9$="NO"
1010 LET I1=V1=N7=N8=N9=0
1030 PRINT"HOW MANY TEMPERATURES DO YOU WANT TO RUN. 50 IS MAX."
1050 PRINT
1070 INPUT N1
1090 PRINT
1110 PRINT "DO YOU WANT TO INPUT THE APERTURE AREA? ELSE (.0065)"
1130 INPUT Q$
1150 IF Q$="N" GOTO 1290
1170 IF Q$="NO" GOTO 1290
1190 PRINT
1210 PRINT "TYPE IN THE AREA IN (CM**2)"
1230 PRINT
1250 INPUT A7
1270 PRINT
1290 PRINT
1310 PRINT "DO YOU WISH TO INPUT VOLTAGE AND CURRENT"
1330 INPUT R$
1350 PRINT
1370 IF R$="N" GO TO 1490
1390 IF R$="NO" GO TO 1490
1410 PRINT "INPUT TEMPERATURE (DEGREES), SCALE (R), (F), (C), (K)"
1430 PRINT "AND (VOLTS), THEN (AMPS)"
1450 IF R$="Y" GOTO 1550
1470 IF R$="YES" GOTO 1550
1490 PRINT "INPUT TEMPERATURE (DEGREES), SCALE (R), (F), (C), (K)"
1510 IF R$="N" GO TO 1670
1530 IF R$="NO" GOTO 1670
1550 PRINT
1570 FOR D9=1 TO N1
1590 INPUT T1(D9), R5$(D9), V1(D9), I1(D9)
1610 NEXT D9
1630 IF R$="Y" GO TO 1750
1650 IF R$="YES" GO TO 1750
1670 PRINT
1690 FOR D9=1 TO N1
1710 INPUT T1(D9), R5$(D9)
1720 LET V1(D9)=I1(D9)=0
1730 NEXT D9
1750 PRINT
1770 FOR D9=1 TO N1
1790 IF R5$(D9)="R" GOTO 1970
1810 IF R5$(D9)="F" GOTO 2010
1830 IF R5$(D9)="C" GOTO 1890
1850 IF R5$(D9)="K" GOTO 1930
1870 GOTO 1030
```

```
1890 LET T=T1(D9)+A5
1910 GOTO 2030
1930 LET T=T1(D9)
1950 GOTO 2030
1970 LET T=(T1(D9)-491.7)*(5/9)+A5
1990 GOTO 2030
2010 LET T=(T1(D9)-32)*(5/9)+A5
2030 GOTO 2050
2050 IF N9=1 GO TO 2650
2070 PRINT
2090 PRINT "INPUT THE NUMBER OF SPECTRAL WINDOWS YOU WANT CALCULATED"
2110 PRINT
2130 INPUT N
2150 PRINT
2170 PRINT
2190 PRINT"ENTER END POINT (MIN), (MAX) MICRONS AND TRANS. IN %"
2210 PRINT "** THE (%) SHOULD BE >9.  TYPE (0) FOR SPECIAL LIST"
2230 PRINT
2250 FOR D8=1 TO N
2270 INPUT F1(D8), F2(D8), T5(D8)
2290 IF T5(D8)>0 GOTO 2590
2310 PRINT
2330 PRINT
2350 PRINT "      1.)  TYPE A (1) FOR A (2 MM) GLASS FILTER"
2370 PRINT "      2.)  TYPE A (2) FOR A (2 MM) SILICON FILTER"
2390 PRINT "      3.)  TYPE A (3) FOR A (2 MM) GERMANIUM FILTER"
2410 PRINT "      4.)  TYPE A (4) FOR A LEAD SULFIDE DETECTOR"
2430 PRINT "      5.)  TYPE A (5) FOR THE VISUAL RESPONSE CURVE"
2450 PRINT "      6.)  TYPE A (6) FOR A SILICON DETECTOR"
2470 PRINT "      7.)  TYPE A (7) FOR #(3)*(4)"
2490 PRINT "      8.)  TYPE A (8) FOR #(1)*(2)*(4)"
2510 PRINT "      9.)  TYPE A (9) FOR #(1)*(3)*(4)"
2530 PRINT
2550 PRINT
2570 GOTO 2190
2590 NEXT D8
2610 PRINT
2630 PRINT
2650 LET L=A4/T
2670 REM L IS THE MAX WAVELENGTH FOR THE GIVEN T IN DEG. C
2690 LET E3=21
2710 REM E3 IS THE VOLUMETRIC CORRECTION FACTOR
2730 REM E3 IS 21 (AVERAGED) FOR A #44 BULB, WHERE  12<E3<35
2750 IF L>6 GOTO 2810
2770 LET E1=L/(E2*E3)
2790 GOTO 2830
2810 LET E1=1/(E3*(0.04))
2830 LET N9=1
2850 LET L2=K2/T
2870 REM THIS IS THE WAVELENGTH FOR MAX. RADIANCE DIFFERENCE/MICRON
2890 LET L4=D4/T
```

```
2910 REM THIS IS THE WAVELENGTH FOR MAX. PHOTON EMISSION FOR T
2930 LET W=A1*A2*(T**4)
2950 REM W IS THE RADIANT EMITTANCE IN (WATTS/CM**2)
2970 LET W1=(2*W*A7*E1)
2990 REM W1 IS THE SOURCE RADIANT POWER IN WATTS
3010 FOR D8=1 TO N
3030 LET K6=0
3050 LET G1=F2(D8)-F1(D8)
3070 LET F3=W2=W3=W4=W8=0
3090 IF F1(D8)>S GO TO 3130
3110 LET F1(D8)=S
3130 FOR X=0 TO G1 STEP S
3150 LET F3=F1(D8)+X
3170 GOTO 3190
3190 LET C4=10000
3210 LET K3=(A6*C4)/(T*F3)
3230 IF K3<88 GOTO 3270
3250 LET K3=88
3270 LET K4=EXP(K3)
3290 LET K5=K4-1
3310 REM ##### T6 IS THE TRANSMISSION FACTOR
3330 IF F3<4.4 GOTO 3390
3350 LET U1=(2.3118E08)/(F3**10)
3370 GOTO 3650
3390 IF F3<3.1 GOTO 3450
3410 LET U1=19.4*F3
3430 GOTO 3650
3450 IF F3<2.8 GOTO 3510
3470 LET U1=60
3490 GOTO 3650
3510 IF F3<.5 GOTO 3570
3530 LET U1=99
3550 GOTO 3650
3570 IF F3<.3 GOTO 3630
3590 LET U1=(124.7)*F3**(1/3)
3610 GOTO 3650
3630 LET U1=EXP(33*F3)/(240.12)
3650 LET U1=U1/100
3670 REM U1 IS FOR AN ORDINARY GLASS FILTER
3690 IF F3<14 GOTO 3750
3710 LET U2=(22294)/(F3+8.77)**2
3730 GOTO 4070
3750 IF F3<11.5 GOTO 3810
3770 LET U2=34
3790 GOTO 4070
3810 IF F3<9.3 GOTO 3870
3830 LET U2=(3024)/(F3-1.6)**2
3850 GOTO 4070
3870 IF F3<9 GOTO 3930
3890 LET U2=EXP(3*F3)/(2.6736E10)
3910 GOTO 4070
```

```
3930 IF F3<8 GOTO 3990
3950 LET U2=147/EXP(F3-7)
3970 GOTO 4070
3990 IF F3<1.2 GOTO 4050
4010 LET U2=54
4030 GOTO 4070
4050 LET U2=EXP(8*F3)/(273.43)
4070 LET U2=U2/100
4090 REM U2 IS FOR A (2 MM) SILICON FILTER
4110 IF F3<16 GOTO 4170
4130 LET U3=800/(F3-11)**2
4150 GOTO 4310
4170 IF F3<9 GOTO 4230
4190 LET U3=(52-2*(F3-9))
4210 GOTO 4310
4230 IF F3<2 GOTO 4290
4250 LET U3=34+2*F3
4270 GOTO 4310
4290 LET U3=EXP(15*F3)/(2.8122E11)
4310 LET U3=U3/100
4330 REM U3 IS FOR A (2 MM) GERMANIUM FILTER
4350 IF F3<2.1 GOTO 4410
4370 LET U4=4084/(F3**5)
4390 GOTO 4490
4410 IF F3<1 GOTO 4470
4430 LET U4=100*(F3**(0.424))/(1.3698)
4450 GOTO 4490
4470 LET U4=73*(F3**2)
4490 LET U4=U4/100
4510 REM U4 IS FOR A LEAD SULFIDE DETECTOR
4530 IF F3<.85 GOTO 4590
4550 LET U5=0
4570 GOTO 4970
4590 IF F3<.62 GOTO 4650
4610 LET U5=25.88/(10**(28*(F3-.619998)))
4630 GOTO 4970
4650 IF F3<.558 GOTO 4710
4670 LET U5=SIN(20.944*F3-3.71757)*100
4690 GOTO 4970
4710 IF F3<.555 GOTO 4770
4730 LET U5=100.2
4750 GOTO 4970
4770 IF F3<.49 GOTO 4830
4790 LET U5=(100)*SIN((20.944*(F3))-3.76993)
4810 GOTO 4970
4830 IF F3<.43 GOTO 4890
4850 LET U5=(20.8)/10**(20*(.49001-F3))
4870 GOTO 4970
4890 IF F3<.35 GOTO 4950
4910 LET U5=(13)/10**(50*(.46-F3))
4930 GOTO 4970
```

```
4950 LET U5=0
4970 LET U5=U5/100
4990 REM U5 IS THE VISUAL RESPONSE CURVE
5010 IF F3<.81 GOTO 5070
5030 LET U6=18.5/(F3**8)
5050 GOTO 5210
5070 IF F3<.57 GOTO 5130
5090 LET U6=(F3**2)/(0.006562)
5110 GOTO 5210
5130 IF F3<.3 GOTO 5190
5150 LET U6=(F3**16)/(2.5E-06)
5170 GOTO 5210
5190 LET U6=0
5210 LET U6=U6/100
5230 REM U6 IS THE RESPONSE OF A SILICON DETECTOR
5250 IF U6>(1.0E-08) GOTO 5310
5270 REM
5290 LET U6=0
5310 IF U5>(1.0E-08) GOTO 5350
5330 LET U5=0
5350 IF U4>(1.0E-08) GOTO 5390
5370 LET U4=0
5390 IF U3>(1.0E-08) GOTO 5430
5410 LET U3=0
5430 IF U2>(1.0E-08) GOTO 5470
5450 LET U2=0
5470 IF U1>(1.0E-08) GOTO 5510
5490 LET U1=0
5510 REM
5530 REM THESE SPACES ARE FOR ADDITIONAL TRANS. CALC. AND EXPANSIONS
5550 REM
5570 REM
5590 REM
5610 REM
5630 IF T5(D8)>1 GOTO 5690
5650 LET T6=U1
5670 GOTO 6130
5690 IF T5(D8)>2 GOTO 5750
5710 LET T6=U2
5730 GOTO 6130
5750 IF T5(D8)>3 GOTO 5810
5770 LET T6=U3
5790 GOTO 6130
5810 IF T5(D8)>4 GOTO 5870
5830 LET T6=U4
5850 GOTO 6130
5870 IF T5(D8)>5 GOTO 5930
5890 LET T6=U5
5910 GOTO 6130
5930 IF T5(D8)>6 GOTO 5990
5950 LET T6=U6
```

```
5970 GOTO 6130
5990 IF T5(D8)>7 GOTO 6050
6010 LET T6=U3*U4
6030 GOTO 6130
6050 IF T5(D8)>8 GOTO 6110
6070 LET T6=U1*U2*U4
6090 GOTO 6130
6110 LET T6=U1*U3*U4
6130 IF T5(D8)<=9 GOTO 6190
6150 LET T6=T5(D8)/100
6170 REM ##### THIS IS THE END OF THE FILTER TABLES
6190 LET W2=(2*P*(C**2)*H)
6210 LET W3=((F3**5)*K5)
6230 IF T6>(1.0E-08) GOTO 6270
6250 LET T6=0
6270 LET Z2=1.0E10
6290 LET W4=(W4+T6*(W2/W3)*Z2)
6310 LET W8=(W8+(W2/W3)*Z2)
6330 LET N8=N8+1
6350 IF N8<>100 GOTO 6550
6370 LET N8=0
6390 GOTO 6550
6410 REM ##### REMOVE THE ABOVE GOTO TO PRINT FROM THIS LOOP ########
6430 IF K6=1 GOTO 6490
6450 PRINT "F3","U1","U5","U6","U4"
6470 PRINT
6490 LET K6=1
6510 PRINT F3,U1,U5,U6;U4
6530 REM ##### THIS SPACE IS FOR LIST WITHIN THE X LOOP
6550 IF R9$="Y" GOTO 6610
6570 IF R9$="YES" GOTO 6610
6590 NEXT X
6610 LET W5=W4/Z2
6630 LET W6=W5*S*(1.0E09)
6650 REM W6 IS THE SPECTRAL RADIANT POWER IN (W/CM**2)
6670 LET W7=(W6*A7)/P
6690 REM W7 IS IN W/STR.
6710 LET D5=D3*W7/621
6730 REM D5 IS IN LUMENS / STR ONLY IF THE VISUAL CURVE IS USED
6750 IF T5(D8)=5 GOTO 6790
6770 LET D5=0
6790 LET O4=W6*L4/(H*C*(1.0E-07))
6810 REM O4 IS IN PHOTONS PER CM**3 , THE MONOCHRO. FLUX
6830 LET O5=O4*(1.0E-24)
6850 REM O5 IS PHOTONS /ANGSTROM **3
6870 LET V2=V1(D9)
6890 IF V2=0 GOTO 6950
6910 LET V3=C5*LOG(V2)
6930 GOTO 6970
6950 LET V3=0
6970 LET I2=I1(D9)
```

```
6990 IF I2=0 GOTO 7050
7010 LET I3=C5*LOG(I2)
7030 GOTO 7070
7050 LET I3=0
7070 LET W9=C5*LOG(W7)
7080 IF I3=0 GOTO 7150
7090 IF V3=0 GOTO 7150
7110 LET K8=C5*LOG(V2*I2)
7130 IF V3<>0 GOTO 7170
7140 IF I3=0 GOTO 7170
7150 LET K8=0
7170 LET O6=C5*LOG(O4)
7190 LET K3=C5*LOG(W1)
7210 LET G2=(F2(D8)-F1(D8))/2
7230 LET B6=L/(G2+F1(D8))
7250 LET B9=2
7270 IF B6<B9 GO TO 7310
7290 LET B6=B9
7310 LET B2=B6*(((1+B1)**4)-1)
7330 LET B5=B6*SQR(2*(B7**2))
7350 LET B3=SQR(B2**2+B5**2)
7370 LET B4=1+B3
7390 LET B8=100*B3
7410 REM B8 IS THE % ERROR OF W7
7430 IF R9$="Y" GOTO 11570
7450 IF R9$="YES" GOTO 11570
7470 PRINT
7490 PRINT
7510 IF N7=1 GOTO 7770
7530 PRINT"TOTAL POWER","EMITTANCE","SPECTRAL","TEMPERATURE","ERROR"
7550 PRINT"IN WATTS   ","WATTS/CM2","WATT/STR","IN DEGREES ","  %  "
7570 PRINT
7590 PRINT"RADIANT POWER","  ATTEN.","  V*I  ","  V*I/WATT","PHOTON/A3"
7610 PRINT" WATTS/CM2    ","    %  "
7630 PRINT
7650 PRINT" WAVELENGTH ","  VOLTS  ","  AMPS  "," WINDOW ","  MAX.  "
7670 PRINT"PEAK MICRONS","          ","        ","  MIN.  "," MICRONS"
7690 PRINT
7710 PRINT "L(V)/L(S)","L(I)/L(S)","L(IV)/L(S)","L(I)/L(P)","LUMEN/STR"
7730 PRINT
7750 PRINT
7770 PRINT W1;W;W7;T1(D9);R5$(D9);B8
7790 PRINT
7810 PRINT W6;100-100*W4/W8;I2*V2;I2*V2/W1;O5
7830 PRINT
7850 LET N7=1
7870 PRINT L;V1(D9);I1(D9);F1(D8);F2(D8)
7890 PRINT
7910 PRINT V3/W9;I3/W9;K8/W9;I3/O6;D5
7930 PRINT
7950 PRINT
```

```
7970 NEXT D8
7990 NEXT D9
8010 PRINT
8030 PRINT "DO YOU WANT TO PLOT DATA/VERSUS WAVELENGTH"
8050 INPUT R9$
8070 PRINT
8090 IF R9$="Y" GOTO 10350
8110 IF R9$="YES" GOTO 10350
8130 PRINT "DO YOU WANT TO RUN MORE DATA "
8150 LET N9=0
8170 INPUT M$
8190 PRINT
8210 IF M$="Y" GOTO 870
8230 IF M$="YES" GOTO 870
8250 GOTO 10310
8270 PRINT
8290 PRINT
8310 PRINT "HOW MANY NEW DISTANCES DO YOU WISH TO CALCULATE"
8330 PRINT
8350 INPUT N3
8370 PRINT
8390 PRINT"INPUT THE NEW DISTANCE AND UNITS (IN), (FT), (CM), (M)"
8410 PRINT
8430 FOR J=1 TO N3
8450 INPUT R3(J), R3$(J)
8470 NEXT J
8490 PRINT
8510 PRINT"HOW MANY POWER DATA POINTS DO YOU HAVE. 50 IS MAX."
8530 PRINT
8550 INPUT P4
8570 PRINT
8590 PRINT"ENTER WATTS/(FT2),(CM2),(M2),(STR). AT DIST. (CM),(M),(IN),(FT)"
8610 PRINT
8630 FOR J1=1 TO P4
8650 INPUT P5(J1), P5$(J1), R1(J1), R1$(J1)
8670 NEXT J1
8690 PRINT
8710 PRINT"INPUT (X) IN 1/R**X. FOR YOUR DISTANCE DEPENDENCY"
8730 PRINT
8750 INPUT P8
8770 PRINT
8790 LET P3=0
8810 FOR J=1 TO N3
8830 IF R3$(J)="IN" GOTO 8930
8850 IF R3$(J)="FT" GOTO 8970
8870 IF R3$(J)="M" GOTO 9010
8890 IF R3$(J)="CM" GOTO 9050
8910 GOTO 8370
8930 LET R3=R3(J)*2.54
8950 GOTO 9070
8970 LET R3=(R3(J)*(2.54)*12)
```

```
8990 GOTO 9070
9010 LET R3=100*R3(J)
9030 GOTO 9070
9050 LET R3=R3(J)
9070 LET P=3.14159
9090 LET R8=1/(R3**P8)
9110 FOR J1=1 TO P4
9130 IF R1$(J1)="CM" GOTO 9290
9150 IF R1$(J1)="IN" GOTO 9330
9170 IF R1$(J1)="M" GOTO 9370
9190 IF R1$(J1)="FT" GOTO 9250
9210 IF P5$(J1)="STR" GOTO 9290
9230 GOTO 8590
9250 LET V9=R1(J1)*(2.54)*12
9270 GOTO 9390
9290 LET V9=R1(J1)
9310 GOTO 9390
9330 LET V9=R1(J1)*2.54
9350 GOTO 9390
9370 LET V9=R1(J1)*100
9390 LET V2=V9**P8
9410 IF P5$(J1)="FT2" GOTO 9510
9430 IF P5$(J1)="M2" GOTO 9550
9450 IF P5$(J1)="CM2" GOTO 9590
9470 IF P5$(J1)="STR" GOTO 9630
9490 GOTO 8570
9510 LET P5=P5(J1)/(929.034)
9530 GOTO 9650
9550 LET P5=P5(J1)/10000
9570 GOTO 9650
9590 LET P5=P5(J1)
9610 GOTO 9650
9630 LET P5=P5(J1)
9650 IF P5$(J1)="STR" GOTO 9710
9670 LET V3=P5*V2
9690 GOTO 9730
9710 LET V3=P5
9730 LET V4=V3*R8
9750 LET D6=(V4*(929.034))/621
9770 REM D6 IS LUMENS / FT**2
9790 LET V5=V4*10000
9810 LET O1=10*C5*LOG(1000*V4)
9830 LET O9=10*C5*LOG(1000*V5)
9850 IF P3=1 GOTO 9970
9870 PRINT
9890 PRINT "LUMEN/FT2 "," INPUT WATTS ","WATT/CM2","WATT/M2","1/R**%"
9910 PRINT
9930 PRINT"NEW DIST","MW/CM2","MW/M2","DBM/CM2","DBM/M2"
9940 PRINT
9950 PRINT
9960 PRINT "*********************************************************************************"
```

```
9970 PRINT
9990 PRINT D6, P5(J1);P5$(J1); V4,V5, P8
10010 PRINT
10030 PRINT R3(J);R3$(J);1000*V4,1000*V5,01,09
10050 PRINT
10070 PRINT
10090 LET P3=1
10110 NEXT J1
10120 PRINT "------------------------------------------------------------------------"
10130 NEXT J
10150 PRINT
10170 PRINT
10190 PRINT
10210 PRINT "DO YOU WANT TO START AGAIN"
10230 INPUT M5$
10250 PRINT
10270 IF M5$="Y" GOTO 870
10290 IF M5$="YES" GOTO 870
10310 STOP
10330 REM THIS IS PLOTPTS
10350 REM THIS PROGRAM PLOTS POINTS IN A BAR TYPE GRAPH
10370 REM INPUT MAY BE FROM FILE NAMED SCR7 BY CHANGING
10390 REM THE #0 IN THE INPUT STATEMENT TO #1
10410 REM OTHERWISE THE INPUT IS FROM THE KEYBOARD.
10430 DIM Y(256), S(256)
10450 PRINT"IF YOU WANT A LINEAR PLOT TYPE (1), OR (2) LOG PLOT"
10470 PRINT
10490 INPUT P1
10510 FILES SCR7
10530 PRINT
10550 PRINT"INPUT WINDOW LIMITS (MIN), (MAX), STEP (X). IN MICRONS"
10570 PRINT
10590 INPUT X1,X2,X3
10610 PRINT
10630 PRINT
10650 IF P1=2 GO TO 10790
10670 PRINT " INPUT (Y) LIMITS, (MIN.) AND (MAX.)"
10690 PRINT
10710 INPUT Q1,Q2
10730 LET M=50/(Q2-Q1)
10750 LET Q3=(Q2-Q1)/50
10770 IF P1=1 GO TO 11010
10790 PRINT " INPUT LOG(Y) LIMITS, (MIN)=10**N AND (MAX)=10**N1"
10810 PRINT " REM YOU CAN'T CROSS ZERO, AND DON'T FORGET THE SIGN"
10830 PRINT
10850 PRINT
10870 INPUT Q1,Q2
10890 LET M=50/(Q2-Q1)
10900 LET Q3=(Q2-Q1)/50
10910 PRINT
10930 PRINT "TYPE IN THE SCALE FACTOR TO INSURE NO ZERO CROSSING"
```

```
10950 PRINT "WHERE S=(10**N2). EX. S=1; OR 10; OR 100; ........"
10970 PRINT
10990 INPUT S8
11010 PRINT
11030 PRINT
11050 PRINT
11070 PRINT"ENTER YOUR (Y) EQ. # <=9.IT IS TO BE ENTERED BEFORE RUN"
11090 PRINT
11110 INPUT H6
11130 IF H6>9 GOTO 11050
11150 LET L7=1+((X2-X1)/X3)
11170 PRINT
11190 PRINT
11210 IF P1=1 GO TO 11270
11230 LET S$=" ( LOG PLOT ), (SCALE FACTOR)"
11250 IF P1=2 GO TO 11290
11270 LET S$=" (LINEAR PLOT), (SCALE FACTOR)"
11290 PRINT
11310 PRINT
11330 PRINT "X",TAB(9);Q1;TAB(28);S$;TAB(62);Q2
11350 PRINT
11370 PRINT "  ##  TEMPERATURE "; T1(D9);R5$(D9)
11390 PRINT
11410 PRINT "(INCREMENTS)",TAB(28);"DELTA Y =";TAB(32);Q3;TAB(49);S8
11430 PRINT
11450 PRINT TAB(17);"I----+----I----+----I----+----I----+----I----+----IQ"
11470 FOR A=1 TO L7 STEP 1
11490 LET F3=(X1+X3*(A-1))
11510 IF F3>0 GOTO 11550
11530 LET F3=S
11550 GOTO 3170
11570 GOTO 11590
11590 IF P1=2 GO TO 12090
11610 LET J8=U1
11630 IF H6=1 GOTO 11970
11650 LET J8=U2
11670 IF H6=2 GOTO 11970
11690 LET J8=U3
11710 IF H6=3 GOTO 11970
11730 LET J8=U4
11750 IF H6=4 GOTO 11970
11770 LET J8=U5
11790 IF H6=5 GOTO 11970
11810 LET J8=U6
11830 IF H6=6 GOTO 11970
11850 LET J8=U3*U4
11870 IF H6=7 GOTO 11970
11890 LET J8=U1*U2*U4
11910 IF H6=8 GOTO 11970
11930 LET J8=U1*U3*U4
11950 IF H6=9 GOTO 11970
```

```
11970 IF J8>Q1 GO TO 12010
11990 LET J8=Q1
12010 IF J8<Q2 GO TO 12050
12030 LET J8=Q2
12050 LET Z1=M*(J8-Q1)
12070 IF P1=1 GO TO 12310
12090 LET C1=0.434294
12110 LET B1=J8/S8
12130 IF B1=0 GO TO 12190
12150 LET L1=C1*LOG(B1)
12170 GO TO 12210
12190 LET L1=Q1
12210 IF L1<Q2 GO TO 12250
12230 LET L1=Q2
12250 IF L1>Q1 GO TO 12290
12270 LET L1=Q1
12290 LET Z1=M*(L1-Q1)
12310 LET Z=(1.5)+Z1
12330 LET M3=X3
12350 LET X4=(X1+M3*(A-1))
12370 PRINT X4,TAB(16);
12390 FOR G=1 TO Z
12410 PRINT TAB(17);"X";
12430 NEXT G
12450 PRINT TAB(68);"I"
12470 NEXT A
12490 PRINT TAB(17);"I----+----I----+----I----+----I----+----I----+----IO"
12510 PRINT
12530 PRINT
12550 PRINT
12570 PRINT "DO YOU WANT ANOTHER PLOT ?"
12590 PRINT
12610 INPUT C$
12630 PRINT
12650 PRINT
12670 PRINT
12690 IF C$="Y" GO TO 10450
12710 IF C$="YES" GO TO 10450
12730 GOTO 8130
12750 STOP
12770 END
```

Sample Run of Optical

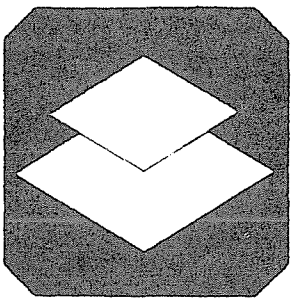
Example #1

This example will illustrate the conversion section of this program. Three light sources will be evaluated at two distances, 50 meters and 10 inches. The first source produces 0.00685 watts/steradians, the second one is 100 watts/meter squared when 50 cm from it and the third is 1 watts/meter sq. when viewed at 25 meters. The 1/r squared dependency is used to determine how the intensity drops off with respect to distance. The converted intensities are tabulated on the second example sheet for the six conditions.

Example #2

In the second example we shall calculate the intensity of a #47 light bulb that stimulates a lead sulfide detector. The detector has a germanium filter placed in front of it and the bulb has a current of 100 mA flowing through it. The results of the calculations are shown on the fourth example page. The temperature of the bulb at 100 mA was measured at 1432 degrees C. The calculations were done over 1.5 to 3.5 microns to reduce calculation time. This will induce almost no error for Germanium doesn't transmit light below 1.6 microns and the bulb has a glass cover that becomes opaque above 3 microns.

The plot on the last page represents the sensitivity of a silicon detector. The plot is from Ø microns to 1.45 microns and the plot is normalized from Ø to 1, left to right. Eight other similar curves are contained in the program.



```
 RUN


THIS PROGRAM CALCULATES BLACKBODY PROBLEMS
IT IS VERY FLEXIBLE AND MAKES NO ASSUMPTIONS
BE CAREFUL TO ENTER ALL UNITS AS INDICATED

#### REMEMBER USE WITH THERMAL RADIATORS ONLY ####



DO YOU WANT CONVERSIONS (1), OR BLACKBODY CALC. (2)

?1


HOW MANY NEW DISTANCES DO YOU WISH TO CALCULATE

?2

INPUT THE NEW DISTANCE AND UNITS (IN), (FT), (CM), (M)

?50,M
?10,IN

HOW MANY POWER DATA POINTS DO YOU HAVE. 50 IS MAX.

?3

ENTER WATTS/(FT2),(CM2),(M2),(STR). AT DIST. (CM),(M),(IN),(FT)

?.00685,STR,0,0
?100,M2,50,CM
?1,M2,25,M

INPUT (X) IN 1/R**X. FOR YOUR DISTANCE DEPENDENCY

?2
```

```
LUMEN/FT2          INPUT WATTS    WATT/CM2          WATT/M2          1/R**X
NEW DIST         MW/CM2           MW/M2           DBM/CM2          DBM/M2

**************************************************************************

     4.09912E-10   .00685 STR      2.74000E-10      2.74000E-06              2
   50 M      2.74000E-07        .00274        -65.6225        -25.6225

     1.49603E-06   100 M2       1.00000E-06          .01                  2
   50 M          .001              10              -30               10

     3.74007E-05     1 M2       2.50000E-05          .25                  2
   50 M          .025             250         -16.0206          23.9794

--------------------------------------------------------------------------

     1.58841E-05   .00685 STR      1.06175E-05     .1061752                 2
   10 IN      .0106175         106.1752        -19.73977         20.26023

    .0579712       100 M2      .0387501          387.5008                 2
   10 IN      38.75008         387500.8         15.88273          55.88273

    1.449281         1 M2      .9687519          9687.519                 2
   10 IN      968.7519          9687519         29.86213          69.86213

--------------------------------------------------------------------------


DO YOU WANT TO START AGAIN
?N
```

```
THIS PROGRAM CALCULATES BLACKBODY PROBLEMS
IT IS VERY FLEXIBLE AND MAKES NO ASSUMPTIONS
BE CAREFUL TO ENTER ALL UNITS AS INDICATED

#### REMEMBER USE WITH THERMAL RADIATORS ONLY ####



DO YOU WANT CONVERSIONS (1), OR BLACKBODY CALC. (2)

?2

HOW MANY TEMPERATURES DO YOU WANT TO RUN. 10 IS MAX.

?1

DO YOU WANT TO INPUT THE APERTURE AREA? ELSE (.0065)
?N

DO YOU WISH TO INPUT VOLTAGE AND CURRENT
?Y

INPUT TEMPERATURE (DEGREES), SCALE (R), (F), (C), (K)
AND (VOLTS), THEN (AMPS)

?1432,C,1.397,.1004

INPUT THE NUMBER OF SPECTRAL WINDOWS YOU WANT CALCULATED

?1

ENTER END POINT (MIN), (MAX) MICRONS AND TRANS. IN %
** THE (%) SHOULD BE >9.  TYPE (0) FOR SPECIAL LIST

?1.5,3.5,9
```

The following abreviations were used in this listing. They are as follows:

V = Volts I = Amps A3 = Cubic Anstroms Window Min = Low Freq. Cutoff in Microns
Max Microns = High Freq. Cutoff of the Filter, if any, in microns
L( ) = log ( ) P = Photons STR = Steradians

TOTAL POWER IN WATTS	EMITTANCE !WATTS/CM2	SPECTRAL WATT/STR	TEMPERATURE IN DEGREES	ERROR %
RADIANT POWER WATTS/CM2	ATTEN. %	!/*!	!*I/WATT	PHOTON/A3
WAVELENGTH PEAK MICRONS	VOLTS	AMPS	WINDOW MIN.	MAX. MICRONS
L(V)/L(S)	L(I)/L(S)	L(IV)/L(S)	L(I)/L(P)	LUMEN/STR
.2016759	47.9244	.0068507	1432 C	20.26794
3.311108	87.69613	.1402588	.6954665	.3587969
1.699469	1.397	.1004	1.5	3.5
-.0670382	.45125	.3941619	-.0430805	0

DO YOU WANT TO PLOT DATA/VERSUS WAVELENGTH

(THIS IS A SILICON DETECTOR'S RESPONSE AS DETERMINED BY THE LOOKUP TABLE)

IF YOU WANT A LINEAR PLOT TYPE (1), OR (2) LOG PLOT

?1

INPUT WINDOW LIMITS (MIN), (MAX), STEP (X), IN MICRONS

?0,1.5,.05

INPUT (Y) LIMITS, (MIN) AND (MAX)

?0,1

ENTER YOUR (Y) EQ. NUMBER. IT IS ENTERED BEFORE RUN

?6

```
X                    0          (LINEAR PLOT), (SCALE FACTOR)

  ##  TEMPERATURE                 1999 K

(INCREMENTS)                 DELTA Y =  .02            0

                  I-----+-----I-----+-----I-----+-----I-----+-----I-----+-----I0
              0   X                                                           I
            .05   X                                                           I
             .1   X                                                           I
            .15   X                                                           I
             .2   X                                                           I
            .25   X                                                           I
             .3   X                                                           I
            .35   X                                                           I
             .4   X                                                           I
            .45   XX                                                          I
             .5   XXXX                                                        I
            .55   XXXXXXXXXXXXXXXX                                            I
             .6   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            .65   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
             .7   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            .75   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
             .8   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            .85   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
             .9   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            .95   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
              1   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
           1.05   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            1.1   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
           1.15   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            1.2   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
           1.25   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            1.3   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
           1.35   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
            1.4   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXI
           1.45   X                                                           I
                  I-----+-----I-----+-----I-----+-----I-----+-----I-----+-----I0
```

DO YOU WANT ANOTHER PLOT ?

?Y

PLANET:

DESCRIPTION

Planet is a very interesting program. It will list the Altitude and Azimuth of the Sun, Moon and the four (4) brightest planets every hour, for a 24 hr. period; for a specific latitude and longitude.

USERS

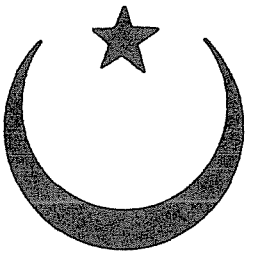
Amateur astronomers would be greatly aided by this program. Anyone who plans to do navigation at night could also make good use of this program.

INSTRUCTIONS

The GHA and DEC for the six (6) bodies are entered for Ø hours, for the day of interest in lines 110 and 120. These values are obtainable from a nautical almanac. The program is then run and all additional data is automatically prompted. List the program for detailed program information.

LIMITATIONS

Line 690 contains a Restore statement. This program executes in standard Basic. The source code is 5K Bytes long and the program will execute in 8K Bytes in most systems.



PLANET

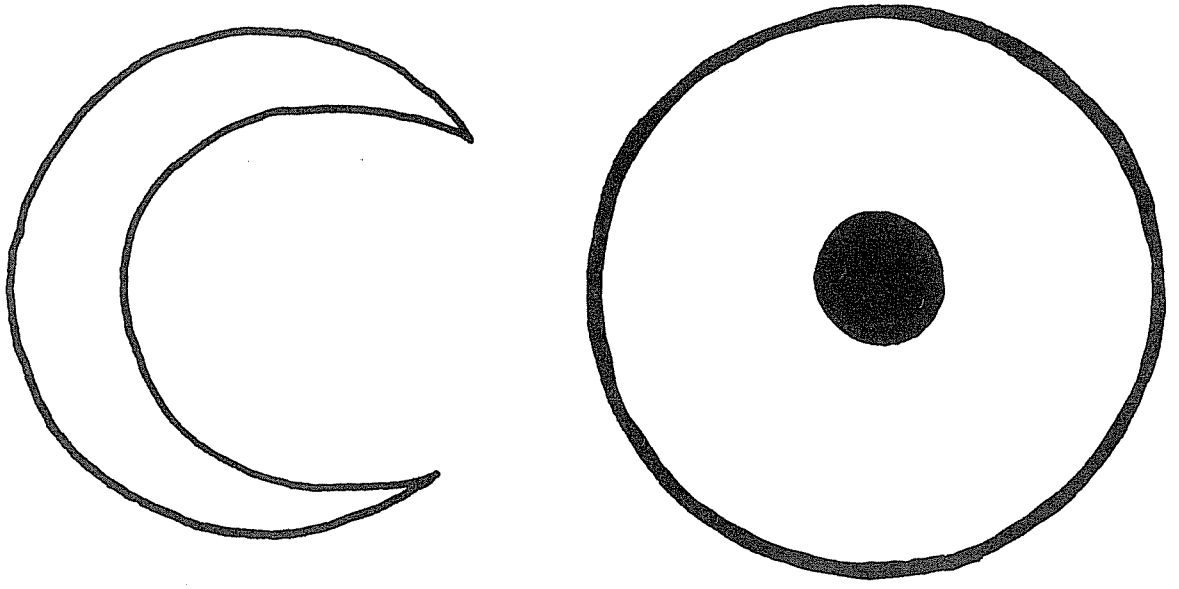
```
100 PRINT " 16  APRIL 75  GMT 24"
110 DATA 180,10.2,143.2,22.3,228.1,-11.3
120 DATA 197.6,1.8,100.4,22.6,115.9,21
130 PRINT
140  PRINT
150 PRINT "SUN MOON AND PLANETS LOCATOR"
160 PRINT "LIST LINES 140 TO 230 FOR DATA DESCRIPTIONS"
170 REM TWELVE NUMBERS IN DATA LINE
180 REM NUMBERS ARE GHA & DEC OF
190 REM VENUS MARS JUPITER SATURN SUN & MOON
200 REM DATA LINE IS 110
210 REM ONLY WHOLE DEGREES USED
220 REM DATA IS TAKEN AT GMT
230 PRINT "GMT IS IN SINGLE HOUR"
240 PRINT "WEST LONGITUDES - "
250 PRINT " SOUTH LATITUDES -"
260 PRINT "DATA ORDER IS SUN VENUS MARS JUPITER SATURN MOON"
270 PRINT "GMT IS TIME AT 0 MERIDIAN WHEN FIRST PRINT OCCURS"
275 Y=57.2957795
277 Y1=1.57079632
280 PRINT
290 PRINT "L IS LATITUDE"
300 PRINT "A7 IS LONGITUDE"
310 PRINT " T9 IS GMT"
320 PRINT "D7 IS DAY OF MONTH"
330 PRINT " L,A7,T9,D7"
340 INPUT L,A7,T9,D7
350  Z=ABS(A7/15)
360 Z1=INT(Z)
370 Z2=Z-Z1
380 IF Z2>.5 GOTO 400
390 GOTO 410
400 Z1=Z1+1
410 IF A7<0 GOTO 430
420 GOTO 440
430 Z1=-Z1
440 T9=T9+Z1+1
450 IF T9<0 GOTO 470
460 GOTO 490
470 T9=T9+24
480 D7=D7-1
```

```
490 T9=T9*100
500 IF T9>2400 GOTO 520
510 GOTO 540
520 T9=T9-2400
530 D7=D7+1
540 J7=L
550 PRINT
560 PRINT
570 PRINT "  BODY        ALTITUDE DEGREES      AZIMUTH DEG"
580 PRINT "-------------------------------------------------------------"
590 IF O>14 GOTO 690
600 READ A
610 C2=A+A7
620 READ D
630 W=W+1
640 IF W=6 GOTO 660
650 GOTO 680
660 O=T*14.9916
670 GOTO 770
680 GOTO 780
690 RESTORE
700 W=0
710 READ A
720 READ D
730 W=W+1
740 IF W=6 GOTO 760
750 GOTO 770
760 O=T*14.49
770 C2=A+O+A7
780 C3=D/Y
790 O=T*15
800 IF C2>1080 THEN 1000
810 IF C2>900 THEN 1020
820 IF C2>720 GOTO 880
830 IF C2>540 GOTO 900
840 IF C2>360 GOTO 920
850 IF C2>180 GOTO 940
860 IF C2<180 THEN 960
870 IF C2<0 THEN 980
880  X2=720-C2
890 GOTO 1030
900  X2=720-C2
910 GOTO 1030
920  X2=360-C2
930 GOTO 1030
940  X2=360-C2
950 GOTO 1030
960  X2=-C2
970 GOTO 1030
980  X2=-C2
990 GOTO 1030
```

```
1000  X2=1080-C2
1010 GOTO 1030
1020  X2=1080-C2
1030  X4=ABS(X2)/Y
1040 IF X4>Y1 THEN 1060
1050 IF X4<Y1 THEN 1080
1060  X3=2*Y1 -X4
1070 GOTO 1090
1080  X3=X4
1090  B=ABS (X3)
1100  B1=C3
1110  B3=J7/Y
1120  B=X3
1130  P=ATN((SIN(B)*COS(B1))/(1-(SIN(B)*COS(B1))↑2)↑.5)
1140  K=ATN((SIN(B1)/COS(P))/(1-(SIN(B1)/COS(P))↑2)↑.5)
1150  K6=ABS(K)
1160  K1=ABS(K)*Y
1170 IF X4>Y1 THEN 1250
1190 IF X4<Y1 THEN 1190
1190 IF K1>90 THEN 1210
1200 IF K1<90 THEN 1230
1210  K2=180-K1
1220 GOTO 1300
1230  K2=K1
1240 GOTO 1300
1250 IF K1>90 THEN 1270
1260 IF K1<90 THEN 1290
1270  K2=K1
1280 GOTO 1300
1290  K2=180-K1
1300 IF K>0 THEN 1320
1310 IF K<0 THEN 1340
1320  K3=K2
1330 GOTO 1350
1340  K3=-K2
1350  K4=K3/Y
1360  B4=-B3
1370  G1=K4+B4
1380  G2=ABS(G1)
1390  G5=G2*Y
1400 H=ATN((COS(P)*COS(G2))/(1-(COS(P)*COS(G2))↑2)↑.5)
1410  H1=H*Y
1420  H3=INT(H1)
1430  H4=H1-H3
1440  H5=H4*60
1450 IF X4>Y1 THEN 1470
1460 GOTO 1480
1470  H=-H
1480  P5=P
1490  Z=ATN((SIN(P5)/COS(H))/(1-(SIN(P5)/COS(H))↑2)↑.5)
1510  Z1=Z*Y
```

```
1520  D3=K/B3
1530  B7=ABS(B3)
1540  K8=K2/Y
1550 IF D3>0 THEN 1570
1560 IF D3<0 THEN 1590
1570 IF K8>B7 THEN 1610
1580 GOTO 1590
1590 IF Z1>90 THEN 1630
1600 IF Z1<90 THEN 1650
1610 IF Z1>90 THEN 1650
1620 IF Z1<90 THEN 1630
1630  Q1=Z1
1640 GOTO 1660
1650  Q1=180-Z1
1660 IF B3>0THEN 1680
1670 IF B3<0 THEN1740
1680 IF X2>0 THEN1700
1690 IF X2<0THEN 1720
1700  D5=Q1
1710 GOTO 1790
1720  D5=360-Q1
1730 GOTO 1790
1740 IF X2>0THEN 1760
1750 IF X2<0THEN 1780
1760  D5=180-Q1
1770 GOTO 1790
1780  D5=180+Q1
1790 GOTO 1800
1800  D6=INT(D5)
1810 IF D6>360 THEN 1830
1820 GOTO 1840
1830  D6=D6-360
1840 IF W=1 GOTO 1900
1850 IF W=2 GOTO 2020
1860 IF W=3 GOTO 2060
1870 IF W=4 GOTO 2120
1880 IF W=5 GOTO 2170
1890 IF W=6 GOTO 2220
1900 PRINT
1910 PRINT
1920 PRINT "DLST";T9"DAY";D7
1930 PRINT
1940 L8=0
1950 IF H3<0 GOTO 2000
1960 PRINT "SUN";H3;D6
1970 IF H3>8 GOTO 1990
1980 GOTO 2000
1990 L8=1
2000 IF O>14 GOTO 710
2010 GOTO 590
2020 IF H3<0 GOTO 2040
```

```
2030 PRINT "VENUS",H3,D6
2040 IF O>14 GOTO 710
2050 GOTO 590
2060 IF H3<0 GOTO 2090
2070 IF L8=1 GOTO 2090
2080 PRINT "MARS",H3,D6
2090 IF O>14 GOTO 710
2100 GOTO 590
2110 IF O>14 GOTO 710
2120 IF H3<0 GOTO 2150
2130 IF L8=1 GOTO 2150
2140 PRINT "JUPITER",H3,D6
2150 IF O>14 GOTO 710
2160 GOTO 590
2170 IF H3<0 GOTO 2200
2180 IF L8=1 GOTO 2200
2190 PRINT "SATURN",H3,D6
2200 IF O>14 GOTO 710
2210 GOTO 590
2220 IF H3<0 GOTO 2240
2230 PRINT "MOON",H3,D6
2240 O=O+15
2250 T=T+1
2260 T9=T9+100
2270 IF T9=>2400 GOTO 2290
2280 GOTO 2310
2290 T9=T9-2400
2300 D7=D7+1
2310 IF O>345 GOTO 2330
2320 GOTO 590
2330 END
```



PSD:

DESCRIPTION

PSD stands for Power Spectral Density and that is what this program calculates. This program is extremely versatile as it allows the use of sliding windows, windowing and bit integrating on one or both of the windows. One of the most unique features of this program (not found in private industry) is it's ability to compute two windows simultaneously without Any loss in accuracy. This program is very compact and efficient and can only be enhanced by the addition of file input and output statements. This program computes FFT transforms and inverse transforms as well, thereby making it useable for computing high speed convolutions of complex systems.

USERS

Anyone who needs or desires to examine data in the frequency domain will have use for this program. The dedicated audiophile who wants his equipment to sound good is one example of a user. Noise pollution can easily be examined in the frequency domain with the emphasis on reducing noise, selectively. Other possible users could be persons interested in modeling analog systems of things such as: Missiles, or aircraft, or marine vechiles, or even automobiles. The auto mechanic can use this program to look for troubles before they happen. For example, if the noise of a car engine is viewed in the frequency domain the various interengine components can be seen; if one of them is starting to wear out it's frequency components will shift well before failure can be detected by other means.

INSTRUCTIONS

The program is self contained and will prompt the user for all required information. The program can be listed if additional program information is desired.

LIMITATIONS

The source code is 5K Bytes long. Execution length is a function of the size of the matrix set in the DIM statement in line 190. With the DIM set at 256 points, execution will require 25K Bytes in most systems.

PSD

```
100 REM THIS IS     PSD
110 REM THIS IS IN BASIC
120 REM
130 REM
140 REM IT IS A POWER SPECTRAL DENSITY PROGRAM.
150 REM IF S1 IS 0 IT DOES THE TIME DOMAIN TO FREQ. TRANSFORM
160 REM IF S1 IS 1 IT DOES THE REVERSE TRANSFORM.
170 REM INPUT SHOULD BE SYMM. ABOUT THE PT. (1+N/2)
180 REM FOR FURTHER INSTRUCTIONS SEE FILE
190 DIM X(256),Y(256),S(256),T(256),U(192),Z(512)
200 PRINT
210 PRINT "THIS PROGRAM CALCULATES POWER SPECTRAL DENSITIES"
220 PRINT "AND FAST FOURIER TRANSFORMS."
230 PRINT "IT USES TWO SLIDING WINDOWS AND AVERAGING & WINDOWING."
240 PRINT
250 PRINT
260 MAT X=ZER
270 MAT Y=ZER
280 MAT Z=ZER
290 PRINT
300 PRINT "HOW MANY DATA POINTS DO YOU HAVE?"
310 INPUT N
320 PRINT
330 PRINT "ENTER YOUR DATA "     (X & Y matrix; if both differ put X first)
340 FOR I=1 TO N                 (followed by the Y data.)
350 INPUT Z(I)
360 NEXT I
370 PRINT
380 PRINT "INPUT THE NUMBER OF BITS PER WINDOW."(X & Y matrix are the same size)
390 PRINT "THEN TYPE A (1), FOR   AVERAGES"
400 PRINT "OR A (2), FOR NO AVERAGING." (the X & Y matrix are averaged together)
410 INPUT A,G9
420 S1=0
430 PRINT
440 C1=.5
450 T1=1
460 REM HANN WINDOW C1=.5, T1=1. HAMMING WIND. C1=.46, T1=1.17
470 A2=LOG(A)
480 A3=A2/(LOG(2))
490 A4=INT(A3+.1)
500 REM A5 IS THE # OF SAMPLES TAKEN PER SEC.
```

```
510 REM UNLESS SPECIFIED ELSEWHERE
520 A6=A/2
530 T=1/A
540 REM T IS THE SAMPLE RATE IN SEC.
550 REM T < (F*.5)
560 REM WHERE F IS THE LARGEST FREQ. PRESENT IN THE TIME DATA
570 F6=A*T
580 F1=A6/F6
590 A9=(1+(A/2))
600 PRINT
610 PRINT "INPUT THE STARTING BIT AND # OF BITS TO BE SKIPPED."
620 INPUT S2,S3
630 E=(S3*(A-1)+S2)
640 P=3.14159
650 W2=(A-1)
660 FOR I=1 TO A
670 W4=(S2+(I-1)*S3)
680 X(I)=Z(W4)
690 NEXT I
700 REM THIS IS THE X MATRIX INPUT
710 PRINT
720 PRINT "DO YOU WANT WINDOWING ON THE DATA?(1 FOR YES, 2 FOR NO)"
730 INPUT W1                          (For the X matrix ONLY)
740 IF W1=2 GOTO 780
750 FOR I=1 TO A
760 X(I)=X(I)*C1*(T1+SIN(P*(I-1)/W2))
770 NEXT I
780 PRINT
790 PRINT
800 PRINT "HOW MANY BITS DO YOU WANT TO SLIDE THE Y MATRIX?"
810 INPUT P1
820 FOR I=1 TO A
830 W4=(S2+(I-1)*S3)
840 Y(I)=Z(W4+P1)
850 NEXT I
860 GOSUB 1250
870 IF S1>0 GOTO 1150
880 IF (G9-1)=0 GOTO 1020
890 N1=0
900 PRINT
910 PRINT "   PSD       (#-1)        X MATRIX    Y MATRIX  START  +SLID"
920 PRINT
930 N2=0
940 FOR I=1 TO A9
950 A5=Y(I)
960 A7=X(I)
970 N3=INT(S2+.5+(S3*(I-1)))
980 N4=INT(N3+P1+.5)
990 PRINT    (I-1),A7,A5,N3;N4
1000 NEXT I
1010 GOTO 1150
```

```
1020 N1=0
1030 PRINT
1040 PRINT "                          TWO WINDOW AVERAGE"
1050 PRINT
1060 PRINT "    PSD    (#-1)       X MATRIX    START    +SLID"
1070 PRINT
1080 N2=0
1090 FOR I=1 TO A9
1100 A7=(X(I)+Y(I))/2
1110 N3=INT(S2+.5+(S3*(I-1)))
1120 N4=INT(N3+P1+.5)
1130 PRINT    (I-1),A7,N3,N4
1140 NEXT I
1150 PRINT
1160 PRINT "IF YOU WISH TO REUSE YOUR OLD DATA TYPE A (1), ELSE (2)."
1170 INPUT W6
1180 IF W6=2 GOTO 1200
1190 GOTO 370
1200 PRINT
1210 PRINT "TO ENTER NEW DATA TYPE A (1), TO STOP TYPE A (2)."
1220 INPUT W6
1230 IF W6=1 GOTO 230
1240 GOTO 2440
1250 REM THIS IS THE DOUBLE SUB
1260 A9=(1+(A/2))
1270 GOSUB 1390
1280 FOR K=2 TO A9
1290 L=(2+A)-K
1300 A8=X(K)+X(L)
1310 B=X(K)-X(L)
1320 C=Y(K)+Y(L)
1330 D=Y(K)-Y(L)
1340 X(K)=(A8*A8+D*D)*(.25)
1350 Y(K)=(C*C+B*B)*(.25)
1360 NEXT K
1370 REM X(1) IS THE D.C. LEVEL.
1380 RETURN
1390 REM THIS IS THE FFT SUB.
1400 REM S1 IS THE TRANSFORM/INV.TRANS. FLAG
1410 REM S1=0 FOR TRANSFORM & =1 FOR INV.TRANS.
1420 REM X() IS THE REAL & Y() IS THE IMAGINARY MATRIX.
1430 REM U() IS THE COSINE TABLE.
1440 L1=0
1450 IF S1=0 GOTO 1480
1460 S8=6
1470 GOTO 1490
1480 S8=7
1490 IF (A4-L1)=0 GOTO 1750
1500 L1=A4
1510 N5=(2**A4)/2
1515 H1=2*N5
```

```
1520 N6=(1+N5)
1530 N7=(N6+1)
1540 F=H1
1550 P2=(2*P)/F
1560 N8=N5+1
1570 N9=N8+1
1580 M=INT((H1/4)+.1)
1590 M1=M+1
1600 M2=((3*M)+1)
1610 U(1)=1
1620 U(M1)=0
1630 U(N8)=-1
1640 U(M2)=0
1650 FOR I=2 TO M
1660 F2=I-1
1670 T3=F2*P2
1680 T3=COS(T3)
1690 U(I)=T3
1700 I1=I+N5
1710 I2=N9-I
1720 U(I1)=-T3
1730 U(I2)=-T3
1740 NEXT I
1750 L=1
1760 FOR J=1 TO A4
1770 I5=L
1780 L=2*L
1790 G1=H1/L
1800 G2=G1
1810 FOR I=1 TO I5
1820 G3=(I-1)*G1
1830 G4=G3+1
1840 G5=G4+M
1850 W1=U(G4)
1860 IF S8=7 GOTO 1890
1870 W2=-U(G5)
1880 GOTO 1900
1890 W2=U(G5)
1900 FOR K=1 TO G2
1910 I1=K+G3
1920 I2=I1+G3
1930 I3=I2+G1
1940 I4=I1+N5
1950 G6=W1*X(I3)-W2*Y(I3)
1960 REM G6 IS THE REAL AND G7 IS THE IMAGINARY MATRIX.
1970 G7=W2*X(I3)+W1*Y(I3)
1980 S(I1)=X(I2)+G6
1990 T(I1)=Y(I2)+G7
2000 S(I4)=X(I2)-G6
2010 T(I4)=Y(I2)-G7
2020 NEXT K
```

```
2030 NEXT I
2040 FOR K=1 TO H1
2050 X(K)=S(K)
2060 Y(K)=T(K)
2070 NEXT K
2080 NEXT J
2090 IF S1=0 GOTO 2140
2100 FOR K=1 TO H1
2110 X(K)=X(K)/2
2120 Y(K)=Y(K)/2
2130 NEXT K
2140 IF S1=1 GOTO 2190
2150 FOR K=1 TO H1
2160 X(K)=X(K)/N5
2170 Y(K)=Y(K)/N5
2180 NEXT K
2190 PRINT
2200 PRINT
2210 PRINT "FREQUENCY BANDWIDTH = ";F1;" HERTZ."
2220 PRINT
2230 PRINT "FREQUENCY PER BIT   = ";F1/N5;" HERTZ."
2240 PRINT
2250 IF S1=1 GOTO 2310
2260 PRINT
2270 PRINT "TYPE A (1) TO PRINT THE FFT OR (2) FOR THE PSD ONLY"
2280 INPUT V
2290 PRINT
2300 IF V=2 GOTO 2430
2310 PRINT
2320 PRINT
2330 PRINT "  FFT   (#-1)   (REAL) X MATRIX  Y MATRIX"
2340 PRINT
2350 FOR K=1 TO N6
2360 PRINT (K-1),X(K),Y(K)
2370 NEXT K
2380 PRINT
2390 FOR K=N7 TO H1
2400 PRINT (K-1),X(K),Y(K)
2410 NEXT K
2420 PRINT
2430 RETURN
2440 END
```

Sample Run of PSD

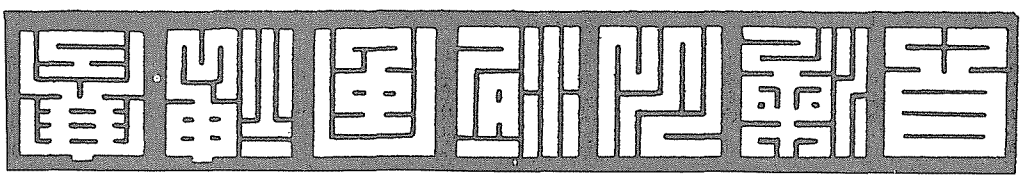
The data is entered into the program during execution. For this example 32 data points are entered. These 32 points represent 2 cycles of a square wave. These 32 data points will be saved in this sequence throughout the program so that different characteristics of the data may be studied. These 32 data points are entered on the first example page.

The first run uses 16 bit windows and uses HANN windowing on the X data only. The second page lists the FFT of every other data bit. This X and Y matrix listing represents the complex interactions of the two inputs, one with windowing and one without. The PSD for the two conditions is shown at the top of the third example page. The Ø or first bit (#-1) represents the D.C. level of the transformed window. The second and sixth harmonics are predominate in the PSD calculations. Notice the slight effect windowing has on the X matrix when compared to the Y matrix data. The original data is unsymmetrical.

The data is reused again with the two windows still set at 16 bits in length but this time no windowing is used on the X data. These 16 data points now represent one cycle of a square wave. The X and Y data points are set equal. The PSD of this is shown at the top of example page 4. It can be seen that both PSD's are equal and represent just the odd square wave harmonics. This test shows that two windows may be calculated simultaneously without loss in accuracy and also that the program is working properly.

At the bottom of this page the data is run again, here the Y data is slipped 7 bits ahead of the X data/ Again both PSD's are equal indicating proper operation of the program.

On the fourth and final run of the data the Y data is slipped 32 bits ahead of the X data. As there were only 32 bits of data that means all of the Y matrix will be zero. When the PSD is calculated we find that the X matrix is the same as previously calculated, however the Y matrix is approximately zero, which is as it should be as the input data was also zero. There are a number of interesting calculations that can be performed on test data to check out the operating characteristics of FFT's and PSD's, this has been one of them.



```
*RUN


THIS PROGRAM CALCULATES POWER SPECTRAL DENSITIES
AND FAST FOURIER TRANSFORMS.
IT USES TWO SLIDING WINDOWS AND AVERAGING & WINDOWING.


HOW MANY DATA POINTS DO YOU HAVE?
?32

ENTER YOUR DATA
?0
?1
?1
?1
?1
?1
?1
?1
?1
?

 0
?0
?0
?0
?0
?0
?0
?0
?1
?1
?1
?1
?1
?1
?1
?1
?1
?0
?0
?0
?0
?0
?0
?0
```

```
INPUT THE NUMBER OF BITS PER WINDOW.
THEN TYPE A (1), FOR   AVERAGES
OR A (2), FOR NO AVERAGING.
?16,2


INPUT THE STARTING BIT AND # OF BITS TO BE SKIPPED.
?1,2

DO YOU WANT WINDOWING ON THE DATA?(1 FOR YES, 2 FOR NO)
?1


HOW MANY BITS DO YOU WANT TO SLIDE THE Y MATRIX?
?0


FREQUENCY BANDWIDTH =      8  HERTZ.

FREQUENCY PER BIT   =      1  HERTZ.


TYPE A (1) TO PRINT THE FFT OR (2) FOR THE PSD ONLY
?1


  FFT  (#-1)   (REAL) X MATRIX  Y MATRIX

           0        .8183496              1
           1       -.0669249       .0373335
           2        .3877707      -.7413636
           3        .0114945        .069203
           4        .0036355       .0107477
           5        .0291014       .0091874
           6       -.0970298      -.3322683
           7         .026329       .0161572
           8       .0071122              0

           9         .026329      -.0161572
          10       -.3041371      -.1677317
          11        .0291014      -.0091874
          12        .0036355      -.0107477
          13        .0114945       -.069203
          14       -.8193366       .2413636
          15       -.0669249      -.0373335
```

```
   PSD        (#-1)          XMATRIX     Y MATRIX   START   +SLID

            0        .8183496                1        1       1
            1        .0058727         2.16840E-19     3       3
            2        .2880005         .426777         5       5
            3        .0049212         2.47334E-19     7       7
            4        .0001287                0        9       9
            5        .0009313         1.35525E-20    11      11
            6        .0470018        .0732234       13      13
            7        .0009543         1.21973E-19    15      15
            8         5.05827E-05            0       17      17

IF YOU WISH TO REUSE YOUR OLD DATA TYPE A (1), ELSE (2).
?~1
RETRANSMIT LAST LINE
1

INPUT THE NUMBER OF BITS PER WINDOW.
THEN TYPE A (1), FOR   AVERAGES
OR A (2), FOR NO AVERAGING.
?16,2


INPUT THE STARTING BIT AND # OF BITS TO BE SKIPPED.
?1,1

DO YOU WANT WINDOWING ON THE DATA?(1 FOR YES, 2 FOR NO)
?2


HOW MANY BITS DO YOU WANT TO SLIDE THE Y MATRIX?
?0


FREQUENCY BANDWIDTH =      8  HERTZ.

FREQUENCY PER BIT   =      1  HERTZ.


TYPE A (1) TO PRINT THE FFT OR (2) FOR THE PSD ONLY
?2
```

```
   PSD       (#-1)          XMATRIX     Y MATRIX   START   +SLID

          0                 1                1        1       1
          1          .410534          .410534        2       2
          2                 0                0        3       3
          3         .0506224         .0506224        4       4
          4                 0                0        5       5
          5          .022601          .022601        6       6
          6                 0                0        7       7
          7         .0162432         .0162432        8       8
          8                 0                0        9       9

IF YOU WISH TO REUSE YOUR OLD DATA TYPE A (1), ELSE (2).
?1

INPUT THE NUMBER OF BITS PER WINDOW.
THEN TYPE A (1), FOR   AVERAGES
OR A (2), FOR NO AVERAGING.
?16,2


INPUT THE STARTING BIT AND # OF BITS TO BE SKIPPED.
?1,1

DO YOU WANT WINDOWING ON THE DATA?(1 FOR YES, 2 FOR NO)
?2


HOW MANY BITS DO YOU WANT TO SLIDE THE Y MATRIX?
?7


FREQUENCY BANDWIDTH =      8  HERTZ.

FREQUENCY PER BIT   =      1  HERTZ.


TYPE A (1) TO PRINT THE FFT OR (2) FOR THE PSD ONLY
?2


   PSD       (#-1)          XMATRIX     Y MATRIX   START   +SLID

          0                 1                1        1       8
          1          .410534         .4105339        2       9
          2                 0                0        3      10
          3         .0506224         .0506224        4      11
          4                 0                0        5      12
          5          .022601          .022601        6      13
          6                 0                0        7      14
          7         .0162432         .0162433        8      15
          8                 0                0        9      16
```

```
IF YOU WISH TO REUSE YOUR OLD DATA TYPE A (1), ELSE (2).
?1

INPUT THE NUMBER OF BITS PER WINDOW.
THEN TYPE A (1), FOR   AVERAGES
OR A (2), FOR NO AVERAGING.
?16,2


INPUT THE STARTING BIT AND # OF BITS TO BE SKIPPED.
?1,1

DO YOU WANT WINDOWING ON THE DATA?(1 FOR YES, 2 FOR NO)
?2


HOW MANY BITS DO YOU WANT TO SLIDE THE Y MATRIX?
?32


FREQUENCY BANDWIDTH =      8  HERTZ.

FREQUENCY PER BIT   =      1  HERTZ.


TYPE A (1) TO PRINT THE FFT OR (2) FOR THE PSD ONLY
?2


   PSD       (#-1)          XMATRIX     Y MATRIX  START  +SLID

            0                1                0        1     33
            1          .410534         9.67362E-19    2     34
            2                0                0        3     35
            3         .0506224                0        4     36
            4                0                0        5     37
            5          .022601                0        6     38
            6                0                0        7     39
            7         .0162432         2.16840E-19    8     40
            8                0                0        9     41

IF YOU WISH TO REUSE YOUR OLD DATA TYPE A (1), ELSE (2).
?2

TO ENTER NEW DATA TYPE A (1), TO STOP TYPE A (2).
?2
```

RAND 1:

DESCRIPTION

Upon execution this program generates random numbers between Ø and 1. This program may be used as a subroutine in a larger program. Almost every value will be an irrational number.

USERS

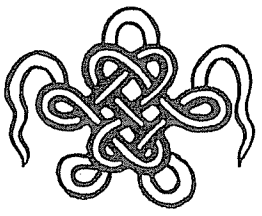
Anyone who needs a random number generator. Rand 1 can be used to set up initial conditions for a program such as a game or an analog simulation.

INSTRUCTIONS

List the program for program information. Type RUN and the program will list the first 100 random numbers to the accuracy of your computer.

LIMITATIONS

This program will store and execute in less then 1K Byte of memory.



```
RAND 1
```

```
10 REM GENERATES RANDOM #'S   0< X <1
20 LET A=4
30 LET A=A/3
50 LET B=(A+R)**8
60 LET R=B-INT(B)
70 LET I=I+1
80 IF I>100 THEN 110
90 PRINT R
100 GOTO 50
110 LET I=0
120 END
```

```
RUN
```

```
.988721231
.238685931
.296056223
.682040811
.173264307
.544727012
```

RAND 2:

DESCRIPTION

Rand 2 generates random numbers between X and Y, where X and Y are integers. This program may be used as a subroutine for a larger program. The generated numbers are all integers.

USERS

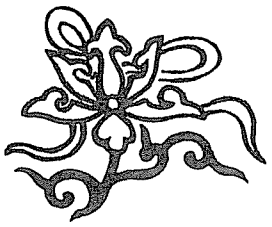
Anyone who needs a specific type of random number generator. Example: only integers between 1 and 6 to simulate dice rolls.

INSTRUCTIONS

Type RUN. Input Y (the largest #) and then input X (the smallest #). Then the program will print the first 100 random numbers between X and Y.

LIMITATIONS

Rand 2 will store and execute in less then 1K Byte of memory.



RAND 2

```
10 REM THIS PROGRAM GENERATES RANDOM #'S BETWEEN  X< R <Y
20 PRINT "TYPE IN THE LOWEST NUMBER DESIRED  -X- "
30 INPUT X
40 X=INT(X)
45 PRINT "TYPE IN THE LARGEST NUMBER -Y- "
50 INPUT Y
55 Y=INT(Y)
60 Y=Y-X
70 A=0.654321
80 A=A*1101
90 J=INT(A)
100 A=A-J
110 R=INT(A*Y+X+.5)
120 I=I+1
130 IF I>100 THEN 160
140 PRINT R
150 GOTO 80
160 I=0
170 END
```

RUN

```
LET Y=100
LET X=1
```

```
41
57
66
47
37
40
87
92
78
26
24
55
18
 8
```

SOLVE:

DESCRIPTION

Solve calculates the roots of any polynomial that has real coefficients.

USERS

This program will find most of it's users to be Engineers, Scientists, Mathematicians and high school students.

INSTRUCTIONS

Initial data must be entered before the program is run. When entering this data use the following format:

```
30 DATA  N,A(N),A(N-1),.....
```

where

N - is the order of the polynomial
A( ) - is the coefficients in descending order

Then type RUN. Bairstow's method is used to solve for the roots. For additional information list the program.

LIMITATIONS

Line 320 contains a Restore statement. Lines 780 and 960 contain ABS( ) statements. The source code will require 5K Bytes of memory for storage and 7K Bytes for execution. The sample problem finds the roots for the following three polynomials.

$3X + 6 = 0$ $X^2 - 6X + 5 = 0$ $3X^3 - 7X^2 + 4 = 0$

SOLVE

```
30 DATA 1E38
300 READ 08
310 IF 08 = 1E38 THEN  340
320 RESTORE
330 GO TO 430
340 PRINT "INSTRUCTIONS ? (1=MIN. , 2=ALL) ... WHICH ";
350 INPUT 00
360 IF 00 = 1  THEN  1980
370 IF 00 = 2  THEN  1960
380 LET 09 = 09+1
390 IF 09>3  THEN 9999
400 PRINT TAB(INT(36*RND(09)));" 1 OR 2 ... WHICH ";
410 GO TO 350
420 DIM  A(26), B(26), X(26)
430 LET G8=0
440 PRINT
450 LET C1=0
460 LET D1=1
470 LET 0=0
480 LET G8=G8+1
490 PRINT
500 READ N
510 LET N1 = N+1
520 LET M1 = N-1
530 IF N=0 THEN 9999
540 PRINT
550 PRINT "POLYNOMIAL NUMBER";G8;"IS OF ORDER";N
560 PRINT
570 PRINT "   COEFFICIENTS (IN DESCENDING ORDER) ARE:"
580 PRINT
590 PRINT "       ";
600 FOR I = 1 TO N1
610 READ A(I)
620 PRINT A(I);
630 LET B(I)=A(I)
640 NEXT I
650 PRINT
660 PRINT
670 PRINT "   THE ROOT";
680 IF N>1  THEN 710
690 PRINT " IS:"
700 GO TO 720
710 PRINT "S ARE:"
```

```
720 PRINT
730 IF N<=2 THEN 1420
740 IF A(N1)=0 THEN 1500
750 IF (N/2-INT(N/2))=0 THEN 780
760 GOSUB 1600
770 GOTO 730
780 IF ABS(A(M1))< 1.E-25 THEN 820
790 LET P=A(N)/A(M1)
800 LET Q=A(N1)/A(M1)
810 GOTO 840
820 LET P=A(N)
830 LET Q=A(N1)
840 FOR I=1 TO N1
850 LET X(I)=A(I)
860 NEXT I
870 GOSUB 1550
880 FOR I = 1 TO M1
890 LET B(I)=X(I)
900 NEXT I
910 LET R = X(N)
920 LET S = A(N1) - P*X(N) - Q*X(M1)
930 GOSUB 1550
940 LET X(N) = -P*X(M1) - Q*X(N-2)
950 LET D = X(M1)↑2 - X(N)*X(N-2)
960 IF ABS(D ) > 1.E-25 THEN 990
970 PRINT "SOLUTION UNOBTAINABLE WITH THIS PROGRAM."
980 GOTO 440
990 LET P1 = P + (R*X(M1) - S*X(N-2)) / D
1000 LET Q1 = Q + (S*X(M1) - R*X(N)) / D
1010 IF ABS(P ) > 1.E-25 THEN 1050
1020 IF ABS(P1) > 1.E-25 THEN 1050
1030 IF ABS( Q) > 1.E-25 THEN 1060
1040 GOTO 1070
1050 IF ABS(P1/P-1)>.000001 THEN 1070
1060 IF ABS (Q1/Q-1)<.000001 THEN 1190
1070 LET P=P1
1080 LET Q=Q1
1090 LET C1=C1+1
1100 IF C1=D1*25 THEN 1120
1110 GOTO 840
1120 PRINT
1130 GOSUB 1920
1140 INPUT K1
1150 IF K1=1 THEN 1170
1160 GOTO 440
1170 LET D1=D1+1
1180 GOTO 840
1190 FOR I = 2 TO M1
1200 LET A(I)=B(I)
1210 NEXT I
1220 LET N=N-2
```

```
1230 LET D=P*P-4*Q
1240 IF D<0 THEN 1340
1250 LET D=SQR(D)
1260 PRINT "       ";(-P+D)/2;" AND  ";(-P-D)/2
1270 LET C1 = 0
1280 LET D1=1
1290 IF N-2 <= 0  THEN  1420
1300 LET N1 = N+1
1310 LET M1 = N-1
1320 GO TO 740
1330 GOTO 1420
1340 LET D=SQR(-D)
1350 PRINT "       ";-P/2;"+ J *";D/2;" AND  ";-P/2;"- J *";D/2
1360 LET C1=0
1370 LET D1=1
1380 IF N-2 <= 0  THEN  1420
1390 LET N1 = N+1
1400 LET M1 = N-1
1410 GO TO 740
1420 IF N=1 THEN 1480
1430 IF N=0 THEN 440
1440 LET P = B(2) / B(1)
1450 LET Q = B(3) / B(1)
1460 LET N=0
1470 GOTO 1230
1480 PRINT "      ";-B(2) / B(1)
1490 GOTO 440
1500 PRINT "        0.000000"
1510 LET N1 = N
1520 LET N  = N-1
1530 LET M1 = N-1
1540 GOTO 730
1550 LET X(2) = X(2) - P*X(1)
1560 FOR I = 3 TO N
1570 LET X(I)=X(I)-P*X(I-1)-Q*X(I-2)
1580 NEXT I
1590 RETURN
1600 IF B(2) = 0  THEN 1630
1610 LET X = -B(2) / B(1)
1620 GOTO 1640
1630 LET X = -B(N1) / B(1)
1640 LET F=0
1650 LET F1=0
1660 FOR I = 1 TO N1
1670 LET J = N1 - I + 1
1680 IF B(J)=0 THEN 1720
1690 LET F = B(J)*X↑(I-1)  + F
1700 IF I = 1  THEN  1720
1710 LET F1 = (I-1)*B(J)*X↑(I-2) + F1
1720 NEXT I
```

```
1730 LET X1=X-F/F1
1740 IF ABS(X/X1-1) < .000001 THEN 1840
1750 LET X=X1
1760 LET C1=C1+1
1770 IF C1=D1*25 THEN 1790
1780 GOTO 1640
1790 GOSUB 1920
1800 INPUT K1
1810 LET D1=D1+1
1820 IF K1=1 THEN 1640
1830 GOTO 440
1840 PRINT "       ";X1
1850 LET N=N-1
1853 LET N1=N+1
1856 LET M1=N-1
1860 FOR I = 2 TO N1
1870 LET A(I)=B(I)+X1*A(I-1)
1880 LET B(I)=A(I)
1890 NEXT I
1900 RETURN
1910 STOP
1920 PRINT "THE SOLUTION DID NOT CONVERGE AFTER";C1;"ITERATIONS TO"
1930 PRINT "CONTINUE THE SOLUTION FOR 25 MORE ITERATIONS TYPE 1"
1940 PRINT "OTHERWISE TYPE 0.";
1943 IF C1<75 THEN 1950
1946 PRINT "  ANSWERS MAY BE INACCURATE.";
1950 RETURN
1960 PRINT
1970 PRINT"THIS PROGRAM FINDS ROOTS OF POLYNOMIALS BY BAIRSTOW'S METHOD."
1980 PRINT
1990 PRINT "TO USE, ENTER DATA IN DATA STATEMENTS AS FOLLOWS:"
2000 PRINT
2010 PRINT"   30 DATA N, A(N+1), A(N), A(N-1), ... , A(2), A(1)"
2020 PRINT
2030 IF 00 = 1  THEN  9999
2040 PRINT "WHERE N IS THE ORDER OF THE POLYNOMIAL AND"
2050 PRINT "A(I+1) IS THE COEFFICIENT OF THE I-TH DEGREE"
2060 PRINT "TERM.  MORE THAN ONE DATA LINE MAY BE USED"
2070 PRINT "TO SUPPLY COEFFICIENTS FOR ONE POLYNOMIAL,"
2080 PRINT "AND ADDITIONAL POLYNOMIALS MAY BE SOLVED ON"
2090 PRINT "A SINGLE RUN BY SUPPLYING DATA FOR THEM ON"
2100 PRINT "SUBSEQUENT DATA LINES (NOT BEYOND LINE 299)."
2110 PRINT
2120 PRINT "CAUTION:  THERE ARE A FEW TYPES OF POLYNOMIALS"
2130 PRINT "WHICH THIS PROGRAM IS UNABLE TO SOLVE. PROGRAM"
2140 PRINT "WILL SO INDICATE AND GO ON TO NEXT CASE.  ALSO,"
2150 PRINT "FOR HIGH ORDER POLYNOMIALS, PROGRAM MAY REQUIRE"
2160 PRINT "MANY ITERATIONS, SO RUNNING TIME CAN BE HIGH."
2170 DATA 0
9999 END
```

```
SOLVE


10 DATA 1,3,6
11 DATA 2,1,-6,5
12 DATA 3,3,-7,0,4


RUN

POLYNOMIAL NUMBER   1 IS OF ORDER  1
     COEFFICIENTS (IN DECENDING ORDER) ARE:
          3    6
     THE ROOT IS:
         -2


POLYNOMIAL NUMBER   2 IS OF ORDER   2
     COEFFICIENTS (IN DECENDING ORDER) ARE!
          1   -6     5
     THE ROOTS ARE:
          5   AND    1


POLYNOMIAL NUMBER   3 IS OF ORDER   3
     COEFFICIENTS (IN DECENDING ORDER)ARE:
          3    -7    0    4
     THE ROOTS ARE:
          2
          1  AND   -0.6666667
```

SPHERE TRIAN:

DESCRIPTION

This program solves spherical triangles that lie on the surface of the earth. Each must have it's apex at the North Pole.

USERS

Individuals plotting a course for air or sea navigation can use this program to calculate the most direct course.

INSTRUCTIONS

The information about the triangle must be entered before the program is run. The format is as follows:

```
     10 DATA  LTD,LTM,LGD,LGM,RLTD,RLTM,RLGD,RLGM,ALD,ALM
```

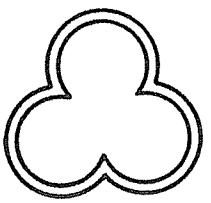
where

```
     LTD,LTM   = local latitude in degrees and minutes
     LGD,LGM   = local longitude in degrees and minutes
     RLTD,RLTM = remote latitude in degrees and minutes
     RLGD,RLGM = remote longitude in degrees and minutes
     ALD,ALM   = observed altitude in degrees and minutes
```

South and East are entered as negative numbers. For additional information list the program.

LIMITATIONS

Sphere Trian requires 5K Bytes for storage and should execute in 7K Bytes of memory in most systems.



SPHERE TRIAN

```
10 GO TO 434
100 DATA 1E50
102 PRINT
104 PRINT"   S P H E R I C A L   T R I A N G L E   S O L U T I O N"
106 PRINT
108 PRINT
110 LET L = 0
112 LET P = 3.14159265
114 LET C = 180/P
116 READ D0
118 IF D0 = 1E50 THEN 9999
120 READ M0, T0, N0, D1, M1, T1, N1, H0, J0
122 LET L = L+1
124 PRINT"CASE NUMBER";L
126 PRINT
128 PRINT
130 PRINT"LOCAL POSITION:"
132 PRINT
134 PRINT ABS(D0);"DEG   ";M0;"MIN  ";
136 IF D0 < 0 THEN 142
138 PRINT"NORTH ";
140 GOTO 144
142 PRINT"SOUTH ";
144 PRINT"LATITUDE"
146 PRINT ABS(T0);"DEG   ";N0;"MIN  ";
148 IF T0 < 0 THEN 154
150 PRINT"WEST ";
152 GOTO 156
154 PRINT"EAST ";
156 PRINT"LONGITUDE"
158 PRINT
160 PRINT
162 PRINT"REMOTE POSITION:"
164 PRINT
166 PRINT ABS(D1);"DEG   ";M1;"MIN  ";
168 IF D1 < 0 THEN 174
170 PRINT"NORTH ";
172 GOTO 176
174 PRINT"SOUTH ";
176 PRINT"LATITUDE"
178 PRINT ABS(T1);"DEG   ";N1;"MIN  ";
180 IF T1 < 0 THEN 186
182 PRINT"WEST ";
```

```
184 GOTO 188
186 PRINT"EAST ";
188 PRINT"LONGITUDE"
190 PRINT
192 PRINT
194 LET M0 = M0/60
196 LET N0 = N0/60
198 LET M1 = M1/60
200 LET N1 = N1/60
202 IF D0 >= 0 THEN 208
204 LET D0 = (ABS(D0)+M0+90)/C
206 GOTO 210
208 LET D0 = (90-(D0+M0))/C
210 IF D1 >= 0 THEN 216
212 LET D1 = (ABS(D1)+M1+90)/C
214 GOTO 218
216 LET D1 = (90-(D1+M1))/C
218 IF T0 >= 0 THEN 224
220 LET T0 = -(ABS(T0)+N0)/C
222 GOTO 226
224 LET T0 = (T0+N0)/C
226 IF T1 >= 0 THEN 232
228 LET T1 = -(ABS(T1)+N1)/C
230 GOTO 234
232 LET T1 = (T1+N1)/C
234 LET T = ABS(T1-T0)
236 LET F = COS(D0)*COS(D1)+SIN(D0)*SIN(D1)*COS(T)
238 LET F1 = SQR(ABS(1-F↑2))
240 LET Z = ATN(ABS(F1/F))
242 IF F >= 0 THEN 246
244 LET Z = P-Z
246 LET G = (COS(D1)-F*COS(D0))/(F1*SIN(D0))
248 LET B0 = ATN(ABS(SQR(ABS(1-G↑2))/G))
250 IF G >= 0 THEN 254
252 LET B0 = P-B0
254 IF (T0-T1) >= 0 THEN 258
256 LET B0 = 2*P-B0
258 LET H = (COS(D0)-F*COS(D1))/(F1*SIN(D1))
260 LET B1 = ATN(ABS(SQR(ABS(1-H↑2))/H))
262 IF H >= 0 THEN 266
264 LET B1 = P-B1
266 IF (T1-T0) >= 0 THEN 270
268 LET B1 = 2*P-B1
270 PRINT"LOCAL HOUR ANGLE (AT NORTH POLE):"
272 PRINT
274 PRINTINT(10*T*C+.5)/10;"DEG"
276 PRINT INT(T*C);"DEG   ";
278 PRINT INT(600*(T*C-INT(T*C))+.5)/10;"MIN"
280 LET H7 = T*C/15
282 LET M7 = (H7-INT(H7))*60
284 LET S7 = (M7-INT(M7))*60
```

```
286 PRINT INT(H7);"HRS   ";INT(M7);"MIN   ";INT(10*S7+.5)/10;"SEC"
288 PRINT
290 PRINT
292 PRINT"ZENITH (GREAT CIRCLE) DISTANCES:"
294 PRINT
296 PRINT INT(10*Z*C+.5)/10;"DEG"
298 PRINT INT(Z*C);"DEG   ";
300 PRINT INT(600*(Z*C-INT(Z*C))+.5)/10;"MIN"
302 PRINT INT(600*Z*C+.5)/10;"NAUTICAL MILES"
304 PRINT INT(600*Z*C*6080.2/5280+.5)/10;" STATUTE MILES"
306 PRINT
308 PRINT
310 PRINT"TRUE BEARINGS (GREAT CIRCLE COURSES):"
312 PRINT
314 PRINT" REMOTE POSITION FROM LOCAL POSITION:"
316 PRINT INT(10*B0*C+.5)/10;"DEG"
318 PRINT INT(B0*C);"DEG   ";INT(600*(B0*C-INT(B0*C))+.5)/10;"MIN"
320 PRINT
322 PRINT" LOCAL POSITION FROM REMOTE POSITION:"
324 PRINT INT(10*B1*C+.5)/10;"DEG"
326 PRINT INT(B1*C);"DEG   ";INT(600*(B1*C-INT(B1*C))+.5)/10;"MIN"
328 PRINT
330 PRINT
332 PRINT"ALTITUDE (REMOTE CELESTIAL POSITION"
334 PRINT"   ABOVE LOCAL POSITION HORIZON):"
336 PRINT
338 LET A7 = 90-Z*C
340 LET A8 = ABS(A7)
342 IF A7 < 0 THEN 348
344 PRINT INT(10*A7+.5)/10;
346 GOTO 350
348 PRINT -1*INT(10*A8+.5)/10;
350 PRINT"DEG"
352 IF A7 < 0 THEN 360
354 PRINT INT(A7);"DEG   ";
356 PRINT INT(600*(A7-INT(A7))+.5)/10;"MIN"
358 GOTO 364
360 PRINT -1*INT(A8);"DEG   ";
362 PRINT INT(600*(A8-INT(A8))+.5)/10;"MIN"
364 PRINT
366 PRINT
368 IF (1+H0)*(1+J0) = 1 THEN 418
370 PRINT"OBSERVED ALTITUDE:"
372 PRINT
374 PRINT H0;"DEG   ";J0;"MIN"
376 LET H1 = H0 + J0/60
378 PRINT INT(100*H1+.5)/100;"DEG"
380 PRINT
382 PRINT
384 PRINT"LINE OF POSITION:"
386 PRINT
```

```
388 LET I4 = 60*(H1-A7)
390 PRINT INT(10*ABS(I4)+.5)/10;"MILES ";
392 IF I4 > 0 THEN 398
394 PRINT"AWAY ";
396 GOTO 400
398 PRINT"TOWARDS ";
400 PRINT"ON LINE BEARING  ";
402 IF I4 > 0 THEN 416
404 LET B9 = B0*C+180
406 IF B9 > 360 THEN 412
408 PRINT INT(10*B9+.5)/10;"DEGREES TRUE"
410 GOTO 418
412 LET B9 = B9-360
414 GOTO 408
416 PRINT INT(10*B0*C+.5)/10;"DEGREES TRUE"
418 PRINT
420 PRINT
422 PRINT
424 PRINT
426 PRINT
428 PRINT
430 GOTO 116
432 PRINT
434 PRINT"      THE 'SPHERE' PROGRAM SOLVES SPHERICAL TRIANGLES HAVING"
436 PRINT"THE APEX AT THE NORTH POLE AND THE OTHER TWO CORNERS DEFINED"
438 PRINT"BY THEIR RESPECTIVE LATITUDES AND LONGITUDES.  DATA FOR AS"
440 PRINT"MANY CASES AS DESIRED MAY BE ENTERED SUCCESSIVELY IN DATA"
442 PRINT"STATEMENTS 10-99 IN THE FOLLOWING FORMAT:"
444 PRINT
446 PRINT"   10 DATA  LTD,LTM, LGD,LGM, RLTD,RLTM, RLGD,RLGM, ALD,ALM"
448 PRINT
450 PRINT"WHERE EACH PAIR OF NUMBERS SPECIFIES A LOCATION IN THE FORM"
452 PRINT"'DEGREES,MINUTES' AS FOLLOWS:"
454 PRINT
456 PRINT"   LTD,LTM    = LOCAL LATITUDE"
458 PRINT"   LGD,LGM    = LOCAL LONGITUDE"
460 PRINT"   RLTD,RLTM  = REMOTE LATITUDE"
462 PRINT"   RLGD,RLGM  = REMOTE LONGITUDE"
464 PRINT"   ALD,ALM    = OBSERVED ALTITUDE (IF ANY)"
466 PRINT
468 PRINT"           FOR SOUTH LATITUDES AND EAST LONGITUDES, ENTER "
470 PRINT"THE DEGREE VALUES AS NEGATIVE NUMBERS AND THE MINUTE"
471 PRINT"VALUES AS POSITIVE NUMBERS.  IF THERE IS NO OBSERVED"
472 PRINT"ALTITUDE, SET ALD AND ALM EQUAL TO ZERO."
474 PRINT
9999 END
```

```
SPHERE TRIAN


10 DATA 40,50,73,30,23,26,133,30,37,20


RUN


       SPHERICAL TRIANGLE SOLUTION

CASE NUMBER   1

LOCAL POSITION:

      40 DEG      50 MIN    NORTH LATITUDE
      73 DEG      30 MIN    WEST LONGITUDE


REMOTE POSITION:

      23 DEG      26 MIN    NORTH LATITUDE
     133 DEG      30 MIN    WEST LONGITUDE


LOCAL HOUR ANGLE (AT NORTH POLE):

      60 DEG
      59 DEG      60 MIN
       3 HRS      59 MIN       60 SEC


ZENITH (GREAT CIRCLE) DISTANCES:

 52.6 DEG
   52  DEG        37 MIN
3157 NAUTICAL MILES
  3635.5  STATUTE MILES
```

```
TRUE BEARINGS (GREAT CIRCLE COURSES):

 REMOTE POSITION FROM LOCAL POSITION:
      270.1 DEG
      270 DEG      4 MIN

 LOCAL POSITION FROM REMOTE POSITION:

 55.6 DEG
  55 DEG        33 MIN



ALTITUDE (REMOTE CELESTIAL POSITION
      ABOVE LOCAL POSITION HORIZON):

    37.4 DEG
     37 DEG         23 MIN

 OBSERVED ALTITUDE:

     37 DEG         20 MIN
     37.33 DEG

LINE OF POSITION:

     3  MILES AWAY ON LINE BEARING   90.1 DEGREES TRUE.
```

STARS:

DESCRIPTION

This program lists out the altitude and azimuth of fifty (50) of the most prominent celestial stars.

USERS

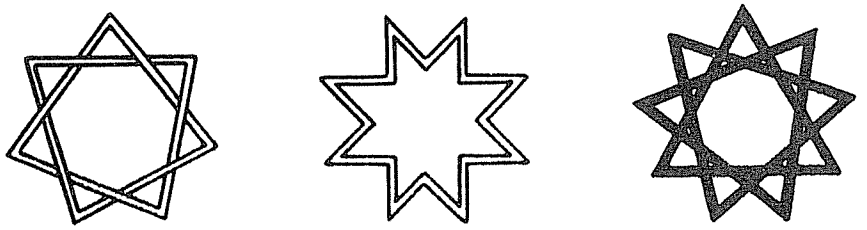
Stars will be of use for anyone interested in knowing the location of a particular star. This would include astronomers, mariners and people doing navigation at night.

INSTRUCTIONS

Type RUN and the program will ask for all necessary data. For detailed program information list the program. If an exact position is desired the GHA of Aries will have to be entered.

LIMITATIONS

The source code is 8K Bytes long. Execution of Star requires approximately 11K Bytes of memory.



STARS

```
10 PRINT "STAR LOCATION PROGRAM"
20 PRINT "NORTH + SOUTH - EAST + WEST -"
30 PRINT "FOR EXACT ARIES TYPE 1"
40 PRINT " FOR APPROX TYPE 2"
50 INPUT W9
60 IF W9=1 THEN 160
70 IF W9=2 THEN 80
80 PRINT "A2 IS THE DAY OF THE YEAR"
90 PRINT" A3 IS THE ZONE TIME(24 HOUR DECIMAL)"
100 PRINT "L LATITUDE DEGREES"
110 PRINT "A2,A3,L,"
120 INPUT A2,A3,L
130  J7=L
140 B6=100.025+(A2*.985)+(A3*15)
150 GOTO 240
160 PRINT "A2 A3 GHA ARIES DEG MIN"
170 PRINT "L,L1 LATITUDE   DEG MIN"
180 PRINT "A7 A8 LONGITUDE  DEG  MIN"
190 PRINT "A2,A3,L,L1,A7,A8"
200 INPUT A2,A3,L,L1,A7,A8
210  J7=L+(L1/60)
220 PRINT
230  B6=A2+(A3/60)+A7+(A8/60)
240  W=0
250 PRINT
260 PRINT "    STAR          ALTITUDE  DEGREES      AZIMUTH  DEG"
270 PRINT "-------------------------------------------------------------"
280 READ A,A1
290 IF A=400 GOTO 3280
300  C2=B6+A+(A1/60)
310 READ D,D1
320  C3=(D+(D1/60))/57.2957795
330 IF C2>1080 THEN 530
340 IF C2>900 THEN 550
350 IF C2>720 GOTO 410
360 IF C2>540 GOTO 430
370 IF C2>360 GOTO 450
380 IF C2>180 GOTO 470
390 IF C2<180 THEN 490
400 IF C2<0 THEN 510
410  X2=720-C2
420 GOTO 560
```

```
430  X2=720-C2
440 GOTO 560
450  X2=360-C2
460 GOTO 560
470  X2=360-C2
480 GOTO 560
490  X2=-C2
500 GOTO 560
510  X2=-C2
520 GOTO 560
530  X2=1080-C2
540 GOTO 560
550  X2=1080-C2
560  X4=ABS(X2)/57.2957795
570 IF X4>1.57079632 THEN 590
580 IF X4<1.57079632 THEN 610
590  X3=3.14159265 -X4
600 GOTO 620
610  X3=X4
620  B=ABS (X3)
630  B1=C3
640  B3=J7/57.2957795
650  B=X3
660  P=ATN((SIN(B)*COS(B1))/(1-(SIN(B)*COS(B1))↑2)↑.5)
670  K=ATN((SIN(B1)/COS(P))/(1-(SIN(B1)/COS(P))↑2)↑.5)
680  K6=ABS(K)
690  K1=ABS(K)*57.2957795
700 IF X4>1.57079632 THEN 780
710 IF X4<1.57079632 THEN 720
720 IF K1>90 THEN 740
730 IF K1<90 THEN 760
740  K2=180-K1
750 GOTO 830
760  K2=K1
770 GOTO 830
780 IF K1>90 THEN 800
790 IF K1<90 THEN 820
800  K2=K1
810 GOTO 830
820  K2=180-K1
830 IF K>0 THEN 850
840 IF K<0 THEN 870
850  K3=K2
860 GOTO 880
870  K3=-K2
880  K4=K3/57.2957795
890  B4=-B3
900  G1=K4+B4
910  G2=ABS(G1)
920  G5=G2*57.2957795
930 H=ATN((COS(P)*COS(G2))/(1-(COS(P)*COS(G2))↑2)↑.5)
```

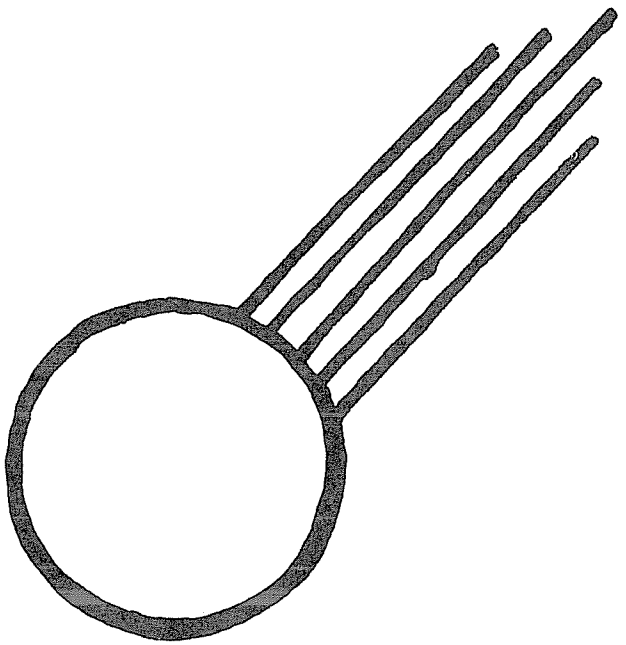
```
940  H1=H*57.2957795
950  H3=INT(H1)
960  H4=H1-H3
970  H5=H4*60
980 IF X4>1.57079632 THEN 1000
990 GOTO 1010
1000  H=-H
1010  P5=P
1020  Z=ATN((SIN(P5)/COS(H))/(1-(SIN(P5)/COS(H))↑2)↑.5)
1030  Z1=Z*57.2957795
1040  D3=K/B3
1050  B7=ABS(B3)
1060  K8=K2/57.2957795
1070 IF D3>0 THEN 1090
1080 IF D3<0 THEN 1110
1090 IF K8>B7 THEN 1130
1100 GOTO 1110
1110 IF Z1>90 THEN 1150
1120 IF Z1<90 THEN 1170
1130 IF Z1>90 THEN 1170
1140 IF Z1<90 THEN 1150
1150  Q1=Z1
1160 GOTO 1180
1170  Q1=180-Z1
1180 IF B3>0THEN 1200
1190 IF B3<0 THEN1260
1200 IF X2>0 THEN1220
1210 IF X2<0THEN 1240
1220  D5=Q1
1230 GOTO 1310
1240  D5=360-Q1
1250 GOTO 1310
1260 IF X2>0THEN 1280
1270 IF X2<0THEN 1300
1280  D5=180-Q1
1290 GOTO 1310
1300  D5=180+Q1
1310  W=W+1
1320 IF H3>0 THEN 1340
1330 IF H3<0 THEN 280
1340  D6=INT(D5)
1350 IF D6>360 THEN 1370
1360 GOTO 1380
1370  D6=D6-360
1380 IF W=1 THEN 1960
1390 IF W=2 THEN 1980
1400 IF W=3 THEN 2000
1410 IF W=4 THEN 2020
1420 IF W=5 THEN 2040
1430 IF W=6 THEN 2060
1440 IF W=7 THEN 2080
```

```
1450 IF W=8 THEN 2100
1460 IF W=9 THEN 2120
1470 IF W=10 THEN 2140
1480 IF W=11 THEN 2160
1490 IF W=12 THEN 2180
1500 IF W=13 THEN 2200
1510 IF W=14 THEN 2220
1520 IF W=15 THEN 2240
1530 IF W=16 THEN 2260
1540 IF W=17 GOTO 2280
1550 IF W=18 GOTO 2300
1560 IF W=19 GOTO 2320
1570 IF W=20 GOTO 2340
1580 IF W=21 GOTO 2360
1590 IF W=22 GOTO 2380
1600 IF W=23 GOTO 2400
1610 IF W=24 GOTO 2420
1620 IF W=25 GOTO 2440
1630 IF W=26 GOTO 2460
1640 IF W=27 GOTO 2480
1650 IF W=28 GOTO 2500
1660 IF W=29 GOTO 2520
1670 IF W=30 GOTO 2540
1680 IF W=31 GOTO 2560
1690 IF W=32 GOTO 2580
1700 IF W=33 GOTO 2600
1710 IF W=34 GOTO 2620
1720 IF W=35 GOTO 2640
1730 IF W=36 GOTO 2660
1740 IF W=37 GOTO 2680
1750 IF W=38 GOTO 2700
1760 IF W=39 GOTO 2720
1770 IF W=40 GOTO 2740
1780 IF W=41 GOTO 2760
1790 IF W=42 GOTO 2780
1800 IF W=43 GOTO 2800
1810 IF W=44 GOTO 2820
1820 IF W=45 GOTO 2840
1830 IF W=46 GOTO 2860
1840 IF W=47 GOTO 2880
1850 IF W=48 GOTO 2900
1860 IF W=49 GOTO 2920
1870 IF W=50 GOTO 2940
1880 IF W=51 GOTO 2960
1890 IF W=52 GOTO 2980
1900 IF W=53 GOTO 3000
1910 IF W=54 GOTO 3020
1920 IF W=55 GOTO 3040
1930 IF W=56 GOTO 3060
1940 IF W=57 GOTO 3080
1950 IF W=58 GOTO 3100
```

```
1960 PRINT "ALPHERATZ";H3;D6
1970 GOTO 280
1980 PRINT"ANKAA";H3;D6
1990 GOTO 280
2000PRINT "SCHEDAR";H3;D6
2010 GOTO 280
2020 PRINT "DIPHDA";H3;D6
2030 GOTO 280
2040 PRINT "ACHENAR";H3;D6
2050 GOTO 280
2060 PRINT "HAMAL";H3;D6
2070 GOTO 280
2080 PRINT "ACAMAR";H3;D6
2090 GOTO 280
2100 PRINT "MENKAR";H3;D6
2110 GOTO 280
2120 PRINT "MIRFAK";H3;D6
2130 GOTO 280
2140 PRINT "ALDEBARAN";H3;D6
2150 GOTO 280
2160 PRINT "RIGEL";H3;D6
2170 GOTO 280
2180 PRINT "CAPELLA";H3;D6
2190 GOTO 280
2200 PRINT "BELLATRIX";H3;D6
2210 GOTO 280
2220 PRINT "ELTH";H3;D6
2230 GOTO 280
2240 PRINT "ALNILAM";H3;D6
2250 GOTO 280
2260 PRINT "BETELGEUSE";H3;D6
2270 GOTO 280
2280 PRINT "CANOPUS";H3;D6
2290 GOTO 280
2300 PRINT "SIRIUS";H3;D6
2310 GOTO 280
2320 PRINT "ADHARA";H3;D6
2330 GOTO 280
2340 PRINT "PROCYON";H3;D6
2350 GOTO 280
2360 PRINT "POLLUX";H3;D6
2370 GOTO 280
2380 PRINT "AVOIR";H3;D6
2390 GOTO 280
2400 PRINT "SUHAIL";H3;D6
2410 GOTO 280
2420 PRINT "MIAPLACIDUS";H3;D6
2430 GOTO 280
2440 PRINT "ALPHARD";H3;D6
2450 GOTO 280
2460 PRINT "REGULUS";H3;D6
```

```
2470 GOTO 280
2480 PRINT "DUBHE";H3;D6
2490 GOTO 280
2500 PRINT "DENEBOLA";H3;D6
2510 GOTO 280
2520 PRINT "GIENAH";H3;D6
2530 GOTO 280
2540 PRINT "ACRUX";H3;D6
2550 GOTO 280
2560 PRINT "GACRUX";H3;D6
2570 GOTO 280
2580 PRINT "ALIOTH";H3;D6
2590 GOTO 280
2600 PRINT "SPICA";H3;D6
2610 GOTO 280
2620 PRINT "ALKAID";H3;D6
2630 GOTO 280
2640 PRINT "HADAR";H3;D6
2650 GOTO 280
2660 PRINT "MENKENT";H3;D6
2670 GOTO 280
2680 PRINT "ARCTURUS";H3;D6
2690 GOTO 280
2700 PRINT "RIGEL KENT";H3;D6
2710 GOTO 280
2720 PRINT "ZUBEN;UBI";H3;D6
2730 GOTO 280
2740 PRINT "KOCHAB";H3;D6
2750 GOTO 280
2760 PRINT "ALPHECCA";H3;D6
2770 GOTO 280
2780 PRINT "ANTARES";H3;D6
2790 GOTO 280
2800 PRINT "ATRIA";H3;D6
2810 GOTO 280
2820 PRINT "SABIK";H3;D6
2830 GOTO 280
2840 PRINT "SHAULA";H3;D6
2850 GOTO 280
2860 PRINT "RASALHAGUE";H3;D6
2870 GOTO 280
2880 PRINT "ELTANIN";H3;D6
2890 GOTO 280
2900 PRINT "KAUS AUST";H3;D6
2910 GOTO 280
2920 PRINT "VEGA";H3;D6
2930 GOTO 280
2940 PRINT "NUNKI";H3;D6
2950 GOTO 280
2960 PRINT "ALTAIR";H3;D6
2970 GOTO 280
```

```
2980 PRINT "PEACOCK";H3;D6
2990 GOTO 280
3000 PRINT "DENEB";H3;D6
3010 GOTO 280
3020 PRINT "ENIF";H3;D6
3030 GOTO 280
3040 PRINT "AL NAIR";H3;D6
3050 GOTO 280
3060 PRINT "FOMALHAUT";H3;D6
3070 GOTO 280
3080 PRINT "MARKAB";H3;D6
3090 GOTO 280
3100 PRINT "POLARIS";H3;D6
3110 GOTO 280
3120 DATA 358,17,28,56,353,47,-42,-28,350,18,56,22,349,28,-18,-9
3130 DATA 335,51,-57,-23,328,37,23,19,315,43,-40,-25,314,49,3,59
3140 DATA 309,27,49,45,291,27,16,27,281,43,-8,-14,281,23,45,58
3150 DATA 279,7,6,20,278,54,28,35,276,19,-1,-13,271,37,7,24
3160 DATA 264,11,-52,-41,259,3,-16,-40,255,38,-28,-56,245,34,5,18
3170 DATA 244,7,28,6,234,32,-59,-25,223,17,-43,-19,221,48,-69,-36
3180 DATA 218,28,-8,-32,208,18,12,7,194,31,61,55,183,7,14,44
3190 DATA 176,26,-17,-23,173,46,-62,-56,172,37,-56,-57
3200 DATA 166,49,56,7,159,5,-11,-1,153,24,49,28,149,34,-60,-14
3210 DATA 148,46,-36,-14,146,25,19,20,140,36,-60,-43
3220 DATA 137,41,-15,-55,137,17,74,17,126,38,26,49,113,6,-26,-22
3230 DATA 108,36,-68,-59,102,49,-15,-42,97,5,-37,-5
3240 DATA 96,36,12,35,91,1,51,29,84,26,-34,-24,81,0,38,45
3250 DATA 76,38,-26,-20,62,39,8,47,54,9,-56,-50,49,53,45,10
3260 DATA 34,19,9,44,28,24,-47,-6,15,59,-29,-47,14,10,15,3
3270 DATA 328,39,89,8.5,400,0
3280 END
```



TRACK:

DESCRIPTION

This program calculates the great circle distance course and track between two locations on the earth, given the Lat. and Long. of your departure and destination.

USERS

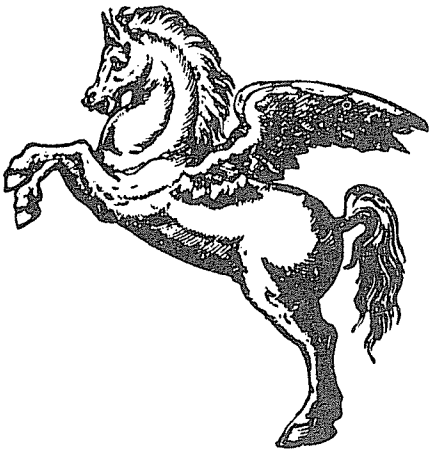
Track will be of use to navigators or individuals who plot courses. Aviators and sailors, all classes will be among the users of this program.

INSTRUCTIONS

List the program for detailed program information. Type RUN and the program will request all needed data.

LIMITATIONS

Line 420 contains an ABS( ) statement. The source code for Track requires 4K Bytes of memory and should execute in 6K Bytes.



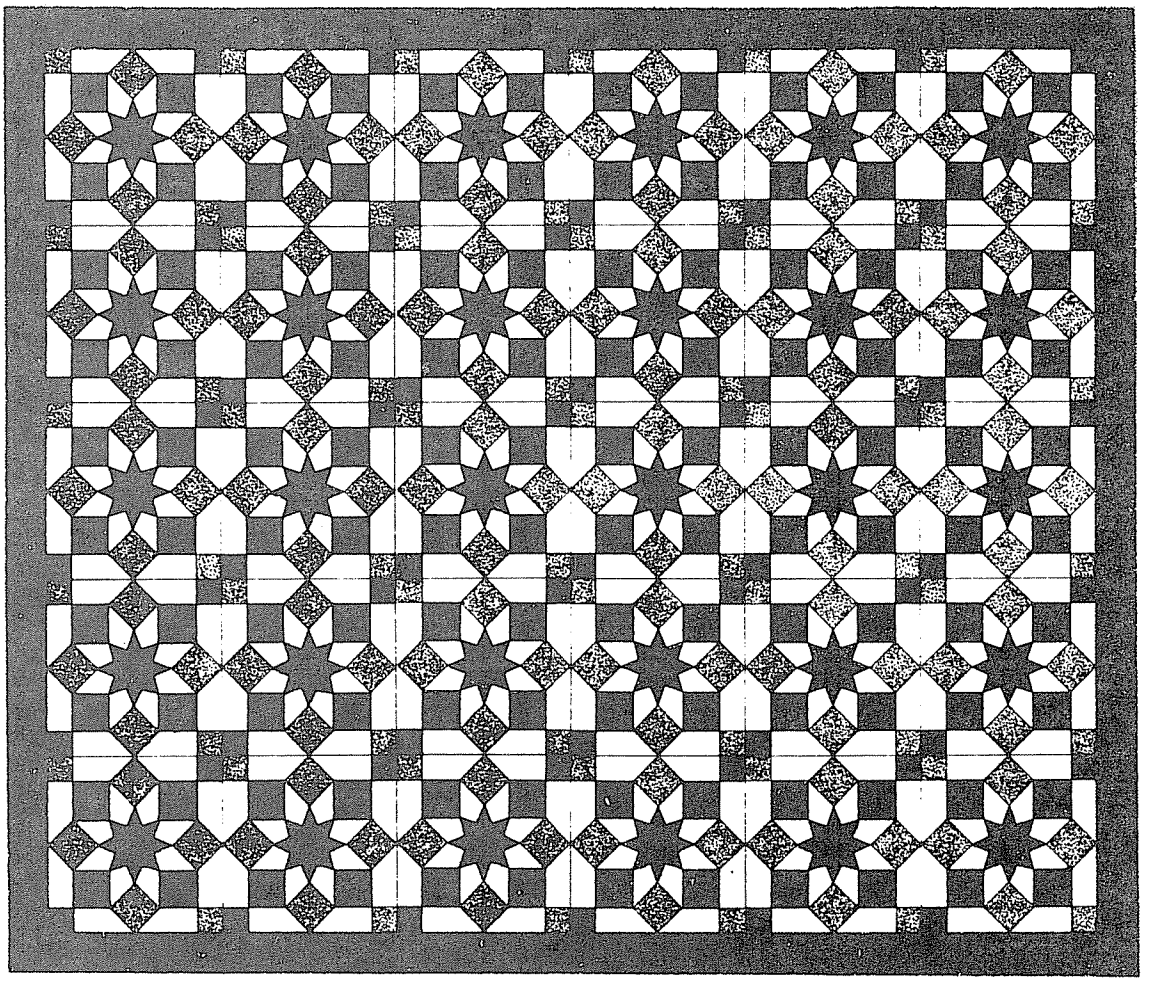
TRACK

```
100 PRINT "GREAT CIRCLE DISTANCE & COURSE +TRACK"
110 PRINT "A1,A2 LAT OF DEPARTURE "
120 PRINT "A3,A4 LONG DEPARTURE"
130 PRINT "DEG MIN       DEG MIN"
140 PRINT " A5,A6 LAT DESTINATION"
150 PRINT" A7,A8 LONG DESTINATION"
160 PRINT "EAST+ WEST- NORTH+ SOUTH-"
170 PRINT "J1 IS LONG OF FIRST PRINT OUT"
180 PRINT "J2 IS DEGREE STEP"
190 PRINT "J3 IS LONG OF SECOND PRINT OUT"
200 PRINT"TO TERMINATE WITHOUT TRACK MAKE J2=0"
210 PRINT
220 PRINT
230 PRINT
240 PRINT "A1,A2,A3,A4,A5,A6,A7,A8,J1,J2,J3"
250  Y=57.29577951
260  Y1=1.57079632
270 INPUT A1,A2,A3,A4,A5,A6,A7,A8,J1,J2,J3
280 PRINT
290 PRINT
300 B=(A1+(A2/60))/Y
310  A=A3+(A4/60)
320  B1=(A5+(A6/60))/Y
330  A9=A7+(A8/60)
340 IF A<0 GOTO 360
350 GOTO 370
360 A=360+A
370 IF A9<0 GOTO 390
380 GOTO 400
390 A9=360+A9
400 T2=A-A9
410 B3=B
420 T6=ABS(T2)
430 IF T6>180 GOTO 450
440 GOTO 490
450 IF T2<0 GOTO 480
460 T2=-360+T2
470 GOTO 490
480 T2=360+T2
490 T=T2
500 GOTO 1120
510 H7=H6
```

```
520 IF T2>0 GOTO 540
530 GOTO 550
540 D7=360-D7
550 PRINT "GREAT CIRCLE DISTANCE & INITIAL COURSE"
560 PRINT "N MILES";H6"COURSE";D7
570 IF J2=0 GOTO 1680
580 IF J1<0 GOTO 600
590 GOTO 640
600 J1=360+J1
610 IF J3<0 GOTO 630
620 GOTO 640
630 J3=360+J3
640 X=X+1
650 E5=E5+E1
660 IF F=J3 GOTO 1680
670 PRINT "COURSE";D4"DISTANCE NM";E5
680 IF X>1.5 GOTO 740
690 T=ABS(A-J1)/Y
700 IF T>(2*Y1) GOTO 720
710 GOTO 730
720 T=(4*Y1)-T
730 GOTO 800
740 T=ABS(J2/Y)
750 B3=Y1-H6
760 D8=(180-D9)/Y
770 GOTO 800
780 PRINT
790 PRINT
800 M=ATN(TAN(T)*SIN(B3))
810 M=Y1-M
820 R=(COS(T)*SIN(D8-M))/SIN(M)
830 D9=ATN(SQR(1-R↑2)/R)
840 IF D9<0 GOTO 860
850 GOTO 870
860 D9=(2*Y1)+D9
870 N=ATN(TAN(D8)*SIN(B3))
880 N=Y1-N
890 H6=ATN((TAN(B3)*COS(T-N))/COS(N))
900 E=ATN(TAN(B3)*COS(D8-M)/COS(M))
910 E1=(Y1-E)*Y*60
920 H6=Y1-H6
930 H8=90-(H6*Y)
940 D9=D9*Y
950 D4=180-D9
960 IF T2>0 GOTO 980
970 GOTO 990
980 D4=360-D4
990 GOTO 1000
1000 IF X>1.5 GOTO 1030
1010 F=J1
1020 GOTO 1250
```

```
1030 IF T2<0 GOTO 1050
1040 GOTO 1080
1050 F=F+J2
1070 GOTO 1250
1080 F=F-J2
1100 GOTO 1250
1110 PRINT
1120 T1=(ABS(T))/Y
1130 M=ATN(COT(B1)*COS(T1))
1140 R=(SIN(B1)*(COS(Y1-B3-M))/COS(M))
1150 H6=ATN(SQR(1-R↑2)/R)
1160 IF H6<0 GOTO 1180
1170 GOTO 1190
1180 H6=(2*Y1)+H6
1190 H6=H6*Y*60
1200 H8=90-(H6/60)
1210 D6=ATN((COT(T1)*SIN(Y1-B3-M))/SIN(M))
1220 D8=Y1-D6
1230 D7=90-(D6*Y)
1240 GOTO 510
1250 IF H8<0 GOTO 1320
1260 L=H8
1270 GO SUB 1620
1280 PRINT
1290 PRINT
1300 PRINT "              LAT";L1"DEG"L5"MIN NORTH"
1310 GOTO 1380
1320 H8=-H8
1330 L=H8
1340 GOSUB 1620
1350 PRINT
1360 PRINT
1370 PRINT "              LAT";L1"DEG"L5"MIN SOUTH"
1380 IF F>360 GOTO 1420
1390 IF F>180 GOTO 1470
1400 IF F>0 GOTO 1580
1410 IF F<0 GOTO 1490
1420 F1=F-360
1430 L=F1
1440 GOSUB 1620
1450 GOTO 1510
1460 GOTO 1530
1470 F1=360-F
1480 GOTO 1530
1490 F1=-F
1500 GOTO 1530
1510 PRINT               "              LONG";L1"DEG";L5"MIN EAST"
1520 GOTO 640
1530 GOTO 1540
1540 L=F1
1550 GOSUB 1620
```

```
1560 PRINT "                LONG";L1"DEG";L5"MIN WEST"
1570 GOTO 640
1580 F1=F
1590 L=F1
1600 GOSUB 1620
1610 GOTO 1510
1620 L1=INT(L)
1630 L2=(L-L1)*60
1640 L3=INT(L2)
1650 L4=(L2-L3)*10
1660 L5=L3+(INT(L4)/10)
1670 RETURN
1680 END
```



TRIANGLE:

DESCRIPTION

Triangle will solve for all parts of any plane triangle given one side and any two other parts of the triangle.

USERS

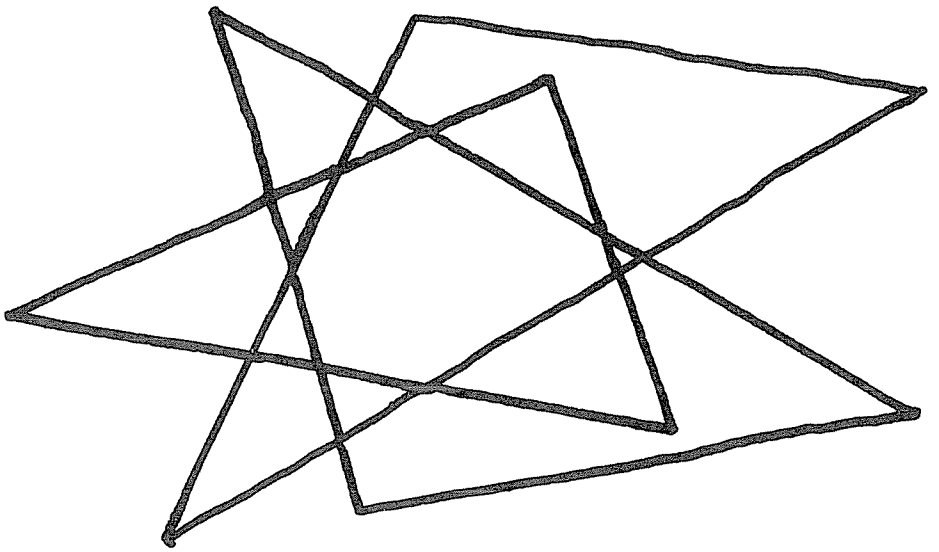
Surveyors, high school students, and anyone else who uses trigonometry can find use for this program.

INSTRUCTIONS

Type RUN and enter your data as asked. For additional information list the program.

LIMITATIONS

Line 105 contains a DEF FNS statement. This is used throughout the program. Triangle's source code requires 4K Bytes of memory to store and 6K Bytes of memory for execution.



TRIANGLE

```
10 DIM A(10),B(10),C(10),H(10),R(10),S(10)
100 PRINT
105 DEF FNS(I)= .01*INT(6000*(B(I)-INT(B(I)))+.5)
110 PRINT "THIS PROGRAM WILL FIND THE UNKNOWN FEATURES OF ANY"
115 PRINT "TRIANGLE, GIVEN ONE SIDE AND ANY TWO OTHER PARTS."
120 PRINT
125 PRINT
130 PRINT "WHAT WILL BE GIVEN (1=SSS, 2=SAS, 3=SSA, 4=ASA, 5=AAS)";
135 INPUT N
140 PRINT
145 LET R8=0
150 LET R9=0
155 IF (N-1)*(N-2)*(N-3)*(N-4)*(N-5)=0 THEN 170
160 PRINT "ANSWER 1, 2, 3, 4, OR 5 PLEASE...."
165 GOTO 125
170 IF N=1 THEN 210
175 PRINT "NOTE: SPECIFY ANGLES AS 'DEGREES, MINUTES, SECONDS'  OR"
180 PRINT "      'DEGREES, MINUTES.DECIMAL, 0' (I.E., SECONDS=0)"
185 PRINT
190 IF N=2 THEN 230
195 IF N=4 THEN 270
200 IF N=5 THEN 335
205 IF N=3 THEN 385
210 PRINT "WHAT ARE SIDE1, SIDE2, SIDE3";
215 INPUT S(1),S(2),S(3)
220 GOSUB 540
225 GOTO 630
230 PRINT "WHAT ARE SIDE1, ANGLE3, SIDE2";
235 INPUT S(1),A(3),B(3),C(3),S(2)
240 GOSUB 830
245 LET I=3
250 GOSUB 620
255 LET S(3)=SQR(S(1)↑2+S(2)↑2-2*S(1)*S(2)*COS(R(3)))
260 GOSUB 540
265 GOTO 630
270 PRINT "WHAT ARE ANGLE1, SIDE3, ANGLE2";
275 INPUT A(1),B(1),C(1),S(3),A(2),B(2),C(2)
280 GOSUB 830
285 FOR I=1 TO 2
290 GOSUB 620
295 NEXT I
300 LET R(3)=3.14159265-R(1)-R(2)
305 LET I=3
```

```
310 GOSUB 575
315 FOR I=1 TO 2
320 GOSUB 530
325 NEXT I
330 GOTO 630
335 PRINT "WHAT ARE ANGLE3, ANGLE1, SIDE3";
340 INPUT A(3),B(3),C(3),A(1),B(1),C(1),S(3)
345 GOSUB 830
350 FOR I=1 TO 3 STEP 2
355 GOSUB 620
360 NEXT I
365 LET R(2)=3.14159265-R(1)-R(3)
370 LET I=2
375 GOSUB 575
380 GOTO 315
385 PRINT "WHAT ARE SIDE1, SIDE3, ANGLE1";
390 INPUT S(1),S(3),A(1),B(1),C(1)
395 GOSUB 830
400 LET R8=1
405 LET I=1
410 GOSUB 620
415 LET T=(S(3)*SIN(R(1)))/S(1)
420 IF T-1>0.0000001 THEN 515
425 IF 1-T>0.00000001 THEN 440
430 LET R(3)=1.57079632
435 GOTO 445
440 LET R(3)=ATN(T/SQR(1-T↑2))
445 LET R(2)=3.14159265-R(1)-R(3)
450 LET I=2
455 GOSUB 530
460 FOR I=2 TO 3
465 GOSUB 575
470 NEXT I
475 IF R9=1 THEN 510
480 IF S(3)>S(1) THEN 490
485 GO TO 505
490 PRINT
495 PRINT "ONE SOLUTION IS:"
500 GO TO 510
505 LET R8 = 0
510 GOTO 630
515 PRINT
520 PRINT "NO SOLUTION POSSIBLE...."
525 GOTO 800
530 LET S(I)=(S(3)/SIN(R(3)))*SIN(R(I))
535 RETURN
540 LET P=(S(1)+S(2)+S(3))/2
545 LET R=SQR(((P-S(1))*(P-S(2))*(P-S(3)))/P)
550 FOR I=1 TO 3
555 LET R(I)=2*ATN(R/(P-S(I)))
560 GOSUB 575
```

```
565 NEXT I
570 RETURN
575 LET X1=57.2957795*R(I)
580 LET A(I)=INT(X1)
585 LET B(I)=60*(X1-A(I))
590 IF ABS(60-B(I))>0.00005 THEN 605
595 LET B(I)=0
600 LET A(I)=A(I)+1
605 IF ABS(B(I))>0.0001 THEN 615
610 LET B(I)=0
615 RETURN
620 LET R(I)=(A(I)+B(I)/60)/57.2957795
625 RETURN
630 GOSUB 540
635 LET W=R*P
640 LET R1=S(1)/(2*SIN(R(1)))
645 LET H(1)=S(3)*SIN(R(2))
650 LET H(2)=S(1)*SIN(R(3))
655 LET H(3)=S(2)*SIN(R(1))
660 PRINT
665 IF R9=1 THEN 685
670 PRINT
675 PRINT " ","         1","         2","         3"
680 PRINT " ","     -------","     -------","     -------"
685 PRINT
690 PRINT "SIDE",S(1),S(2),S(3)
695 PRINT
700 PRINT "ANGLE (RAD)",R(1),R(2),R(3)
705 PRINT
710 PRINT "ANGLE (DEG)",A(1),A(2),A(3)
715 PRINT "       (MIN)", INT(B(1)),INT(B(2)),INT(B(3))
720 PRINT "       (SEC)", FNS(1), FNS(2), FNS(3)
725 PRINT
730 PRINT "ALT TO SIDE",H(1),H(2),H(3)
735 PRINT
740 PRINT
745 PRINT "RADIUS OF CIRCUMSCR CIRCLE = ";R1
750 PRINT "RADIUS OF INSCRIBED CIRCLE = ";R
755 PRINT "           AREA OF TRIANGLE = ";W
760 PRINT
765 IF R9=1 THEN 800
770 IF R8=0 THEN 800
775 PRINT
780 PRINT "A SECOND SOLUTION IS:"
785 LET R9=1
790 LET R(3)=3.14159265-R(3)
795 GOTO 445
800 PRINT
805 PRINT
810 PRINT "ANOTHER CASE (1=YES)";
815 INPUT Q
820 IF Q=1 THEN 120
825 STOP
830 FOR I=1 TO 3
835 LET B(I)= B(I) + C(I)/60
840 NEXT I
845 RETURN
9999 FNM
```

```
TRIANGLE


RUN



THIS PROGRAM WILL FIND THE UNKNOWN FEATURES OF ANY
TRIANGLE, GIVEN ONE SIDE AND ANY TWOOTHER PARTS.

WHAT WILL BE GIVEN (1=SSS, 2=SAS, 3=SSA, 4=ASA, 5=AAS) ?2

NOTE: SPECIFY ANGLES AS DEGREES, MINUTES, SECONDS OR
      DEGREES, MINUTES.DECIMAL, 0   (I.E., SECONDS=0)

WHAT ARE SIDE1, ANGLE3, SIDE2 ?17,53,47,58.67,7.8


                      1              2              3
                   -------        --------       -------

SIDE                  17              7.8           13.9

ANGLE (RAD)        1.732681      0.4699307      0.9389807


ANGLE (DEG)           99             26             53
      (MIN)           16             55             47

      (SEC)         31.16          30.17          58.67

ALT TO SIDE        6.294261       13.71826       7.698017

RADIUS OF CIRCUMSCR CIRCLE =  8.612608
RADIUS OF INSCRIBED CIRCLE =  2.764921
          AREA OF TRIANGLE = 53.50121


ANOTHER CASE (1=YES) ?0
```

VARIABLE:

DESCRIPTION

This program is used to find and list all variables used in another program, written in any Basic. This program will enable the user to know how many variables have been used, what they were used for, and how many are free to be used for other quantities.

USERS

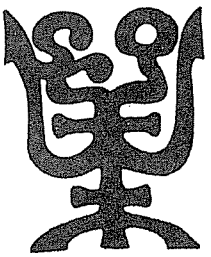
Anyone who is modifying a long program written in Basic will find this program extremely helpful.

INSTRUCTIONS

Before the program is run put the name of your use file after the FILES statement in line 40. Then type RUN. The program will then list the variables it finds in the use file and how the variables are used; refer to the example at the end of the program listing for more information.

LIMITATIONS

Line 90 contains a Restore statement. Line 40 FILES and line 100 Read #1. Starting in line 120 R$ & "___" this and string contantation SST( ) are used extensively throughout this program. The source code will store in 2K Bytes of memory. The amount of space required for program execution is a function of the length of the use file.



VARIABLE

```
20 REM THIS PROGRAM WILL LIST ALL OF THE VARIABLES IN THE
30 REM PROGRAM CALLED IN THE   FILES     STATEMENT
40 FILES %%%%%%%%
50 DIM V$(400),L$(400)
60 FOR I=1 TO 400
70 LET V$(I)="*"
80 NEXT I
90 RESTORE #1
100 READ #1,R$
110 IF END #1 GO TO 580
120 LET A$=R$&"%%%%"
130 FOR I=1 TO LEN(R$)
140 A=4
150 REM A DETERMINES HOW MANY CHAR. PRINT OUT ON THE RIGHT.
160 Z=I+A
170 LET K$=SST(A$,I,1)
180 IF K$<"0" GO TO 210
190 IF K$>"9" GO TO 210
200 NEXT I
210 N$=SST(A$,1,Z)
220 FOR J=I TO LEN(R$)
230 LET K$=SST(A$,J,1)
240 IF K$<"A" GO TO 270
250 IF K$>"Z" GO TO 270
260 GO TO 290
270 NEXT J
280 GO TO 100
290 LET J=J+1
300 LET K$=SST(A$,J,1)
310 IF K$="$" GO TO 440
320 IF K$<"A" GO TO 410
330 IF K$>"Z" GO TO 410
340 FOR K=J TO LEN(R$)
350 LET K$=SST(A$,K,1)
360 IF K$<"A" GO TO 400
370 IF K$>"Z" GO TO 400
380 NEXT K
390 GO TO 100
400 LET I=K+1\GO TO 220
410 IF K$<"0" GO TO 440
420 IF K$>"9" GO TO 440
430 GO TO 290
440 LET K$=SST(A$,I,J-I+1)
450 FOR K=1 TO 100
```

```
460 IF V$(K)="*" GO TO 500
470 IF K$=V$(K) GO TO 510
480 NEXT K
490 PRINT "TOO MANY VARIABLES"↖GO TO 580
500 LET V$(K)=K$
510 LET L$(K)=L$(K)&N$
520 LET L$(K)=L$(K)&":"
530 IF LEN(L$(K))<50 GO TO 560
540 PRINT V$(K),L$(K)
550 LET L$(K)=""
560 LET I=J+1
570 GO TO 220
580 FOR I=1 TO 100
590 IF V$(I)="*" GO TO 620
600 PRINT V$(I),L$(I)
610 NEXT I
620 END
```

```
RUN VARIABLE
```

```
A           REM T:
N1%         REM T:
C=          C=1%%:
D=          D=9*C:
9*C%        D=9*C:
B=          B=(D)*:
(D)*        B=(D)*:
*2)/6*D%    B=(D)*:
B%          PRINT :
```

```
LIST OF USE FILE .
```

```
10 REM THIS IS A TEST XX N1
20 C=1
30 D=9*C
40 B=(D**2)/6*D
50 PRINT B
60 END
```

VECTOR:

DESCRIPTION

When this program is run it will determine your present latitude and longitude, knowing only the starting point, distance traveled and course.

USERS

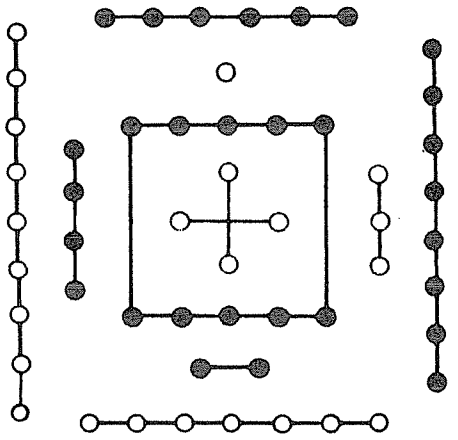
This program will be of assistance to anyone who navigates a boat, airplane or land vehicle. This program doesn't use data derived from a sextant but relies instead on data derived from a compass and/or odometer.

INSTRUCTIONS

The program will ask all necessary questions and prompt you for all inputs. For additional program information list Vector.

LIMITATIONS

Vector's source code is 2K Bytes in length and should execute in 4K Bytes of memory in most systems.



VECTOR

```
110 DATA 40,0,40,0,144,67,200,156,123,234,100,312,0,0
120 PRINT"FOR DESCRIPTION OF INPUTS LIST LINES130 to 190 "
130 REM  X,X1 LATITUDE OF DEPARTURE DEG MIN
140 REM X2,X3 LONGITUDE OF DEPARTURE DEG MIN
150 REM D DISTANCE TRAVELED NAUTICAL MILES
160 REM (EAST IS +)  (WEST IS-) (NORTH IS+) (SOUTH IS-)
170 REM H ANGLE FROM TRUE NORTH DEGREES
180 REM ANY NUMBER OF VECTORS
190 REM LAST INPUT IS 0,0
200 READ X,X1,X2,X3
210 LET X4=X+(X1/60)
220 LET X6=X2+(X3/60)
230 PRINT
240 PRINT
250 PRINT
260 LET K9=0
270 LET L9=0
280 READ D,H
290 IF D=0THEN 720
300 LET C=D
310 LET C1=H
320 IF C1>=270 THEN 360
330 IF C1>=180 THEN 380
340 IFC1>=90 THEN 400
350 IF C1<=90 THEN 420
360 LET A=360-C1
370 GOTO 440
380 LET A=C1-180
390 GOTO 500
400 LET A=180-C1
410 GOTO 560
420 LET A=C1
430 GOTO 620
440 LET B=A/57.2957795
450 LET B1=C*COS(B)
460 LET B2=C*SIN(B)
470 LET K=B1
480 LET L=-B2
490 GOTO 670
500 LET B=A/57.2957795
```

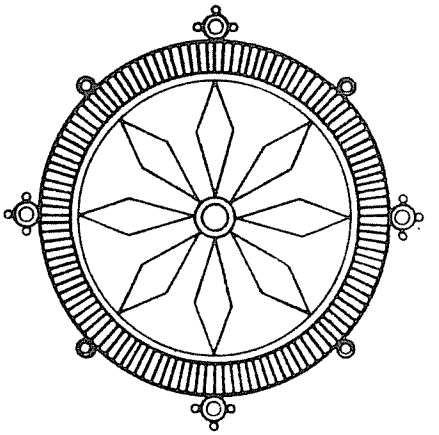
```
510 LET B1=C*COS (B)
520 LET B2=C*SIN(B)
530 LET K=-B1
540 LET L=-B2
550 GOTO 670
560 LET B=A/57.2957795
570 LET B1=C*COS(B)
580 LET B2=C*SIN(B)
590 LET K=-B1
600 LET L=B2
610 GOTO 670
620 LET B=A/57.2957795
630 LET B1=C*COS(B)
640 LET B2=C*SIN(B)
650 LET K=B1
660 LET L=B2
670 GOTO 680
680 LET K9=K+K9
690 LET L9=L+L9
700 PRINT"N+ S-",K,"E+ W-",L
710 GOTO 280
720 PRINT "L",K9,"P",L9
730 LET K1=ABS(K9)
740 LET L1=ABS(L9)
750 LET K7=ATN(L1/K1)
760 LET K8=K7*57.2957795
770 LET K2=L9/SIN(K7)
780 IF K9>0THEN 800
790 IF K9<0THEN 860
800 IF L9>0THEN 820
810 IF L9<0THEN 840
820 LET K3=K8
830 GOTO 910
840 LET K3=360-K8
850 GOTO 910
860 IF L9>0THEN 880
870 IF L9<0THEN 900
880 LET K3=180-K8
890 GOTO 910
900 LET K3=180 +K8
910 LET X8=X4+(K9/60)
920 PRINT
930 LET X9=(X4+X8)/114.5915590
940 LET Y=INT(X8)
950 LET Y1=X8-Y
960 LET Y2=Y1*60
970 LET Y3=(L9/COS(X9))/60
980 PRINT
990 LET Y4=X6+Y3
1000 PRINT
1010 LET Y5=INT(Y4)
1020 LET Y6=Y4-Y5
1030 LET Y7=Y6*60
1040 PRINT"ANGLE",K3,"DIST",K2
1050 PRINT "LAT    DEG        MIN              LO      DEG       MIN"
1060 PRINT Y,Y2,Y5,Y7
1070 END
```

PART 4

PLOTTING

AND

STATISTICS



BINOMIAL:

DESCRIPTION

The Binomial program calculates binomial distributions. The distributions are determined for problems where there are N number of trials and each trial has a probability P of success.

USERS

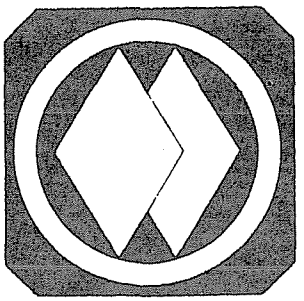
Gamblers, statisticians and mathematicians will have the most use for this program.

INSTRUCTIONS

When the program is run it will ask if you would like instructions. When asked for P, N and M enter their values. P and N are defined above. Unless this is used for plotting let M = 100. For additional instructions list the program.

LIMITATIONS

This program should store and execute in 3K Bytes of memory in most Basic speaking computer systems.



BINOMIAL

```
30 PRINT "DO YOU WANT INSTRUCTIONS (1=YES, 0=NO)";
40 INPUT I8
50 IF I8=0 THEN 300
60 PRINT
70 PRINT "THIS PROGRAM COMPUTES THE BINOMIAL DISTRIBUTION"
80 PRINT "FUNCTION FOR PROBLEMS OF THE FORM:"
90 PRINT
100 PRINT "     IN A SERIES OF N INDEPENDENT TRIALS, EACH WITH PROBABILITY"
110 PRINT "    OF SUCCESS P, WHAT IS THE PROBABILITY THAT THERE WILL"
120 PRINT "    BE EXACTLY X SUCCESSES?"
130 PRINT
140 PRINT "GIVEN P AND N, THE PROGRAM WILL SOLVE THIS PROBLEM FOR"
150 PRINT "ALL VALUES OF X IN WHICH THE PROBABILITY IS SIGNIFICANTLY"
160 PRINT "LARGE (UP TO ABOUT 7 STANDARD DEVIATIONS AWAY FROM P*N, THE"
170 PRINT "EXPECTED NUMBER OF SUCCESSES)."
180 PRINT
190 PRINT "THE NUMBER W WHICH IS REQUESTED IS A MULTIPLIER OF THE"
200 PRINT "BINOMIAL FUNCTION, SCALING IT TO A CONVENIENTLY GRAPHABLE"
210 PRINT "RANGE.  (W DOES NOT AFFECT THE VALUES OF THE BINOMIAL"
220 PRINT "FUNCTION, BUT IS MULTIPLIED AND PRINTED OUT IN A SEPARATE"
230 PRINT "COLUMN.  EACH VALUE OF B*W IS ROUNDED OFF TO THE NEAREST"
240 PRINT "INTEGER.)"
250 PRINT
260 PRINT
270 PRINT "TO TERMINATE PROGRAM TYPE STOP AT ANY REQUEST FOR"
280 PRINT "INFORMATION"
290 PRINT
300 PRINT
310 PRINT "WHAT ARE P,N,M";
320 INPUT P,N,W
330 PRINT
340 PRINT
350 PRINT "BINOMIAL DISTRIBUTION:  N=";N;"  P=";P
360 PRINT
370 PRINT "          X            X/N       B(X;P,N)          B*";W
380 PRINT
390 LET Q=1-P
400 LET M=N*P
410 LET S=SQR(N*P*Q)
420 LET X1=M-7*S
430 LET X2=M+7*S
440 IF X1<=0 THEN 460
450 IF P>.5 THEN 590
```

```
460 LET B=Q↑N
470 LET X=0
480 IF X1>0 THEN 510
490 PRINT X,X/N,B,INT(B*W+.5)
500 LET I2=N
510 FOR I=1 TO I2
520 LET B=B*(N-X)*P/((X+1)*Q)
530 LET X=X+1
540 IF (X-X1)*(X2-X)<0 THEN 560
550 PRINT X,X/N,B,INT(B*W+.5)
560 IF X>X2 THEN 290
570 NEXT I
580 GO TO 290
590 LET X=N
600 LET B=P↑N
610 LET B=B*X*Q/((N-X+1)*P)
620 LET X=X-1
630 IF X1<=X THEN 610
640 LET I2=N-X
650 GO TO 510
9999 END
```

```
BINOMIAL


RUN


DO YOU WANT INSTRUCTIONS (1=YES, 0=NO) ?0

WHAT ARE P,N,M ?.166667,30,100


BINOMIAL DISTRIBUTION:     N =  30    P =   0.166667
```

X	X/N	B(X;P,N)	B* 100
0	0	0.0042127	0
1	0.0333333	0.0252761	3
2	0.0666667	0.0733008	7
3	0.1	0.1368285	14
4	0.1333333	0.1847189	18
5	0.1666667	0.1921081	19
6	0.2	0.1600905	16
7	0.2333333	0.1097766	11
8	0.2666667	0.0631217	6
9	0.3	0.0308596	3
10	0.3333333	0.012961	1
11	0.3666667	0.0047131	0
12	0.4	0.0014925	0

```
WHAT ARE P,N,M ?STOP
```

CHI-SQ:

DESCRIPTION

This program performs a chi-square analysis to samples. This analysis will determine the number of expected results in the sample size and compare it to the actual results obtained. From this the results are determined to be accidental or deliberate.

USERS

Individuals who deal with data samples would find this program quite useful. This program could be utilized by individuals interested in Quality control or parts Reliability.

INSTRUCTIONS

To use this program enter your data starting in line 100. The format for this data is as follows:

```
     100  DATA    N1,S1,N2,S2,N3,S3,.........
```

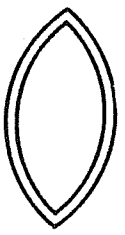
where

N(I) is the sample size
S(I) is the number of successes or failures for the sample

Then type RUN. Additional information may be obtained by listing the program.

LIMITATIONS

The DEF FND statement is used in line 20 of this program. This program should store and execute in 4K Bytes of memory.



CHI-SQ.

```
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)
20 DEF FND(V) = X(V+1) - X(V)
21 FOR I = 1TO 49
22 READ X(I)
23 NEXTI
100 GO TO 465
200 DATA 1E38, 1E38
205 DIM Z(100),N(100)
210 LET I=0
215 LET S1=0
220 LET S2=0
225 LET S3=0
230 LET I=I+1
235 READ N(I),Z(I)
240 IF N(I)=1E38 THEN 265
245 LET S1=S1+N(I)
250 LET S2=S2+Z(I)
255 LET S3=S3+(Z(I)↑2)/N(I)
260 GOTO 230
265 LET M=I-1
270 LET A=(S3-(S2↑2)/S1)/(M-1)
275 LET B=(S2/S1)*(1-S2/S1)
280 LET C2=(M-1)*A/B
285 LET B1=SQR(2*C2)-SQR(2*M-3)
290 GOSUB 370
295 PRINT
300 PRINT "CHI-SQUARE TEST OF SEVERAL PROPORTIONS:"
305 PRINT
310 PRINT " SAMPLE      SUCCESSES / TOTAL         % SUCCESSES"
315 PRINT
320 FOR I=1 TO M
325 PRINT"    ";I;"    ";Z(I);"/";N(I),"    ";.001*INT(.5+1E5*Z(I)/N(I))
330 NEXT I
335 PRINT
340 PRINT "CHI-SQUARED FOR THESE DATA    = ";C2
345 PRINT "CORRESPONDING NORMAL DEVIATE = ";B1
```

```
350 PRINT
355 PRINT " THE PROBABILITY OF THIS VALUE OF CHI-SQUARE"
360 PRINT " BEING EXCEEDED BY CHANCE ALONE IS";1-B2
365 STOP
370 IF B1<-4.5 THEN 425
375 IF B1<0 THEN 410
380 IF B1<4.5 THEN 395
385 LET B2=1
390 RETURN
395 GOSUB 435
400 LET B2=0
405 RETURN
410 GOSUB 435
415 LET B2=1-0
420 RETURN
425 LET B2=0
430 RETURN
435 LET Z=10*ABS(B1)
440 LET K=INT(Z)
445 LET D1=Z-K
450 LET Q=X(K+1)+D1*FND(K+1)+(D1*(D1-1)/2)*(FND(K+2)-FND(K+1))
455 LET Q=1E-3*INT(.5+1E-4*Q)
460 RETURN
465 PRINT
470 PRINT "THIS PROGRAM WILL APPLY A CHI-SQUARE TEST TO";
471 PRINT " SAMPLE PROPORTIONS."
475 PRINT "TO USE, TYPE:"
480 PRINT
485 PRINT "    100 DATA  N1,S1, N2,S2, N3,S3, ....."
490 PRINT "    RUN"
495 PRINT
500 PRINT "WHERE S1 IS THE NUMBER OF SUCCESSES IN SAMPLE 1";
501 PRINT ", SIZE N1"
505 PRINT "AND SO ON, FOR THE  M  SAMPLES."
506 PRINT
507 PRINT "YOU CAN USE LINES 100 THROUGH 199 FOR DATA."
9999 END
```

```
CHI-SQ.


100 DATA 1217,45,948,49,1165,33,1121,44

RUN


CHI-SQUARE TEST OF SEVERAL PROPORTIONS:

SAMPLE       SUCCESSES / TOTAL       % SUCCESSES

   1             45 /       1217        3.698
   2             49 /        948        5.169
   3             33 /       1165        2.833
   4             44 /       1121        3.925

CHI-SQUARED FOR THESE DATA   =    7.819924
CORRESPONDING NORMAL DEVIATE =    1.718657

THE PROBABILITY OF THIS VALUE OF CHI-SQUARE
BEING EXCEEDED BY CHANCE ALONE IS   0.043
```

COEFF:

DESCRIPTION

Coeff calculates the fourier coefficients of a periodic function which best fits experimental data points or a defining function. The final equation with the calculated coefficients will generate the initial data or equations with only sine and cosine functions.

USERS

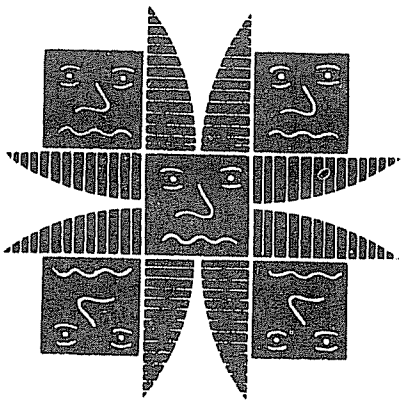
Experimenters would have use for this program. Anyone who analyzes data would also find Coeff very useful. The curve may be plotted using one of the plotting programs included in this section.

INSTRUCTIONS

Before the program is run all of the data must be entered in Data Statements. If the data is points of X and Y, enter it starting in line 2000. If the data is in the form of an equation or equations, enter these starting in line 6000. For detailed program instructions list the program.

LIMITATIONS

Starting in line 954 is a Print Using statement. This statement is used extensively throughout the program. The source code for this program is 7K Bytes long. Coeff requires 11K Bytes of memory for storage and execution.



COEFF

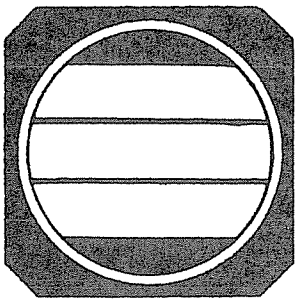
```
20 DIM P(200)
30 DIM A(50),B(50)
34REM...A AND B MUST BE DIMENSIONED AT LEAST ONE GREATER THAN THE MAX
42REM   # OF HARMONICS TO BE FITTED.  P MUST BE DIMENSIONED GREATER THAN
50REM   THE # DATA POINTS TO BE ENTERED...
58  LET N=201
66  LET J1=21
74REM...IF DATA POINTS ARE SUPPLIED BY A FUNCTION, 'N' POINTS WILL"
82REM   BE USED FOR THE FIT.  THE TABLE OF PREDICTED VALUES WILL"
90REM   HAVE 'J1' ENTRIES..."
98  LET P1=3.1415926536
106REM...P1 IS 'PI' & CAN BE USED IN DEFINING THE FUNCTION...
114REM
122  REM   THE COEFFICIENTS OF THE FOURIER SERIES THAT APPROXIMATE
130  REM   THE GIVEN FUNCTION ARE CALCULATED AS FOLLOWS:
138  REM
146  REM         C3=COS(2*PI/(2N+1))                          (1)
154  REM         S1=SIN(2*PI/(2N+1))                          (2)
162  REM         U1=0
170  REM         C=1
178  REM         S=0
186  REM         J=1
194  REM
202  REM   THE FOLLOWING RECURSIVE SEQUENCE IS USED TO COMPUTE
210  REM   U0,U1,AND U2:
218  REM
226  REM         U0=F(2M*PI/(2N+1))+2*C*U1-U2                 (3)
234  REM         U2=U1
242  REM         U1=U0
250  REM
258  REM   FOR VALUES OF M=2N,2N-1,.......,1
266  REM   THE COEFFICIENTS ARE THEN:
274  REM
282  REM         A(J)=2/(2N+1)[F(0)+C*U1-U2]
290  REM         B(J)=2/(2N+1)*S*U1
298  REM
306  REM   THE VALUES OF "C" AND "S" ARE UPDATED TO:
314  REM
322  REM         S=C3*S+S1*C
330  REM         C=0
338  REM
346  REM   "J" IS STEPPED BY 1 AND THE SEQUENCE STARTING AT EQUATION (3)
```

```
354  REM   IS NOW REPEATED UNTIL M+1 PAIRS OF COEFFICIENTS HAVE BEEN
362  REM   COMPUTE D.
370  PRINT
378  PRINT
386  PRINT
394 PRINT"-------------------------FOURIER SERIES PROGRAM-----------------"
402  PRINT
410  PRINT
418  PRINT
426  PRINT
434 LET D1=0.0
442  PRINT"DO YOU DESIRE INSTRUCTIONS---1=YES,0=NO";
450  INPUT A
458  IF A=1 THEN 1242
466  PRINT"ARE YOU USING A FUNCTION TO SUPPLY DATA POINTS (TYPE '1') OR"
474  PRINT"A SET OF DATA POINTS (TYPE '2')";
482  INPUT T
490  IF T=2 THEN 4080
498  LET N=N-1
506  LET M1=INT(N/2)
514  PRINT"WHAT IS THE MAXIMUM ORDER OF THE HARMONICS TO BE FITTED";
522  INPUT M
530  IF M <= M1 THEN 562
538  PRINT"THE NUMBER OF POINTS MUST BE GREATER THAN OR EQUAL TO THE"
546  PRINT"(MAXIMUM ORDER +1)*2"
554  STOP
562  IF M>49 THEN 578
570  GO TO 610
578  PRINT"THE NUMBER OF HARMONICS YOU HAVE REQUESTED IS LARGER THAN THE"
586 PRINT"STORAGE ALLOCATIONS OF THE PROGRAM-----TYPE '30 DIM A(N),B(N)'"
594  PRINT"WHERE N=NUMBER OF HARMONICS+1; THEN TYPE RUN"
602  STOP
610  REM
618  REM COMPUTE AND PRESET CONSTANTS
626  LET C1=2/N
634  LET C2=P1*C1
642  LET S1=SIN(C2)
650  LET C3=COS(C2)
658  LET C=1.0
666  LET S=0.0
674  LET J=1
682  LET X=0.0
690  GOSUB 1990
698  LET P2=Y
706  LET U2=0.0
714  LET U1=0.0
722  LET A2=N-1
730  REM
738  REM FORM FOURIER COEFFICIENTS RECURSIVELY
746  REM
754  LET X=A2*C2
```

```
762  GOSUB 1990
770  LET U0=Y+2.*C*U1-U2
778  LET U2=U1
786  LET U1=U0
794  LET A2=A2-1.
802  IF A2 >= 1E-4 THEN 754
810  LET A(J)=C1*(P2+C*U1-U2)
818  LET B(J)=C1*S*U1
826  LET K=J-(M+1)
834  IF K >= 0 THEN 882
842 LET Q=C3*C-S1*S
850 LET S=C3*S+S1*C
858 LET C=Q
866 LET J=J+1
874 GO TO 706
882  IF M1 <> N/2 THEN 914
890 IF M1 > M THEN 914
898  LET A(M1+1)=.5*A(M1+1)
906 LET B(M1+1)=0
914 PRINT
922 PRINT
930 PRINT
938 PRINT"FOURIER SERIES COEFFICIENTS:"
946 PRINT
954 PRINT USING 955
955:         A(I)              B(I)                 I
962 PRINT
970 FOR I=0 TO M
978 LET K1=I+1
986 PRINT A(K1),B(K1),I
987:#.########↑↑↑↑    #.########↑↑↑↑    ####
994 NEXT I
1002 PRINT
1010 PRINT
1018 PRINT
1026 PRINT
1034 PRINT"DO YOU WISH TO SEE A TABLE OF PREDICTED  VS. ACTUAL VALUES"
1042 PRINT"(1-YES,0=NO)";
1050 INPUT N
1058 IF N=1 THEN 1074
1066 STOP
1074 PRINT
1082 PRINT
1090 PRINT
1098 PRINT USING 1099
1099:            X          PRED-Y        ACTUAL-Y           ERROR
1106 PRINT
1114 LET K=2*P1/(J1-1)
1122 REM
1130 FOR X=0 TO 2*P1 STEP K
1138 LET Y1=A(1)*.5
```

```
1146 FOR I=2 TO M+1
1154 LET Y1=Y1+(A(I)*COS((I-1)*X)+B(I)*SIN((I-1)*X))
1162 NEXT I
1170 GOSUB 1990
1178 LET D=ABS(Y-Y1)
1186 LET D1=D1+D
1194 PRINT X,Y1,Y,D
1195:#.##########^^^^   #.##########^^^^   #.##########^^^^   #.##########^^^^
1202 NEXT X
1210 PRINT
1218 PRINT"TOTAL ERROR=",D1
1226 PRINT"MEAN ERROR=",D1/J1
1234 STOP
1242 PRINT
1250 PRINT"INSTRUCTIONS:"
1258 PRINT
1266 PRINT"THIS PROGRAM COMPUTES THE FOURIER COEFFICIENTS FOR A GIVEN"
1274 PRINT"PERIODIC FUNCTION OF THE FORM:"
1282 PRINT
1290 PRINT"                  N"
1298 PRINT"                 ___"
1306 PRINT"                 \"
1314 PRINT"   F(X)=A(0)/2+ /[A(N)*COS(N*X)+B(N)*SIN(N*X)]"
1322 PRINT"                 ___"
1330 PRINT"                N=1"
1338 PRINT
1346 PRINT"GIVEN THE FOLLOWING DATA:"
1354 PRINT
1362 PRINT"   1A. A SUBROUTINE IN LINES 2000-4000 THAT DEFINES 'Y' AS A"
1370 PRINT"       FUNCTION OF 'X' OVER THE INTERVAL [0,2*PI].  'P1'"
1378 PRINT"       CONTAINS THE VALUE OF PI AND CAN BE USED IN THE"
1386 PRINT"       SUBROUTINE.  FOR EXAMPLE:"
1394 PRINT
1402 PRINT"          2000 IF X < P1 THEN 2030"
1410 PRINT"          2010 LET Y=X/P1"
1418 PRINT"          2020 GO TO 2040"
1426 PRINT"          2030 LET Y=X/P1-1.0"
1434 PRINT"          2040 RETURN"
1442 PRINT
1450 PRINT"       OR"
1458 PRINT
1466 PRINT"   1B. DATA STATEMENTS IN LINES 6000-6998 CONTAINING THE"
1474 PRINT"       TABULATED FUNCTION VALUES EQUALLY SPACED ON [0,2*PI]"
1482 PRINT"       SO THAT THE FIRST POINT IS THE FUNCTION VALUE AT ZERO"
1490 PRINT"       AND THE LAST POINT IS THE FUNCTION VALUE AT 2*PI."
1498 PRINT
1506 PRINT"    2. THE NUMBER OF POINTS IF (1B) IS USED TO DEFINE THE"
1514 PRINT"       FUNCTION. (REQUESTED DURING PROGRAM EXECUTION)"
1522 PRINT"       IF (1A) IS USED, THE PROGRAM USES ";N;"POINTS."
1530 PRINT
1538 PRINT"    3. THE DESIRED ORDER OF THE FOURIER COEFICIENTS."
```

```
1546 PRINT"        (REQUESTED DURING PROGRAM EXECUTION)"
1554 PRINT
1562 PRINT"NOTE:  THE ORDER OF THE SERIES MUST BE >= ZERO."
1570 PRINT
1578 PRINT"        THE # POINTS MUST BE > TWICE THE ORDER OF THE SERIES."
1586 PRINT
1594 PRINT"        B(0) IS ALWAYS ZERO."
1602 PRINT
1610 STOP
1990 IF T=2 THEN 4020
4010 RETURN
4020 LET H=(X/(2.*P1))*(J1-1)+1.0
4030 LET L=INT(H)
4040 IF H-L<0.5 THEN 4060
4050 LET L=L+1
4060 LET Y=P(L)
4070 RETURN
4080 PRINT"HOW MANY DATA POINTS";
4090 INPUT N
4100 IF N< 200 THEN 4150
4110 PRINT"THE NUMBER OF POINTS IS GREATER THAN THE STORAGE "
4120 PRINT"ALLOCATIONS OF THE PROGRAM.  TYPE '20 DIM P(N)' WHERE N"
4130 PRINT"EQUALS THE NUMBER OF POINTS, THEN TYPE'RUN'."
4140 STOP
4150 FOR I=1 TO N
4160 READ P(I)
4170 NEXT I
4180 LET J1=N
4190 GO TO 493
9998 DATA 1E35
9999 END
```



```
2000 IF ABS(X)<1E-4 THEN 2510
2010 IF ABS(X-P1)<1E-4 THEN 2510
2020 IF ABS(X-2*P1)<1E-4 THEN 2510
2030 IF X>P1 THEN 2540
2040 LET Y=X/P1
2050 RETURN
2510 LET Y=.5
2520 RETURN
2540 LET Y=(X-P1)/P1
2550 RETURN


RUN


----------FOURIER SERIES PROGRAM----------


DO YOU DESIRE INSTRUCTIONS ---1=YES,0=NO ?0
ARE YOU USING A FUNCTION TO SUPPLY DATA POINTS (TYPE 1) OR
A SET OF DATA POINTS (TYPE 2) ?1
WHAT IS THE MAXIMUM ORDER OF THE HARMONICS TO BE FITTED ?10


FOURIER SERIES COEFFICIENTS:

   A(I)          B(I)              I

.9998307          0                0
1.50909E-05     9.64904E-06        1
1.79135E-05     -.3181999          2
4.32217E-06     5.53251E-07        3
1.00547E-05     -.1589442          4
2.51442E-06     -6.33756E-08       5
5.79206E-06     -.1057887          6
1.66236E-06     -1.99909E-07       7
4.05776E-06     -.0791582          8
1.21063E-06     -2.49538E-07       9
3.18545E-06     -.0631376         10


DO YOU WANT TO SEE A TABLE OF PREDICTED VS. ACTUAL VALUES
(1=YES, 0=NO) ?0


TOTAL ERROR =       0.6367112
MEAN ERROR =        0.0303196
```

CONFIDENCE 1:

DESCRIPTION

When executed Confidence 1 computes confidence limits for simple linear regressions.

USERS

People engaged in statistical analysis of data will have use for this program.

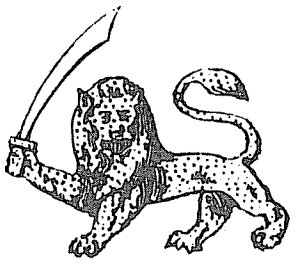
INSTRUCTIONS

Enter the X and Y data into the program before it is run. Enter the X data sequentially starting in line 100, as follows:

```
          100 DATA  X(1),X(2),X(3),.....
and for Y:
          400 DATA  Y(1),Y(2),Y(3),.....
```

LIMITATIONS

A Restore statement is used in line 730. The program will store and execute in 8K Bytes of memory.



CONFIDENCE 1

```
30 READ W9
40 IF W9=1E38 THEN 1710
730 RESTORE
740 PRINT
750 PRINT "S I M P L E   L I N E A R   R E G R E S S I O N"
760 PRINT
770 PRINT "           EQUATION:    Y = A + B*X"
780 PRINT
790 PRINT
800 DIM U(250),V(250)
810 PRINT "HOW MANY OBSERVATIONS ON EACH VARIABLE";
820 INPUT N
825 LET N1 = N-1
826 LET R1 = N/N1
830 PRINT
840 PRINT
850 IF N>2 THEN 880
860 PRINT "MUST HAVE THREE OR MORE OBSERVATIONS."
870 STOP
880 LET S1=0
890 LET S2=0
900 LET S3=0
910 LET S4=0
920 LET S5=0
930 FOR I=1 TO N
940 READ V(I)
950 NEXT I
960 FOR I=1 TO N
970 READ U(I)
980 LET S1=S1+U(I)
990 LET S2=S2+U(I)↑2
1000 LET S3=S3+V(I)
1010 LET S4=S4+V(I)↑2
1020 LET S5=S5+U(I)*V(I)
1030 NEXT I
1040 LET M1=S1/N
1050 LET M2=S3/N
1060 LET D1 = S2/N - M1↑2
1065 LET E1 = D1 * R1
1070 LET D2 = S4/N - M2↑2
1075 LET E2 = D2 * R1
1080 LET D3=S5/N-M1*M2
1090 LET C1=N*D1
```

```
1100 PRINT "VARIABLE", "      MEAN", "   VARIANCE", "STD DEVIATION"
1110 PRINT
1120 PRINT "   X",M1,E1,SQR(E1)
1130 PRINT "   Y",M2,E2,SQR(E2)
1140 PRINT
1150 PRINT
1160 LET R8=0
1170 LET B=D3/D1
1180 LET A=M2-B*M1
1190 LET D4=D2-B*D3
1200 IF D4<D2 THEN 1240
1210 LET R8=1
1220 LET R2=0
1230 GO TO 1250
1240 LET R2=1-(D4/D2)
1250 PRINT " INDEX (R↑2)", "    EXPL VAR", "   UNEXPL VAR", "   STD ERROR"
1260 PRINT
1270 IF R8=0 THEN 1300
1280 PRINT "  ZERO          (ST ERROR OF ESTIMATE EXCEEDS STD DEV OF Y)"
1290 GOTO 1310
1300 PRINT R2,D2-D4,D4,SQR(D4)
1310 PRINT
1320 PRINT
1330 LET D4=N*D4/(N-2)
1340LET T=1.96513+2.37231/(N-2)+2.812/(N-2)↑2+2.51562/(N-2)↑3
1350LETT=T+1.03125/(N-2)↑4+1.8125/(N-2)↑5
1360 LET D5=SQR(D4/C1)
1370LETD6=SQR(D4*S2/(N*C1))
1380 LET B1=B-T*D5
1390 LET B2=B+T*D5
1400 LET A1=A-T*D6
1410 LET A2=A+T*D6
1420 PRINT "PARAMETER","    VALUE","   95 PCT CONFIDENCE LIMITS"
1430 PRINT
1440 PRINT "   A",A,A1,A2
1450 PRINT "   B",B,B1,B2
1460 PRINT
1470 PRINT
1480 PRINT "ESTIMATED VALUES OF Y (FROM THE REGRESSION) AND CONFIDENCE"
1490 PRINT "LIMITS FOR INDIVIDUAL VALUES OF Y, FOR EACH VALUE OF X:"
1500 PRINT
1510 PRINT"      X-ACTUAL","       Y-ACTUAL","     Y-CALC",
1520 PRINT "   95 PCT CONFIDENCE LIMITS"
1530 PRINT
1540 DEF FNA(U)=SQR(D4*(1+C2+((U-M1)↑2)/C1))
1550 LET C2=1/N
1560 FOR I=1 TO N
1570 LET D7=FNA(U(I))
1580 LET Y=A+B*U(I)
1590 LET C3=T*D7
1600 PRINT U(I),V(I),Y,Y-C3,Y+C3
```

```
1610 NEXT I
1620 PRINT
1630 READ X
1640 IF X=1E38 THEN 9999
1650 LET D8=FNA(X)
1660 LET C4=T*D8
1670 LET Y=A+B*X
1680 PRINT X,"    ",Y,Y-C4,Y+C4
1690 GOTO 1630
1700 DATA 1E38
1710 PRINT

1725PRINT"----"
9999 END
```

```
 100 DATA 116,132,104,139
 200 DATA 105,120,85,121
```

```
 RUN
```

```
 SIMPLE LINEAR REGRESSION

            EQUATION:  Y = A + B*X

 HOW MANY OBSERVATIONS ON EACH VARIABLE ?4

 VARIABLE          MEAN          VARIANCE          STD DEVIATION

    X              107.75        283.5833            16.83993
    Y              122.75        248.9167            15.77709


 INDEX (R**2)      EXPL VAR        UNEXPL VAR       STD ERROR

   0.94237758       175.9298        10.75772         3.279896


 PARAMETER          VALUE        95 PCT CONFIDENCE LIMITS

     A             24.75228          -49.42544        98.93
     B             0.9094916          0.2272877       1.591696


 ESTIMATED VALUES OF Y (FROM THE REGRESSION) AND CONFIDENCE
 LIMITS FOR INDIVIDUAL VALUES OF Y, FOR EACH VALUE OF X:

      X-ACTUAL       Y-ACTUAL      Y-CALC       95 PCT CONFIDENCE LIMITS

          105            116        120.2489        97.923      142.5749
          120            132        133.8913       110.1265     157.6561
           85            104        102.0591        74.93341    129.1847
          121            139        134.8008       110.7876     158.814
```

CONFIDENCE 2:

DESCRIPTION

This program calculates confidence limits for an unknown sample mean.

USERS

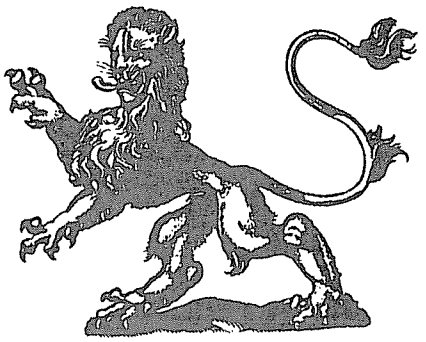
Individuals studying data variables or irregularities within a sample could find this program useful.

INSTRUCTIONS

Enter the sample size in a data statement on line 50. Then enter the data sequentially in data statements starting in line 100. Then type RUN. For additional instructions list the program.

LIMITATIONS

The DEF FN__ is used in program lines 38, 39, 40, and 41. The program should execute in most systems without incident. The source code requires 3K Bytes of memory for storage and will require 4K Bytes for execution.



CONFIDENCE 2

```
26 PRINT
27 PRINT
28 PRINT
29 REM NORMAL CURVE PROBABLILITES...KEEP 'EM HERE...DO NOT DELETE.
30 DATA 5000000,5398278,5792597,6179114,6554217,6914625,7257469
31 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
32 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
33 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
34 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
35 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
36 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
37 DIM X(49)
38 DEF FND(U)=M+U*S
39 DEF FND(U)=X(U)-X(U-1)
40 DEF FNB(U)=U-U*(U-1)*(D2/(2*D1)+(U-2)*D3/(6*D1))
41 DEF FNZ(U) = 1+(U↑2+1)/(4*D)+((U↑2+3)*(5*U↑2+1))/(96*D↑2)
42 DATA 0,0,0
50 DATA  1E20
100 DATA 2,3,6,9,8,11,13,16,23,45,67,89,35,67,13,43,21,67,98,7,6
200 DATA 3E33
210 DATA .5,.75,.9,.95,.99,.999,.9999,.99999,1E38
220 FOR I = 1 TO 49
230 READ X(I)
240 NEXT I
250 READ I,S1,S2
260 READ H
270 LET I=I+1
280 READ W
290 IF W=3E33 THEN 330
300 LET S1=S1+W
310 LET S2=S2+W↑2
320 GOTO 270
330 LET N=I-1
340 PRINT
350 PRINT "VALUES OF SAMPLE STATISTICS:"
360 PRINT
370 PRINT "   SIZE OF SAMPLE",N
380 LET M=S1/N
390 PRINT "   SAMPLE MEAN VALUE",M
400 LET S8=S2/N-M↑2
410 LET D = N-1
420 PRINT "   VARIANCE OF SAMPLE",S8
430 PRINT "   SAMPLE STD DEVIATION",SQR(S8)
440 LET S5=S8*N*(H-1)/(H*(N-1))
```

```
450 PRINT "   ESTIMATED POPN STD DEV";SQR(S5)
460 LET S6=S5*(N-N)/(N*(N-1))
470 LET S=SQR(S6)
480 PRINT "   STANDARD ERROR OF MEAN";S
490 PRINT
500 PRINT
510 PRINT "CONFIDENCE LIMITS ON POPULATION MEAN:"
520 PRINT
530 PRINT "  CONF LEVEL","  LOWER LIM","   UPPER LIM"
540 PRINT
550 READ P
560 IF P=1E38 THEN 9999
570 LET A1 = 0.5*(1+P)
580 GOSUB 610
590 LET A3 = A2 * FNZ(A2)
592 PRINT 100*P, FNQ(-A3), FNQ(A3)
600 GOTO 550
610 IF A1>0.5 THEN 660
620 LET A1=1-A1
630 GOSUB 700
640 LET A2=-Q
650 GOTO 680
660 GOSUB 700
670 LET A2=Q
680 RETURN
690 REM  REVERSE INTERPOLATION FOR STD NORMAL DEVIATE:
700 LET Z=1E7*A1
710 FOR I = 1 TO 46
720 IF Z<X(I) THEN 740
730 NEXT I
740 LET D1 = FND(I)
750 LET E1 = FND(I+1)
760 LET D2=E1-D1
770 LET D3 = FND(I+2) -E1 - D2
780 LET U = (Z - X(I-1)) / D1
790 LET Q = FNB(FNB(U))
800 LET Q = .1*(Q+I-2)
810 RETURN
820 PRINT
830 PRINT "THIS PROGRAM COMPUTES CONFIDENCE LIMITS FOR"
840 PRINT "AN UNKNOWN POPULATION MEAN, BASED ON RANDOM"
850 PRINT "SAMPLE DATA GIVEN.   TO USE, TYPE:"
860 PRINT
870 PRINT "   50  DATA  (SIZE OF POPULATION)"
880 PRINT "       (OMIT THIS INPUT IF INFINITE POP'N)"
890 PRINT "   100 DATA  X(1), X(2),.....,X(N)"
900 PRINT
910 PRINT "WHERE THE X(I) ARE THE SAMPLE OBSERVATIONS."
9999 END
```

```
CONFIDENCE 2

100 DATA 2,3,6,9,9,11,13,16,23,45,67,99,35,67,13,43,21,67,90,7,6

RUN


VALUES OF SAMPLE STATISTICS:

    SIZE OF SAMPLE                  21
    SAMPLE MEAN VALUE               30.90476
    VARIANCE OF SAMPLE              857.0386
    SAMPLE STD DEVIATION            29.27522
    ESTIMATED POPN STD DEV          29.99817
    STANDARD ERROR OF MEAN          6.546138

CONFIDENCE LIMITS ON POPULATION MEAN:

CONF LEVEL      LOWER LIM       UPPER LIM

        50        26.40788        35.40164
        75        23.14931        38.66021
        90        19.61534        42.19419
        95        17.25186        44.55766
        99        12.28898        49.52055
      99.9        5.752042        56.05748
     99.99       -0.6232984       62.43281
    99.999       -7.094567        68.90409
```

CORRELATIONS:

DESCRIPTION

This program will perform auto or cross correlations on data. It plots the results as program output. It will also function for correlations on equations. This will show all points where the two equations have equal components.

USERS

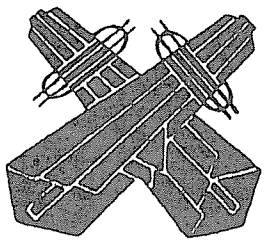
Anyone who would like to compare data sets for subdued characteristics. This would include engineers, statisticians, experimenters, as well as hobbyists to name just a few.

INSTRUCTIONS

If the input is discrete data points, type RUN. The program will prompt for all necessary data. If the input is a function, as in the accompanying example: X(I) = f(I), they must be entered in line 3220 and Y(I) = f(X(I)) must be entered in line 3400. Then type RUN.

LIMITATIONS

The source code for this program will require 6K Bytes of memory for storage. With the DIM statements in line 3070 set at 2200 it will require 50K Bytes of on line memory for execution. If the DIM statements are reduced to 200 then it will execute in 11K Bytes of memory. This will limit the number of data sets to a maximum of 200. In the examples: X(I) = RND(1) and then X(I) for I = 1 to N, step 3 is set equal to 0.5 and Y(I+5) = X(I). In the accompanying plots the correlation at multiples of 3 and 5 are distinctly visible.



NOTE: In these examples the X data is represented by a pulse train, with the pulses Ø.5 unit in amplitude, immersed in noise equal to or greater then the signal and at up to three times the pulse frequency. This signal is very difficult to detect as the S/N is Ø dB. The Y data is the same signal as the X data except it has been displaced to the right by five time or unit intervals. The correlation between these two signals, each buried in noise, is an extreme test of the sensitivity of this program.

```
3010 PRINT "THIS PROGRAM DOES AUTOCORRELATIONS (1), OR"
3020 PRINT "CROSSCORRELATIONS (2) ON DATA SETS AND"
3030 PRINT "PLOTS THE NORMALIZED RESULTS."
3040 PRINT "TYPE IN THE # OF THE CORREL. YOU WISH TO DO. (1 OR 2)"
3050 INPUT A1
3060 IF A1>2 GOTO 3040
3070 DIM X(2200),Y(2200)
3080 PRINT
3090 PRINT "INPUT N THE # OF DATA POINTS YOU HAVE."
3100 INPUT N
3110 PRINT
3120 PRINT "HOW MANY TIME LAGS DO YOU WANT TO TAKE OVER THIS DATA?(<150)"
3130 INPUT M
3140 PRINT
3150 M=M+1
3160 R=T9=A=B=C=D=E=0
3170 MAT X=ZER
3180 MAT Y=ZER
3190 PRINT
3200 PRINT "INPUT YOUR X DATA NOW."
3210 FOR I=1 TO N
3220 INPUT X(I)
3230 A=A+X(I)
3240 REM A IS THE MEAN OF X
3250 IF ABS(X(I))<ABS(B) GOTO 3280
3260 B=X(I)
3270 REM B IS THE LARGEST X VALUE
3280 NEXT I
3290 A=A/N
3300 FOR I=1 TO N
3310 X(I)=X(I)-A
3320 NEXT I
3330 B=B-A
3340 PRINT "  THE MEAN FOR THE X DATA IS ";A
3350 PRINT
3360 IF A1=1 GOTO 3540
3370 PRINT
3380 PRINT "INPUT ,N POINTS, YOUR Y DATA NOW."
3390 FOR I=1 TO N
3400 INPUT Y(I)
3410 C=C+Y(I)
3420 REM C IS THE MEAN OF THE Y DATA
```

```
3430 IF ABS(Y(I))<ABS(D) GOTO 3460
3440 D=Y(I)
3450 REM D IS THE LARGEST Y
3460 NEXT I
3470 C=C/N
3480 FOR I=1 TO N
3490 Y(I)=Y(I)-C
3500 NEXT I
3510 D=D-C
3520 PRINT "  THE MEAN FOR THE Y DATA IS ";C
3530 PRINT
3540 REM THE DATA IS READY TO RUN
3550 REM *** THIS IS THE PLOT SUB. ***
3590 A8=(M-1)/N
3600 LET J=0
3610 LET P7=0
3620 REM A8 IS THE DELAY TIME FOR A 1 SEC. RECORD
3630 A8=INT(A8*(1000)+.1)/1000
3640 T8=0
3650 GOTO 3980
3660 E=0
3670 IF A1=2 GOTO 3770
3680 FOR I=1 TO (N-R)
3690 FOR R1=1 TO (R+1)
3700 E=E+(X(I)*X(I+R1-1))
3710 NEXT R1
3720 NEXT I
3730 IF B<>0 GOTO 3750
3740 B=1
3750 Y=E/((N-R)*R1*(B**2))
3760 GOTO 3910
3770 REM THIS IS FOR CROSSCORRELATIONS
3780 FOR I=1 TO (N-R)
3790 FOR R1=1 TO (R+1)
3800 IF T9<>0 GOTO 3830
3810 E=E+(X(I)*Y(I+R1-1))
3820 GOTO 3840
3830 E=E+(Y(I)*X(I+R1-1))
3840 NEXT R1
3850 NEXT I
3860 IF B<>0 GOTO 3880
3870 B=1
3880 IF D<>0 GOTO 3900
3890 D=1
3900 Y=E/((N-R)*R1*(B*D))
3910 IF T8=1 GOTO 3940
3920 T8=1
3930 F=Y
3940 Y=Y/F
3950 REM IF Z=01 IT DRAWS THE RIGHT SIDE BORDER
3960 Z=0
```

```
3970 LET Y1=Z
3980 LET T=0
3990 LET T1=0
4000 IF J=0 THEN 4090
4010 RETURN
4020 REM Q0 & Q1 ARE LOWER AND UPPER LIMITS FOR THE Y AXIS
4030 REM Q2 & Q3 ARE MIN AND MAX FOR THE X AXIS
4040 GOTO 4090
4050 PRINT
4060 PRINT "INPUT THE NEW Y LIMITS. -MAX. THEN -MIN."
4070 INPUT Q1,Q0
4080 GOTO 4160
4090 Q0=-1
4100 Q1=1.5
4110 Q2=0
4120 Q3=(M-1)
4130 IF A1=1 GOTO 4160
4140 Q0=-2
4150 Q1=3
4160 LET Q5=(Q1-Q0)/50
4170 LET Q6=0
4180 M2=M-1
4190 REM M2 IS THE # OF X POINTS PLOTTED
4200 Q4=(Q3-Q2)/M2
4210 LET J=2
4220 LET K=0
4230 Q9=Q3+Q4
4240 FOR R=Q2 TO Q9 STEP Q4
4250 GOSUB 3660
4260 IF Y1<>0 GO TO 4280
4270 LET Y1=0
4280 IF Q6=0 THEN 4710
4290 IF Q6 =20 THEN 4390
4300 LET K=K+1
4310 IF K=15 GO TO 4350
4320 IF K=17 GO TO 4370
4330 PRINT "                ";"I";
4340 GO TO 4410                  .
4350 PRINT " DELAY TIME    ";"I";
4360 GOTO 4410
4370 PRINT "(";A8;")";"       I";
4380 GO TO 4410
4390 PRINT "                ";"X";
4400 LET Q6 = 10
4410 IF Y>Q1 GOTO 4650
4420 IF Y1>Q1 GOTO 4630
4430 IF Y<Q0 GOTO 4670
4440 IF Y1<Q0 GOTO 4690
4450 LET Q7 = Q0 + 2*Q5
4460 LET Z=Q7+(Q5/2)
4470 IF T>0 THEN 5160
```

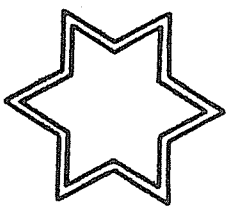
```
4480 IF Z<Y THEN 5150
4490 IF T1>0 THEN 4520
4500 IF Z<Y1 THEN 4520
4510 GOTO 5270
4520 IF Z-Y>=2*Q5 THEN 4580
4530 IF Z-Y>=Q5 THEN 4560
4540 PRINT "  *";
4550 GOTO 4840
4560 PRINT " * ";
4570 GOTO 4840
4580 PRINT "*  ";
4590 GOTO 4840
4600 PRINT "   ";
4610 LET Q7= Q7+ 3*Q5
4620 GOTO 4460
4630 Y1=Q1
4640 GOTO 4450
4650 Y=Q1
4660 GOTO 4450
4670 Y=Q0
4680 GOTO 4450
4690 Y1=Q0
4700 GOTO 4450
4710 PRINT
4720 IF P7=99 THEN 4770
4730 PRINT
4740 REM Y IS PLOTTED '*' AND Z IS PLOTTED '+' AND  0 IS COMMON
4750 PRINT
4760 PRINT
4770 PRINT "THE CORRELATION FUNCTION IS PLOTTED AS (*)"
4780 PRINT "(0) IS COMMON DATA POINT"
4790 PRINT
4800 PRINT "    ";Q0,"    ( Y INCREMENT =";Q5;")";"                "Q1
4810 PRINT
4820 PRINT Q2,"I----+----I----+----I----+----I----+----I----+----I"
4830 GOTO 4390
4840 LET T= 1
4850 IF T1=0 THEN 4610
4860 LET Q6=Q6+1
4870 IF INT(Z)>Q1 THEN 4890
4880 PRINT
4890 NEXT R
4900 LET P7=99
4910 PRINT M,"I----+----I----+----I----+----I----+----I----+----I"
4920 IF T9=0 GOTO 4950
4930 PRINT "( - ";Q3;") REVERSE TIME LAGS."
4940 GOTO 4960
4950 PRINT "(";Q3;") TOTAL TIME LAGS"
4960 PRINT
4970 IF A1=2 GOTO 5110
4980 PRINT
```

```
4990 PRINT
5000 PRINT
5010 PRINT "WOULD YOU LIKE TO SPECIFY NEW Y LIMITS? (YES OR NO)"
5020 INPUT D$
5030 IF D$="Y" GOTO 4050
5040 IF D$="YES" GOTO 4050
5050 PRINT
5060 PRINT "WOULD YOU LIKE TO RUN NEW DATA?"
5070 INPUT D$
5080 IF D$="Y" GOTO 3000
5090 IF D$="YES" GOTO 3000
5100 GOTO 5570
5110 IF T9<>0 GOTO 4980
5120 PRINT "IF YOU WANT THE NEGATIVE SIDE TYPE 1, ELSE TYPE A 0."
5130 INPUT T9
5140 GOTO 5530
5150 IF T1>0 THEN 4600
5160 IF Z<Y1 THEN 4600
5170 IF Z-Y1>=2*Q5 THEN 5230
5180 IF Z-Y1>=Q5 THEN 5210
5190 PRINT "  +";
5200 GOTO 5240
5210 PRINT " + ";
5220 GOTO 5240
5230 PRINT "+  ";
5240 LET T1=1
5250 IF T>0 THEN 4860
5260 GOTO 4610
5270 IF Z-Y>=2*Q5 THEN 5450
5280 IF Z-Y>=Q5 THEN 5370
5290 IF Z-Y1>=2*Q5 THEN 5350
5300 IF Z-Y1>=Q5 THEN 5330
5310 PRINT "  O";
5320 GOTO 4860
5330 PRINT " +*";
5340 GOTO 4860
5350 PRINT "+ *";
5360 GOTO 4860
5370 IF Z-Y1>=2*Q5 THEN 5430
5380 IF Z-Y1>=Q5 THEN 5410
5390 PRINT " *+";
5400 GOTO 4860
5410 PRINT " O";
5420 GOTO 4860
5430 PRINT "+*";
5440 GOTO 4860
5450 IF Z-Y1>=2*Q5 THEN 5510
5460 IF Z-Y1>=Q5 THEN 5490
5470 PRINT "* +";
5480 GOTO 4860
5490 PRINT "*+";
5500 GOTO 4860
5510 PRINT "O";
5520 GOTO 4860
5530 IF T9=1 GOTO 3550
5540 GOTO 4980
5550 PRINT
5560 PRINT "THIS PROGRAM WILL SIMULTANEOUSLY PLOT TWO FUNCTIONS"
5570 END
```

```
THIS PROGRAM DOES AUTOCORRELATIONS (1), OR
CROSSCORRELATIONS (2) ON DATA SETS AND
PLOTS THE NORMALIZED RESULTS.
TYPE IN THE # OF THE CORREL. YOU WISH TO DO. (1 OR 2)
?1

INPUT N THE # OF DATA POINTS YOU HAVE.
?129

HOW MANY TIME LAGS DO YOU WANT TO TAKE OVER THIS DATA?(<150)
?13


INPUT YOUR X DATA NOW.
  THE MEAN FOR THE X DATA IS     .4835869




THE CORRELATION FUNCTION IS PLOTTED AS (*)
(0) IS COMMON DATA POINT

                -1        ( Y INCREMENT =  .05 )                    1.5

             0   I----+----I----+----I----+----I----+----I----+----I
                X                    +                    *
                I                    +        *
                I                    +     *
                I                    +       *
                I                    +     *
                I                    +    *
                I                    +     *
                I                    +    *
                I                    +   *
                I                    +    *
                X                    +    *
                I                    +   *
                I                    +    *
                I                    +   *
                I                    +   *
           129   I----+----I----+----I----+----I----+----I----+----I
(    13 ) TOTAL TIME LAGS



WOULD YOU LIKE TO SPECIFY NEW Y LIMITS? (YES OR NO)
?N

WOULD YOU LIKE TO RUN NEW DATA?
?N
```

```
THIS PROGRAM DOES AUTOCORRELATIONS (1), OR
CROSSCORRELATIONS (2) ON DATA SETS AND
PLOTS THE NORMALIZED RESULTS.
TYPE IN THE # OF THE CORREL. YOU WISH TO DO. (1 OR 2)
?2

INPUT N THE # OF DATA POINTS YOU HAVE.
?128

HOW MANY TIME LAGS DO YOU WANT TO TAKE OVER THIS DATA?(<150)
?11


INPUT YOUR X DATA NOW.
  THE MEAN FOR THE X DATA IS     .4835969


INPUT ,N POINTS, YOUR Y DATA NOW.
  THE MEAN FOR THE Y DATA IS     .0013852




THE CORRELATION FUNCTION IS PLOTTED AS (*)
(0) IS COMMON DATA POINT

              -2        ( Y INCREMENT =    .1 )                    3

          0   I----+----I----+----I----+----I----+----I----+----I
              X                    +         *
              I          *         +
              I                    O
              I               *    +
              I     *              +
              I                    +                              *
              I                    +                        *
              I                    +                   *
              I                    +                 *
              I                    +            *
              X                    +             *
              I                    +             *
              I                    +              *
        128   I----+----I----+----I----+----I----+----I----+----I
(    11 ) TOTAL TIME LAGS

IF YOU WANT THE NEGATIVE SIDE TYPE 1, ELSE TYPE A 0.
?
```

```
THE CORRELATION FUNCTION IS PLOTTED AS (*)
(0) IS COMMON DATA POINT

              -2        ( Y INCREMENT =    .1 )                    3

          0   I----+----I----+----I----+----I----+----I----+----I
              X                    +         *
              I                    +        *
              I                    +               *
              I                    + *
              I                    +       *
              I                    +      *
              I                    +      *
              I                    +            *
              I                    +      *
              I                    +     *
              X                    +      *
              I                    +   *
              I                    +    *
        128   I----+----I----+----I----+----I----+----I----+----I
( - 11 ) TOTAL TIME LAGS


WOULD YOU LIKE TO SPECIFY NEW Y LIMITS? (YES OR NO)
```



CURVE:

DESCRIPTION

This program performs a least squares curve fit for six (6) characteristic curves. The results are tabulated for each curve along with it's index of determination. The six characteristic curves are: Y=X/(A+B*X); Y=1/(A+B*X); Y=A+(B/X); Y=A*(X↑B); Y=A*EXP(B*X); and Y=A+(B*X).

USERS

Anyone could use this program who would like to plot or reduce data. This would include engineers, financial analysts, accountants, statisticians, etc. to name a few.

INSTRUCTIONS

Enter your X data sequentially in the data statements starting in line 100, as follows:

```
          100  DATA    X1,X2,X3,X4,.........
and for Y
          200  DATA    Y1,Y2,Y3,Y4,.........
```

After the data is entered type RUN. The program will prompt for additional input data. For detailed program information list the program. The value of A and B are listed for each of the six curves.

LIMITATIONS

Line 710 contains a Restore statement. The program will require 4K Bytes to store and 13K Bytes of memory for execution. By reducing the DIM statements in line 705 from 200 to a lessor number the program will require less space for execution.

CURVE

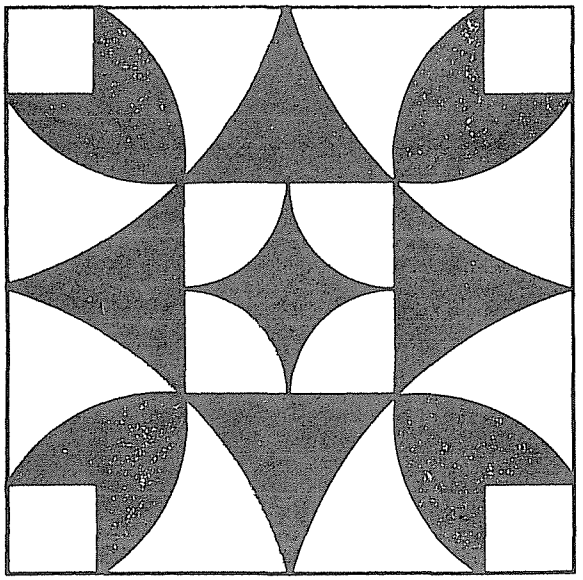
```
100DATA1E38,1E38
101DATA0,0
702 REM
703PRINT
704READP
705DIMX(200),Y(200),U(200),V(200),A(6),B(6),C(6),S(6),F(6)
706IFP=1E38THEN911
707FORK=1TO6
708LETF(K)=1
709NEXTK
710RESTORE
711PRINT
712PRINT"PLEASE SPECIFY THE NUMBER OF VALUES (N) GIVEN AS DATA"
713PRINT"FOR THE TWO INPUT VARIABLES, AND THE OUTPUT CODE (D)."
714PRINT"(D=1 IF OUTPUT IS TO BE IN ORDER OF INCREASING VALUES"
715PRINT"OF THE INDEPENDENT VARIABLE, ELSE D=0).  N,D = ";
716INPUTN,D
717PRINT
718FORI=1TON
719READY(I)
720NEXTI
721FORI=1TON
722READX(I)
723NEXTI
724READX,Y
725 IF X < 1E38  THEN  916
726 IF Y < 1E38  THEN  916
727PRINT
728PRINT
730 PRINT " L E A S T   S Q U A R E S   C U R V E   F I T"
731PRINT
732 PRINT"CURVE TYPE","  INDEX  OF","        A","        B"
733PRINT" ","DETERMINATION"
734PRINT
735FORI=1TO6
736FORI1=1TO6
737LETS(I1)=0
738NEXTI1
739GOSUB844
740IF(I-5)*(I-6)=0THEN755
741IF(I-2)*(I-3)=0THEN748
742FORJ=1TON
743LETU(J)=Y(J)
744GOSUB822
```

```
745NEXTJ
746IFI=1THEN765
747GOTO776
748FORJ=1TON
749IFY(J)<=0THEN762
750LETU(J)=LOG(Y(J))
751GOSUB822
752NEXTJ
753IFI=3THEN770
754GOTO765
755FORJ=1TON
756IFY(J)=0THEN762
757LETU(J)=1/Y(J)
758GOSUB822
759NEXTJ
760IFI=6THEN776
761GOTO765
762PRINT"CAN'T FIT"
763LETF(I)=0
764GOTO783
765FORJ=1TON
766LETU(J)=X(J)
767GOSUB825
768NEXTJ
769GOTO781
770FORJ=1TON
771IFX(J)<=0THEN762
772LETU(J)=LOG(U(J))
773GOSUB825
774NEXTJ
775GOTO781
776FORJ=1TON
777IFX(J)=0THEN762
778LETU(J)=1/X(J)
779GOSUB825
780NEXTJ
781GOSUB886
782PRINTC(I),A(I),B(I)
783NEXTI
784IFD<>1THEN786
785GOSUB829
786PRINT
787PRINT
788PRINT
789PRINT"DETAILS FOR";
790INPUTI
791LETK=I
792LETD1=D
793PRINT
794IFF(I)=1THEN798
795GOSUB844
```

```
796PRINT" COULD NOT BE FIT."
797GOTO786
798GOSUB862
799IF(I-1)*(I-5)*(I-6)<>0THEN810
800FORJ=1TON
801LETY=A(I)+B(I)*X(J)
802IFI=1THEN806
803LETY=1/Y
804IFI=5THEN806
805LETY=X(J)*Y
806GOSUB901
807NEXTJ
808LETD=D1
809GOTO786
810FORJ=1TON
811IFI=2THEN817
812IFI=3THEN815
813LETY=A(4)+B(4)/X(J)
814GOTO818
815LETY=A(3)*(X(J)↑B(3))
816GOTO818
817LETY=A(2)*EXP(B(2)*X(J))
818GOSUB901
819NEXTJ
820LETD=D1
821GOTO786
822LETS(5)=S(5)+U(J)↑2
823LETS(3)=S(3)+U(J)
824RETURN
825LETS(1)=S(1)+U(J)
826LETS(2)=S(2)+U(J)↑2
827LETS(4)=S(4)+U(J)*U(J)
828RETURN
829FORI=1TON-1
830LETM=I
831FORJ=I+1TON
832IFX(M)<=X(J)THEN834
833LETM=J
834NEXTJ
835IFM=ITHEN842
836LETP=X(M)
837LETQ=Y(M)
838LETX(M)=X(I)
839LETY(M)=Y(I)
840LETX(I)=P
841LETY(I)=Q
842NEXTI
843RETURN
844LETK=I
845IFK=1THEN860
846IFK=2THEN858
```

```
847IFK=3THEN856
8481FK=4THEN854
849IFK=5THEN852
850PRINT"6. Y=X/(A+B*X) ";
851RETURN
852PRINT"5. Y=1/(A+B*X) ";
853RETURN
854PRINT"4. Y=A+(B/X)";
855RETURN
856PRINT"3. Y=A*(X↑B)";
857RETURN
858PRINT"2. Y=A*EXP(B*X)";
859RETURN
860PRINT"1. Y=A+(B*X)";
861RETURN
862PRINT"    ";
863GOSUB845
864PRINT" IS A";
865IFK=1THEN870
866IFK=2THEN872
867IFK=3THEN874
868PRINT" HYPERBOLIC";
869GOTO875
870PRINT" LINEAR";
871GOTO875
872PRINT"N EXPONENTIAL";
873GOTO875
874PRINT" POWER";
875PRINT" FUNCTION.  THE RESULTS"
876IFK=1THEN878
877PRINT"      OF A LEAST-SQUARES FIT OF ITS LINEAR TRANSFORM"
878IFD<>1THEN880
879PRINT"       (SORTED IN ORDER OF ASCENDING VALUES OF X)"
880PRINT"      ARE AS FOLLOWS:"
881PRINT
882 PRINT "        X-ACTUAL        Y-ACTUAL     Y-CALC       PCT DIFFER"
883PRINT
884RETURN
885PRINT
886LETB=(N*S(4)-S(1)*S(3))/(N*S(2)-(S(1)↑2))
887LETA=(S(3)-B*S(1))/N
888LETS1=S(5)-(S(3)↑2)/N
889LETS2=(B↑2)*(S(2)-(S(1)↑2)/N)
890LETC(I)=S2/S1
891IF(I-1)*(I-4)*(I-5)=0THEN898
892IF(I-2)*(I-3)=0THEN896
893LETA(6)=B
894LETB(6)=A
895RETURN
896LETA(I)=EXP(A)
897GOTO899
```

```
898LETA(I)=A
899LETB(I)=B
900RETURN
901PRINTX(J),Y(J),Y,
902LETD=Y(J)-Y
903LETD=.1*SGN(D)*INT(1000*ABS(D/Y))
904IFD<0THEN909
905IFD>0THEN908
906PRINT"       0"
907RETURN
908PRINT"         ";
909PRINTD
910RETURN
911 PRINT

913 PRINT"----"
914 PRINT
915 STOP
916 PRINT
917 PRINT"CHECK INPUT"
9999 END
```



DIFFERENCES:

DESCRIPTION

This program analyzes the differences between the means of two sets of data. The variances for these two sets do not have to be equal.

USERS

This program will find most of it's use among individuals who use statistics frequently in their field.

INSTRUCTIONS

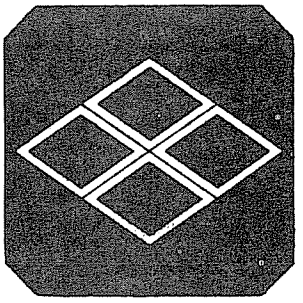
Enter the data starting in line 100 before the program is run. The format is:

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

List the program for detailed program instructions. If the data has not been summarized, enter the raw data starting in line 100. Be sure to put data in line 101 or the program may not run properly. List lines 675 to 885 for additional information on entering raw data.

LIMITATIONS

This program requires 4K Bytes for storage and should execute in 7K Bytes of memory in most systems.

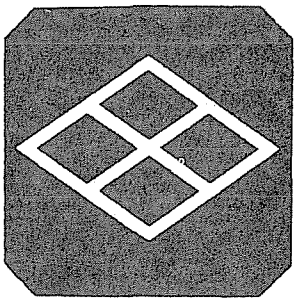


DIFFERENCE 5

```
17 DIM Y(200)
100 GO TO 675
101 GO TO 440
199 GO TO 300
200 DATA .50,.75,.90,.95,.999,.9999,.99999
299 DATA 10
300 READ N1
305 LET Y1=0
310 LET N3=N1-1
315 FOR I=1 TO N1
320 READ Y(I)
325 LET Y1=Y1+Y(I)
330 NEXT I
335 LET Y3=Y1/N1
340 LET Y1=0
345 FOR I=1 TO N1
350 LET Y1=Y1+(Y(I)-Y3)*(Y(I)-Y3)
355 NEXT I
360 LET S1=Y1/N1
365 READ N2
370 LET Y1=0
375 LET N4=N2-1
380 FOR I=1 TO N2
385 READ Y(I)
390 LET Y1=Y1+Y(I)
395 NEXT I
400 LET Y4=Y1/N2
405 LET Y1=0
410 FOR I=1 TO N2
415 LET Y1=Y1+(Y(I)-Y4)*(Y(I)-Y4)
420 NEXT I
425 LET S2=Y1/N2
430 LET Y5=Y3-Y4
435 GO TO 470
440 READ Y,N1,Y3,S1,Z,N2,Y4,S2
445 LET N3=N1-1
450 LET N4=N2-1
455 LET Y5=Y3-Y4
460 LET S1=S1*S1
465 LET S2=S2*S2
470 PRINT TAB(8);"STATISTIC";TAB(30);"SAMPLE 1";TAB(45);"SAMPLE 2"
475 PRINT
```

```
480 PRINT "SAMPLE MEAN";TAB(30);Y3,Y4
485 PRINT "SAMPLE STD DEVIATION";TAB(30);SQR(S1),SQR(S2)
490 PRINT "SAMPLE SIZE";TAB(30);N1,N2
495 PRINT "SAMPLE VARIANCE";TAB(30);S1,S2
500 PRINT
505 PRINT "DIFF OF MEANS";TAB(37);Y5
510 PRINT
515 PRINT
520 PRINT "HIGHEST POSTERIOR DENSITY INTERVALS"
521 PRINT "DIFFERENCE BETWEEN MEANS:"
525 PRINT
530 PRINT" HPD LEVEL","LOWER LIM","UPPER LIM"
535 PRINT
540 READ P
545 IF P=10 THEN 885
550 LET P=(1-P)/2
555 LET T=SQR(LOG(1/P↑2))
560 LET X=2.515517+0.802853*T+0.010328*T*T
565 LET X=T-X/(1+1.432788*T+0.189269*T*T+0.001308*T*T*T)
570 LET G1=X*(1+X*X)/4
575 LET G2=X*(1+16*X*X/3*(1+5*X*X/16))/32
580 LET G3=-5*X*(1-17*X*X/15*(1+19*X*X/17*(1+3*X*X/19)))/128
585 LET G4=1-1482*X*X/1920*(1+776*X*X/1482*(+79*X*X/776))
590 LET G4=-945*X*(1+1920*X*X/245*G4)/92160
595 LET S4=S2/N2
600 LET S3=S1/N1
605 LET C1=S4/(S3+S4)
610 LET C2=S3/(S3+S4)
615 LET N6=N4/(N4-2)
620 LET N5=N3/(N3-2)
625 LET F=(N6*C1+N5*C2)*(N6*C1+N5*C2)
630 LET F=F/((N6*N6*C1*C1)/(N4-4)+(N5*N5*C2*C2)/(N3-4))
635 LET B=F+4
640 LET A=(B-2)*(N6*C1+N5*C2)/B
645 LET T5=X+G1/B+G2/B↑2+G3/B↑3+G4/B↑4
650 LET T6=SQR(A*(S3+S4))*T5
655 LET Y6=Y5-T6
660 LET Y9=Y5+T6
665 PRINT (1-2*P)*100,Y8,Y9
670 GO TO 540
675 PRINT " THIS PROGRAM COMPUTES THE HIGHEST POSTERIOR DENSITY "
680 PRINT "INTERVALS FOR THE DIFFERENCE BETWEEN TWO POPULATION MEANS"
685 PRINT "USING BAYESIAN STATISTICAL INFERENCE.  THE DATA MAY BE"
690 PRINT "ENTERED IN EITHER OF TWO WAYS:"
695 PRINT
700 PRINT "     A. TYPE THE DATA ON ONE LINE AS:"
705 PRINT
710 PRINT "       100 DATA M1,N1,M1,S1,M2,N2,M2,S2"
715 PRINT "       RUN"
720 PRINT
725 PRINT "     WHERE,"
```

```
730 PRINT "        N1=DUMMY TO AGREE WITH CONDIF*** INPUT"
735 PRINT "           (ENTER '0' AS AN APPROPRIATE NUMBER)"
740 PRINT "        N1=SIZE OF SAMPLE 1"
745 PRINT "        M1=ARITHMETIC MEAN OF SAMPLE 1"
750 PRINT "        S1=STANDARD DEVIATION OF SAMPLE 1(BASED ON"
755 PRINT "           DIVISOR OF N1)"
760 PRINT
765 PRINT "      N2,N2,M2,S2, ARE THE SAME FOR SAMPLE 2"
770 PRINT
775 PRINT "     B. TYPE THE DATA ON MORE THAN ONE LINE AS:"
780 PRINT
785 PRINT "       100 DATA N1,Y(1),Y(2),Y(3),Y(4)"
790 PRINT "       101 DATA Y(5),   . . .    ,Y(N1)"
795 PRINT "       102 DATA N2,Z(1),Z(2),Z(3),Z(4)"
800 PRINT "       103 DATA Z(5),   . . .    ,Z(N2)"
805 PRINT "       RUN"
810 PRINT
815 PRINT "     WHERE,"
820 PRINT "       N1=THE SIZE OF SAMPLE 1"
825 PRINT "       N2=THE SIZE OF SAMPLE 2"
830 PRINT "       Y(I),I=1 TO N1  ARE THE ELEMENTS OF SAMPLE 1"
835 PRINT "       Z(I),I=1 TO N2  ARE THE ELEMENTS OF SAMPLE 2"
840 PRINT
845 PRINT
850 PRINT " IF DIFFERENT HIGHEST POSTERIOR DENSITY INTERVALS ARE"
855 PRINT "DESIRED, THEN THEY MAY BE ENTERED AS:"
860 PRINT
865 PRINT "       200 DATA L1,L2,L3,L4,L5,..."
870 PRINT
875 PRINT "       WHERE ALL L=L1,L2,L3,... ARE SUCH THAT"
880 PRINT "            0.5< L < 1.0"
885 STOP
890 END
```



```
DIFFERENCES


100 DATA 0,20,67,4.5,0,25,59,3.9


RUN

                STATISTIC              SAMPLE 1          SAMPLE 2

SAMPLE MEAN                            67                59
SAMPLE STD DEVIATION                   4.5               3.9
SAMPLE SIZE                            20                25
SAMPLE VARIANCE                        20.25             15.21

DIFF OF MEANS                           8

HIGHEST POSTERIOR DENSITY INTERVALS
DIFFERENCE BETWEEN MEANS:

   HPD LEVEL          LOWER LIM          UPPER LIM

         50                  0           8.885841
         75                  0           9.521447
         90                  0           10.19852
         95                  0           10.64231
       99.9                  0           12.68091
      99.99                  0           13.72642
     99.999                  0           14.73568
```

DUAL PLOT:

DESCRIPTION

Dual Plot will plot two functions of a single variable. The plot may be done on a standard teletype or terminal without the need for a special plotter.

USERS

Anyone who would like to plot equations could use this program. The equations could be generated from one of the curve fitting programs included in this section.

INSTRUCTIONS

Enter the first equation in line 10 and the second equation in line 20, as follows:

```
10 LET  Y = f(X)
20 LET  Z = f(X,Y)
```

Then type RUN. If only Y = f(X) is desired, then let Z = Ø in line 20 or vice versa. For additional instructions list the program.

LIMITATIONS

Four K Bytes of on line memory is required by this program for storage and execution.

DUAL PLOT

```
8 LET J=0
9 LET P7=0
10 GO TO 620
100 LET Y1=Z
105 LET T=0
110 LET T1=0
115 IF J=0 THEN 125
120 RETURN
125 PRINT
130 PRINT "WHAT ARE  FMIN,FMAX,XMIN,XMAX,DELX";
135 INPUT Q0,Q1,Q2,Q3,Q4
140 LET Q5=(Q1-Q0)/60
145 LET Q6=0
150 LET J=2
155 FOR X = Q2 TO Q3 STEP Q4
160 GOSUB 10
165 IF Q6=0 THEN 315
170 IF Q6 = 20 THEN 185
175 PRINT " . ";
180 GOTO 195
185 PRINT " - ";
190 LET Q6 = 10
195 IF Y > Q1 THEN 305
200 IF Y1>Q1 THEN 305
205 IF Y < Q0 THEN 305
210 IF Y1<Q0 THEN 305
215 LET Q7 = Q0 + 2*Q5
220 LET Z=Q7+0.5*Q5
225 IF T>0 THEN 435
230 IF Z<Y THEN 430
235 IF T1>0 THEN 250
240 IF Z<Y1 THEN 250
245 GOTO 490
250 IF Z-Y>=2*Q5 THEN 280
255 IF Z-Y>=Q5 THEN 270
260 PRINT "  +";
265 GOTO 370
270 PRINT " + ";
275 GOTO 370
280 PRINT "+  ";
285 GOTO 370
290 PRINT "   ";
295 LET Q7= Q7+ 3*Q5
```

```
300 GOTO 220
305 PRINT "OFF SCALE.  (X,Y,Z) =";X;",";Y;",";Y1
310 GOTO 380
315 PRINT
320 IF P7=99 THEN 335
325 PRINT
330 PRINT "(NOTE: Y IS PLOTTED '+', Z IS '.', AND '0' IS COMMON POINT)"
335 PRINT
340 PRINT "FOR X:        TOP = ";Q2;" BOTTOM = ";Q3;" INCREMENT = ";Q4
345 PRINT "FOR FCTS:   LEFT = ";Q0;"  RIGHT = ";Q1;" INCREMENT = ";Q5
350 PRINT
355 PRINT "   I.........I.........I.........I.........I.........I.........I.........I"
360 PRINT
365 GOTO 185
370 LET T= 1
375 IF T1=0 THEN 295
380 LET Q6=Q6+1
385 IF INT(Z)>Q1 THEN 395
390 PRINT
395 NEXT X
400 LET P7=99
405 PRINT
410 PRINT
415 PRINT "TYPE 'S' TO STOP NOW, OR ELSE SPECIFY NEW VALUES"
420 PRINT "FOR  FMIN,FMAX,XMIN,XMAX,DELX";
425 GOTO 135
430 IF T1>0 THEN 290
435 IF Z<Y1 THEN 290
440 IF Z-Y1>=2*Q5 THEN 470
445 IF Z-Y1>=Q5 THEN 460
450 PRINT "  .";
455 GOTO 475
460 PRINT " . ";
465 GOTO 475
470 PRINT ".  ";
475 LET T1=1
480 IF T>0 THEN 380
485 GOTO 295
490 IF Z-Y>=2*Q5 THEN 580
495 IF Z-Y>=Q5 THEN 540
500 IF Z-Y1>=2*Q5 THEN 530
505 IF Z-Y1>=Q5 THEN 520
510 PRINT "  0";
515 GOTO 380
520 PRINT " .+";
525 GOTO 380
530 PRINT ". +";
535 GOTO 380
540 IF Z-Y1>=2*Q5 THEN 570
545 IF Z-Y1>=Q5 THEN 560
550 PRINT " +.";
```

```
555 GOTO 380
560 PRINT " 0";
565 GOTO 380
570 PRINT ".+";
575 GOTO 380
580 IF Z-Y1>=2*.05 THEN 610
585 IF Z-Y1>=.05 THEN 600
590 PRINT "+ .";
595 GOTO 380
600 PRINT "+.";
605 GOTO 380
610 PRINT "0";
615 GOTO 380
620 PRINT
625 PRINT "THIS PROGRAM WILL SIMULTANEOUSLY PLOT TWO FUNCTIONS"
630 PRINT "OF A SINGLE VARIABLE X.  TO USE, TYPE:"
635 PRINT
640 PRINT "    10 LET Y= (THE FIRST FUNCTION OF X)"
645 PRINT "    20 LET Z= (THE SECOND FUNCTION OF X AND/OR Y)"
650 PRINT "    RUN"
655 PRINT
660 PRINT "THE FUNCTIONS Y AND Z MAY BE ANY LEGITIMATE 'BASIC'"
665 PRINT "EXPRESSIONS.  INTERMEDIATE VARIABLES MAY BE DEFINED"
670 PRINT "USING INTERMEDIATE LINES, IF THE FUNCTIONS ARE TOO"
675 PRINT "COMPLICATED TO FIT ON ONE LINE."
9999 END
```

EXP-DISTRI:

DESCRIPTION

This program calculates confidence limits for data that is exponentially distributed. The number of samples should be greater then eleven.

USERS

Individuals involved in analyzing experimental data will find this program useful.

INSTRUCTIONS

Enter the raw data starting in line 100 before program execution. Use the following format when entering the data:

```
     100  DATA    X(1),X(2),X(3),.....
```

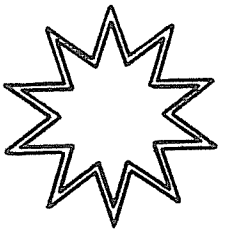
where

X( ) are the random samples or observations.

After the data has been entered type RUN. For additional information on data entry list the program.

LIMITATIONS

Exp-Distri should execute in 3K Bytes of memory in most systems without any problems.



EXP-DISTRI

```
10 PRINT
11 PRINT
12 PRINT
13 PRINT
15 LET I=0
20 LET X1=0
30 LET X2=0
100 DATA -1E38
299 DATA 1E38
300 DATA .5,.6745,.6,.8416,.7,1.036,.8,1.282,.9,1.645,.95,1.960,.98
301 DATA 2.326,.99,2.576,.995,2.807,.999,3.291
305 READ X
310 IF X=1E38 THEN 900
315 IF X=(-1E38) THEN 6000
320 LET I=I+1
330 LET X1=X1+X
340 LET X2=X2+X*X
350 GO TO 305
900 PRINT
1000 PRINT"VALUES OF SAMPLE STATISTICS"
1010 PRINT"------ -- ------ ----------"
1020 PRINT
1030 PRINT"SIZE OF SAMPLE";TAB(30);I
1035 LET X6=X1/I
1040 PRINT"SAMPLE MEAN";TAB(30);X6
1050 LET V1=(I*X2-X1*X1)/(I*(I-1))
1060 PRINT"SAMPLE VARIANCE";TAB(30);V1
1070 PRINT"SAMPLE STD DEVIATION";TAB(30);SQR(V1)
1200 PRINT
1205 IF I<11 THEN 7000
1210 PRINT"CONFIDENCE LIMITS ON POPULATION MEAN"
1215 PRINT"---------- ------ -- ---------- ----"
1220 PRINT
1230 PRINT"CONF LEVEL";TAB(20);"LOWER LIM";TAB(40);"UPPER LIM"
1240 PRINT
1260 FOR C=1 TO 9
1270 READ P,Z
1275 LET Z=Z/SQR(I)
1280 LET A=Z*X6/(1-Z)
1290 LET B=Z*X6/(1+Z)
1300 PRINT P;TAB(20);X6-B;TAB(40);X6+A
1310 NEXT C
```

```
1420 GOTO 9999
6000 PRINT"THIS PROGRAM COMPUTES CONFIDENCE LIMITS FOR AN UNKNOWN"
6010 PRINT"POPULATION MEAN, GIVEN THAT THE POPULATION IS KNOWN"
6020 PRINT"TO BE EXPONENTIALLY DISTRIBUTED."
6030 PRINT"THE CALCULATIONS ARE MADE ON THE ASSUMPTION THAT"
6040 PRINT"AT LEAST ELEVEN SAMPLES ARE TAKEN AT RANDOM FROM THIS"
6045 PRINT"POPULATION.  TO USE,TYPE:"
6090 PRINT
6100 PRINT"     100 DATA X(1),X(2),X(3),....X(N)"
6110 PRINT"        WHERE THE X(I)'S ARE SAMPLE OBSERVATIONS"
6120 GOTO 9999
7000 PRINT"YOUR SAMPLE OF SIZE";I;"IS TOO SMALL TO ASSUME THAT THE"
7010 PRINT"CENTRAL LIMIT THEOREM APPLIES AND THAT YOUR SAMPLE MEANS"
7020 PRINT"ARE NORMALLY DISTRIBUTED. THIS WILL LEAD TO ERRONEOUS"
7030 PRINT"CALCULATIONS OF THE CONFIDENCE LIMITS. YOU SHOULD USE"
7040 PRINT"AT LEAST ELEVEN SAMPLES. SORRY."
9999 END
```

EXP-DISTRIBUTION

```
100 DATA 3.4,3.2,3.5,3.6,2.9,3.4,3.1,3,3.4,3.2,4,2.8
```

RUN

VALUES OF SAMPLE STATISTICS

SIZE OF SAMPLE	12
SAMPLE MEAN	3.291667
SAMPLE VARIANCE	0.1099248
SAMPLE STD DEVIATION	0.3315491

CONFIDENCE LIMITS ON POPULATION MEAN

CONF LEVEL	LOWER LIM	UPPER LIM
0.5	2.755198	4.087561
0.6	2.648272	4.348012
0.7	2.533869	4.696124
0.8	2.402533	5.225544
0.9	2.231834	6.268296
0.95	2.102222	7.581048
0.98	1.969338	10.01902
0.99	1.887827	12.83937
0.995	1.818288	17.35297

LEAST SQUARES:

DESCRIPTION

This program uses the least squares method to fit raw data to one of three curves. The raw data is represented by two variables, X and Y. The three curves are: Y=A*EXP(B*X); Y=A+B*X; and Y=A*(X↑B), exponential, linear and power.

USERS

Experimenters will find this program easy to use and the results easily interpreted.

INSTRUCTIONS

Before Least Squares is run the raw data must be entered into the program. Enter the data into data statements starting in line 2. Use the following format for entering the data:

```
2 DATA    X(1),Y(1),X(2),Y(2),X(3),.....
```

The values of A and B are listed for the curve selected. To select the curve type a 1, 2, or 3 when asked 'which curve'. For additional information list the program.

LIMITATIONS

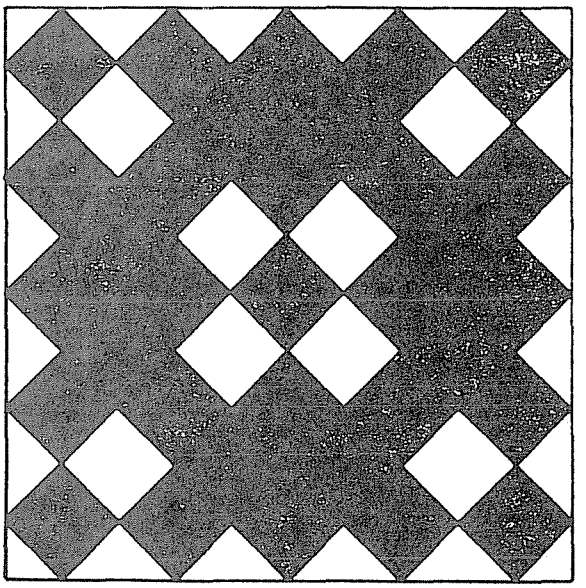
The source code will require 3K Bytes of memory for storage. Execution will require 9K Bytes of available memory. The DIM statement on line 220 sets up a two dimensional array. This array is used through out this program. If the DIM statement is reduced in size the required execution space will also be reduced.

LEAST SQUARES

```
1 GO TO 645

200 DATA 1E35,1E35
205 PRINT "WHAT TYPE OF CURVE DO YOU WANT TO FIT:"
210 PRINT "LINEAR=1, EXPONENTIAL=2, POWER=3";
215 INPUT Q1
220 DIM D(250,2)
225 PRINT
230 LET I=0
235 LET I=I+1
240 READ D(I,1),D(I,2)
245 IF D(I,1)<>1E35 THEN 235
250 LET Q2=I-1
255 LET S9=0
260 IF Q1=1 THEN 280
265 IF Q1=2 THEN 335
270 IF Q1=3 THEN 405
275 GOTO 230
280 LET S9=1
285 GOSUB 505
290 PRINT "LINEAR:  Y=A+B*X  WITH  A=";Q8;"AND B=";Q9
295 GOSUB 580
300 FOR J=1 TO Q2
305 LET W7=Q8+Q9*D(J,1)
310 LET Z7=W7-D(J,2)
315 LET Q4=100*Z7/D(J,2)
320 PRINT D(J,1),D(J,2),W7,Z7,Q4
325 NEXT J
330 STOP
335 FOR J=1 TO Q2
340 LET D(J,2)=LOG(D(J,2))
345 NEXT J
350 GOSUB 505
355 PRINT "EXPONENTIAL:  Y=A*EXP(B*X)  WITH  A=";EXP(Q8);"AND B=";Q9
360 GOSUB 580
365 FOR J=1 TO Q2
370 LET W7=EXP(Q8+Q9*D(J,1))
375 LET W8=EXP(D(J,2))
380 LET Z7=W7-W8
385 LET Q4=100*Z7/W8
390 PRINT D(J,1),W8,W7,Z7,Q4
395 NEXT J
```

```
400 STOP
405 FOR J=1 TO Q2
410 LET D(J,1)=LOG(D(J,1))
415 LET D(J,2)=LOG(D(J,2))
420 NEXT J
425 GOSUB 505
430 PRINT "POWER:  Y=A*(X↑B)  WITH  A=";EXP(Q8);"AND B=";Q9
435 GOSUB 580
440 FOR J=1 TO Q2
445 LET W7=EXP(D(J,1))
450 LET W8=EXP(D(J,2))
455 LET W9=EXP(Q8)*W7↑Q9
460 LET Q4=W9/W8-1
465 LET Z7=W9-W8
470 IF Q4<0 THEN 485
475 LET Q4=INT(1000*Q4+0.5)/10
480 GOTO 490
485 LET Q4=INT(1000*Q4-0.5)/10
490 PRINT W7,W8,W9,Z7,Q4
495 NEXT J
500 STOP
505 LET Q3=0
510 LET Q4=0
515 LET Q5=0
520 LET Q6=0
525 LET Q7=0
530 FOR J=1 TO Q2
535 LET Q3=Q3+D(J,1)
540 LET Q4=Q4+D(J,2)
545 LET Q5=Q5+D(J,1)*D(J,2)
550 LET Q6=Q6+(D(J,1))↑2
555 LET Q7=Q7+(D(J,2))↑2
560 NEXT J
565 LET Q9=(Q2*Q5-Q3*Q4)/(Q2*Q6-Q3↑2)
570 LET Q8=(Q4-Q3*Q9)/Q2
575 RETURN
580 LET Q0=(Q2*Q5-Q3*Q4)/SQR((Q2*Q6-Q3↑2)*(Q2*Q7-Q4↑2))
585 PRINT
590 PRINT "COEFFICIENT OF CORRELATION = ";Q0
595 PRINT "COEFFICIENT OF DETERMINATION = ";Q0↑2
600 PRINT
605 PRINT "DO YOU WANT TO SEE A COMPARISON OF THE ACTUAL Y'S AND"
610 PRINT "THE ESTIMATED Y'S.  1=YES,  0=NO";
615 INPUT P9
620 IF P9<>1 THEN 715
625 PRINT
630 PRINT "X-ACTUAL","Y-ACTUAL","Y-ESTIM","DIFFER","PCT-DIFF"
635 PRINT
640 RETURN
645 PRINT
650 PRINT "THIS IS A LINEAR REGRESSION PROGRAM FOR DATA IN TWO "
```

```
655 PRINT "VARIABLES, X AND Y.  FROM INPUT POINTS, DESCRIBED BY"
660 PRINT "THEIR X AND Y COORDINATES, AN EQUATION IS PRODUCED THAT"
665 PRINT "BEST FITS THESE POINTS IN THE LEAST-SQUARES SENSE.  TO"
670 PRINT "USE THE PROGRAM, TYPE THE FOLLOWING:"
675 PRINT
680 PRINT "   2 DATA  X(1),Y(1), X(2),Y(2),.....,  X(N),Y(N)"
685 PRINT "         (WHERE X(1),Y(1) IS THE FIRST POINT, X(2),"
690 PRINT "         Y(2) IS THE SECOND, AND SO ON UNTIL ALL"
695 PRINT "         POINTS HAVE BEEN ENTERED.  ADDITIONAL DATA"
700 PRINT "         STATEMENTS 3-199 MAY BE USED AS NEEDED.)"
705 PRINT
710 PRINT "THEN TYPE 'RUN'."
715 END
```



PAIRED:

DESCRIPTION

Paired compares two groups of raw data. It uses the Mann-Whitney rank test to compare the two groups.

USERS

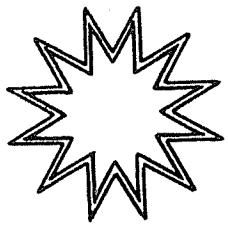
This program best lends itself to individuals in the statistics field.

INSTRUCTIONS

Before running Paired enter the data starting on line 900. List program lines 10 to 87 for instructions on entering your data.

LIMITATIONS

There is MAT Read statements on lines 120 and 150. The source code for this program will store in 2K Bytes of memory. Execution length is 12K Bytes due to the amount of space reserved in the DIM statement on line 90.



PAIRED

```
10    REM  DESCRIPTION: COMPARES TWO GROUPS OF DATA BY MEANS OF
20    REM  THE MANN-WHITNEY TWO SAMPLE RANK TEST.
30    REM  INSTRUCTIONS:  PUT DATA IN LINE 900 AND FOLLOWING.
40    REM  FIRST DATA ARE M, N, AND C.  M AND N ARE THE TWO SAMPLE
50    REM  SIZES, AND C IS THE CRITICAL VALUE FOR THE MANN-WHITNEY
60    REM  TEST WITH FRACTIONAL COUNTS.  IF M OR N IS GREATER THAN 30,
70    REM  THE DIM STATEMENTS IN LINE 90 SHOULD BE CHANGED.
80    REM  AFTER M,N, AND C ARE PUT IN, THE NEXT DATA SHOULD BE
85    REM  THE FIRST SERIES OF TEST DATA, FOLLOWED BY
87    REM  THE SECOND SERIES.
90    DIM A(30), B(30), D(900)
100   READ M, N, C
120     MAT READ A(M)
150     MAT READ B(N)
170   FOR I = 1 TO M
180       LET Q = (I - 1) * N
190       LET X = A(I)
200       FOR J = 1 TO N
210           LET D(Q + J) = X - B(J)
220       NEXT J
230   NEXT I
300   LET P = M * N
310   LET I = 1
320   IF D(I) <= D(I+1) THEN 390
330   LET T = D(I)
340   LET D(I) = D(I+1)
350   LET D(I+1) = T
360   IF I = 1 THEN 390
370   LET I = I - 1
380   GO TO 320
390   LET I = I + 1
400   IF I < P THEN 320
500   LET C = C + 1
510   LET X = INT(C)
520   LET R = C - X
530   LET L = D(X) * (1 - R) + D(X+1) * R
540   LET X = P - X
550   LET U = D(X+1) * (1 - R) + D(X) * R
560   PRINT "CONFIDENCE INTERVAL BY RANK SUM TEST."
570   PRINT "LOWER LIMIT =  ";L, "UPPER LIMIT =  ";U
580   STOP
9999  END
```

```
PAIRED


900 DATA 4,6,2.5
910 DATA 190,160,160,140
920 DATA 117,120,120,145,147,150


RUN


CONFIDENCE INTERVAL BY RANK SUM TEST.
LOWER LIMIT =          2.5      UPPER LIMIT =     57.5
```

The critical value C is composed of two parts. The integer number and it's decimal. The decimal fraction determines the symmetry of the confidence calculations about the two groups of data. The fraction 'Ø.5' is used to generate equal symmetry. The integer number, which should be between Ø and (M*N-1), is used to set the calculation point about both ends of the data groups; as the confidence interval is calculated about the lower portion of the data and the upper protion of the data. A value of (M*N)/2 for the integer would yield a confidence interval about the middle of the two data groups. For this condition C would be equal to (((M*N)/Z + .5). A value of '2.5' for C would be used to determine the confidence interval about the data end points.

PLOT:

DESCRIPTION

This versatile plotting program will plot from one to six functions of a single variable simultaneously. The plotting may be performed on any standard teletype or terminal.

USERS

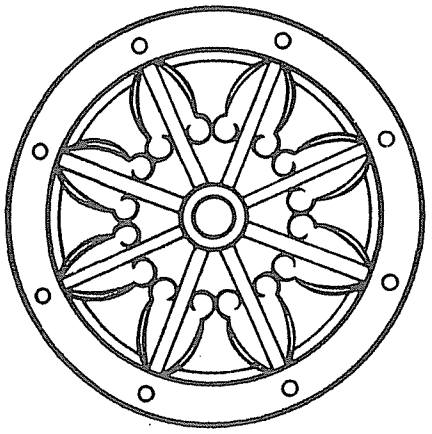
Experimenters should find this program very useful for displaying a variety of functions. These functions are all plotted on the same scale so these functions can be compared over the plotting interval.

INSTRUCTIONS

Before the program is run the equations must be entered starting in line 100 of the program. For detailed information about the data entry list program lines 3200 to 3560.

LIMITATIONS

This program should execute without incident in most systems with 4K Bytes of available memory.



PLOT

```
100  GO TO 3200
2030 READ A1,H,I,J,E1,F1
2045 LET G=(F1-E1)/40
2060 LET G1=G/2
2075 PRINT
2090 PRINT" ","MULTIPLE PLOT OF THE FUNCTIONS ";
2105 FOR K=1TO A1
2120 GOSUB 3575
2135 IF K=A1THEN 2180
2150 PRINT", ";
2165 NEXT K
2180 PRINT
2195 PRINT" ","          HORIZ INCREMENT =";G
2210 GOSUB 3020
2225 GOSUB 3050
2240 PRINT"X-VALUE"
2255 LET X=H-J
2270 PRINT
2285 LET L1=0
2300 LET X=X+J
2315 IF X>I THEN 2945
2330 PRINTX,
2345 LET Y=E1-G
2360 PRINT" ";
2375 LET Y=Y+G
2390 IF Y>F1THEN 2270
2405 IF ABS(A-Y)<G1THEN 2585
2420 IF A1<2THEN 2360
2435 IF ABS(B-Y)<G1THEN 2645
2450 IF A1<3THEN 2360
2465 IF ABS(C-Y)<G1THEN 2705
2480 IF A1<4THEN 2360
2495 IF ABS(D-Y)<G1THEN 2765
2510 IF A1<5THEN 2360
2525 IF ABS(E-Y)<G1THEN 2825
2540 IF A1<6THEN 2360
2555 IF ABS(F-Y)<G1THEN 2885
2570 GOTO  2360
2585 PRINT"A";
2600 LET L1=L1+1
2615 IF L1=A1THEN 2270
2630 GOTO  2375
2645 PRINT"B";
2660 LET L1=L1+1
```

```
2675 IF L1=A1THEN 2270
2690 GOTO  2375
2705 PRINT"C";
2720 LET L1=L1+1
2735 IF L1=A1THEN 2270
2750 GOTO  2375
2765 PRINT"D";
2780 LET L1=L1+1
2795 IF L1=A1THEN 2270
2810 GOTO  2375
2825 PRINT"E";
2840 LET L1=L1+1
2855 IF L1=A1THEN 2270
2870 GOTO  2375
2885 PRINT"F";
2900 LET L1=L1+1
2915 IF L1=A1THEN 2270
2930 GOTO  2375
2945 PRINT
2960 GOSUB 3050
2975 GOSUB 3020
2990 PRINT
3005 GOTO  3080
3020 PRINT TAB(5); E1; TAB(45); F1
3035 RETURN
3050 PRINT" "," :....:....:....:....:....:....:....:....:"
3065 RETURN
3080 PRINT
3095 PRINT
3110 PRINT "TYPE 'S' TO STOP, OR GIVE NEW VALUES OF"
3125 PRINT"NUM,XMIN,XMAX,DELX,HMIN,HMAX";
3140 INPUT A1,H,I,J,E1,F1
3155 PRINT
3170 PRINT
3185 GO TO 2045
3200 PRINT
3215 PRINT"THIS PROGRAM PLOTS ONE TO  SIX FUNCTIONS OF X SIMULTANEOUSLY."
3230 PRINT"ALL FUNCTIONS HAVE THE SAME UPPER AND LOWER LIMITS FOR  THE"
3245 PRINT"PLOT.  THE FUNCTIONS ARE CALLED A,B,C,D,E,F AND ARE PLOTTED"
3260 PRINT"IN THAT ORDER OF PRIORITY.  WHERE PLOTS WOULD OVERLAP, THE"
3275 PRINT"LOWER PRIORITY FUNCTIONS ARE SUPPRESSED.  VALUES EXCEEDING"
3290 PRINT"THE SELECTED BOUNDS ARE DISREGARDED.  ENTER INFOR MATION FOR "
3305 PRINT "A RUN IN THE FOLLOWING FORMAT:"
3320 PRINT
3335 PRINT"    100 DATA NUM,XMIN,XMAX,DELX,HMIN,HMAX"
3350 PRINT"    200 LET  A= ANY 'BASIC' FUNCTION OF X"
3365 PRINT"    210 LET  B= ANY 'BASIC' FUNCTION OF X AND/OR A"
3380 PRINT"    220 LET  C= DITTO  FOR  X AND/OR A AND/OR B"
3395 PRINT"    230 (SIMILARLY FOR  D)"
3410 PRINT"    240 (SIMILARLY FOR  E)"
3425 PRINT"    250 (SIMILARLY FOR  F)"
```

```
3440 PRINT"   RUN"
3455 IF 00=1 THEN 3860
3470 PRINT
3485 PRINT"WHERE NUM IS THE NUMBER OF FUNCTIONS GIVEN (1-6), XMIN AND"
3500 PRINT"XMAX ARE THE LOWER AND UPPER LIMITS FOR  X, DELX IS THE IN-"
3515 PRINT"CREMENT FOR  X, AND HMIN AND HMAX ARE THE LOWER AND UPPER"
3530 PRINT"LIMITS FOR  THE VALUES OF THE FUNCTION.  NOTE THAT THE HOR-"
3545 PRINT"IZONTAL INCREMENT IS ALWAYS (HMAX-HMIN)/40."
3560STO P
3575 IF K=1THEN 3680
3590 IF K=2THEN 3710
3605 IF K=3THEN 3740
3620 IF K=4THEN 3770
3635 IF K=5THEN 3800
3650 IF K=6THEN 3830
3665 PRINT"MORE THAN SIX FUNCTIONS SPECIF IED.  STO P."
3680 PRINT"A";
3695 RETURN
3710 PRINT"B";
3725 RETURN
3740 PRINT"C";
3755 RETURN
3770 PRINT"D";
3785 RETURN
3800 PRINT"E";
3815 RETURN
3830 PRINT"F";
3845 RETURN
3860 END
```

PLOTPTS:

DESCRIPTION

This program plots data points of a single variable. The plot is a type of bar graph. The plot is done on any standard teletype or terminal. This program will provide a linear or log plot.

USERS

Anyone who gathers data can use this program to make the data visable. This would include experimenters, engineers, astronomers, housewives, etc. A housewife could use this program to plot the grocery usages for her family. From this, usage trends could be visually indicated.

INSTRUCTIONS

Type RUN and the program will prompt for all of the information necessary for execution. For additional information list the program.

LIMITATIONS

There should be no problem in executing this program in most systems, providing there is sufficient memory space available. The program is set up to handle 256 data points. The source code requires 3K Bytes for storage. With the DIM statement in line 1060 set for 256 this program will require 9K Bytes for execution. There is a FILES statement in line 1100. This statement should be removed if your system does not allow file operations.

PLOTPTS

```
1010 REM THIS IS PLOTPTS
1020 REM THIS PROGRAM PLOTS POINTS IN A BAR TYPE GRAPH
1030 REM INPUT MAY BE FROM FILE NAMED SCR7 BY CHANGING
1040 REM THE #0 IN THE INPUT STATEMENT TO #1
1050 REM OTHERWISE THE INPUT IS FROM THE KEYBOARD.
1060 DIM Y(256), S(256)
1070 PRINT"IF YOU WANT A LINEAR PLOT TYPE (1), OR (2) LOG PLOT"
1080 PRINT
1090 INPUT P
1100 FILES SCR7
1110 PRINT
1120 PRINT"INPUT X LIMITS, (MIN.) TO (MAX.) STEP (X1)"
1130 PRINT
1140 INPUT X1,X2,X3
1150 PRINT
1160 PRINT
1170 IF P=2 GO TO 1240
1180 PRINT " INPUT (Y) LIMITS, (MIN.) AND (MAX.)"
1190 PRINT
1200 INPUT Q1,Q2
1210 LET M=50/(Q2-Q1)
1220 LET Q3=(Q2-Q1)/50
1230 IF P=1 GO TO 1350
1240 PRINT " INPUT LOG(Y) LIMITS, (MIN)=10**N AND (MAX)=10**N1"
1250 PRINT " REM YOU CAN'T CROSS ZERO, AND DON'T FORGET THE SIGN"
1260 PRINT
1270 PRINT
1280 INPUT Q1,Q2
1290 LET M=50/(Q2-Q1)
1300 PRINT
1310 PRINT "TYPE IN THE SCALE FACTOR TO INSURE NO ZERO CROSSING"
1320 PRINT "WHERE S=(10**N2). EX. S=1; OR 10; OR 100; ........"
1330 PRINT
1340 INPUT S
1350 PRINT
1360 PRINT
1370 PRINT
1380 PRINT "INPUT YOUR (Y) DATA"
1390 PRINT
1400 LET L7=1+((X2-X1)/X3)
```

```
1410 FOR A=1 TO L7 STEP 1
1420 INPUT #0, Y(A)
1430 NEXT A
1440 PRINT
1450 PRINT
1460 IF P=1 GO TO 1490
1470 LET S$=" ( LOG PLOT ), (SCALE FACTOR)"
1480 IF P=2 GO TO 1500
1490 LET S$=" (LINEAR PLOT), (SCALE FACTOR)"
1500 PRINT
1510 PRINT
1520 PRINT "X";TAB(9);Q1;TAB(28);S$;TAB(62);Q2
1530 PRINT
1540 PRINT "(INCREMENTS)";TAB(28);"DELTA Y =";TAB(32);Q3;TAB(49);S
1550 PRINT
1560 PRINT TAB(17);"I----+----I----+----I----+----I----+----I----+----IO"
1570 FOR A=1 TO L7 STEP 1
1580 IF P=2 GO TO 1650
1590 IF Y(A)>Q1 GO TO 1610
1600 LET Y(A)=Q1
1610 IF Y(A)<Q2 GO TO 1630
1620 LET Y(A)=Q2
1630 LET Z1=M*(Y(A)-Q1)
1640 IF P=1 GO TO 1760
1650 LET C1=0.434294
1660 LET B1=Y(A)/S
1670 IF B1=0 GO TO 1700
1680 LET L1=C1*LOG(B1)
1690 GO TO 1710
1700 LET L1=Q1
1710 IF L1<Q2 GO TO 1730
1720 LET L1=Q2
1730 IF L1>Q1 GO TO 1750
1740 LET L1=Q1
1750 LET Z1=M*(L1-Q1)
1760 LET Z=(1.5)+Z1
1770 LET M3=X3
1780 LET X4=(X1+M3*(A-1))
1790 PRINT X4,TAB(16);
1800 FOR G=1 TO Z
1810 PRINT TAB(17);"X";
1820 NEXT G
1830 PRINT TAB(68);"I"
1840 NEXT A
1850 PRINT TAB(17);"I----+----I----+----I----+----I----+----I----+----IO"
1860 PRINT
1870 PRINT
1880 PRINT "DO YOU WANT ANOTHER PLOT ?"
1890 PRINT
1900 INPUT C$
1910 PRINT
1920 PRINT
1930 PRINT
1940 IF C$="Y" GO TO 1070
1950 IF C$="YES" GO TO 1070
1960 STOP
1970 END
```

POLYNOMIAL FIT:

DESCRIPTION

This program performs a least square fit to equate bivariate data to a polynomial. This is done using an orthogonal polynomial method. The limits are an 11th degree polynomial and 100 data points. The polynomials are fitted one degree at a time in ascending order. At each level the index of determination is listed.

USERS

This is one of the most powerful data fitting programs available. Any individual who analyzes or gathers data will find this program ectremely useful.

INSTRUCTIONS

To use the program input data as follows:

```
     10 DATA  N,D
```

where

N is the number of points to be read
D is the lowest degree to be fit

```
     100 DATA  X(1),Y(1),X(2),Y(2),X(3),.....
```

where

X( ) and Y( ) is the raw data

For additional instructions list the program.

LIMITATIONS

Lines 34 and 39 contain DEF FN__ statements. For more then 100 data points, increase the DIM statements in line 25. At the present this program will require 12K Bytes for storage and execution.

POLYNOMIAL FIT

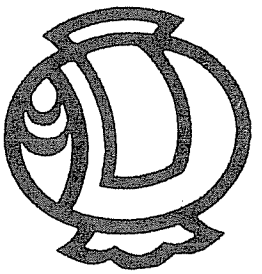
```
10 GO TO 1890
19 READ M,N
20 DIM A(15),B(15),S(15),G(16),U(15)
25 DIM O(100),P(100),X(100),Y(100),C(100)
30 LET Z=0
33 LET O = 1
34 DEF FNL(P) = .43429448 * LOG(P)
35 LET O9 = 1E-19
36 LET O8 = 1E19
37 LET O7 = 1E38
38 LET O6 = 1E-38
39 DEF FNR(P) = INT(P*10000000+.5)/10000000
40 LET K=12
45 LET N=N+1
50 IF N > 12 THEN 1870
55 IF M <  N THEN 2170
60 IF M >100 THEN 1840
70 LET T7=Z
75 LET T8=Z
80 LET W7=Z
100 DATA 1,38.1,1.5,14.67,2,12.76,2.5,13.15,3,11.78
101 DATA 3.5,1.67,4,5.35,4.5,14.6,5,5.3,5.5,1.67,6
102 DATA 8.91,6.5,15.67
300 FOR I=1 TO M
310 READ X(I),Y(I)
320 LET W7=W7+X(I)
330 LET T7=T7+Y(I)
340 LET T8=T8+Y(I)↑2
350 NEXT I
360 LET T9=(M*T8-T7↑2)/(M↑2-M)
370 PRINT
380 PRINT "L E A S T - S Q U A R E S   P O L Y N O M I A L S"
390 PRINT

392 PRINT
400 PRINT "        NUMBER OF POINTS =";M
410 PRINT "         MEAN VALUE OF X =";W7/M
420 PRINT "         MEAN VALUE OF Y =";T7/M
430 PRINT "          STD ERROR OF Y =";SQR(T9)
440 PRINT
450 PRINT "   NOTE:  CODE FOR 'WHAT NEXT?' IS:"
460 PRINT
470 PRINT "            0 = STOP PROGRAM"
480 PRINT "            1 = COEFFICIENTS ONLY"
```

```
490 PRINT "            2 = ENTIRE SUMMARY"
500 PRINT "            3 = FIT NEXT HIGHER DEGREE"
510 PRINT
520 PRINT
530 FOR I=1 TO M
540 LET P(I) = Z
550 LET Q(I) = 0
560 NEXT I
570 FOR I = 1 TO 11
580 LET A(I) = Z
590 LET B(I) = Z
600 LET S(I) = Z
610 NEXT I
620 LET E1=Z
630 LET F1=Z
640 LET W1=M
650 LET N4=K
660 LET I=1
670 LET K1=2
680 IF N=0 THEN 700
690 LET K1=N4
700 LET W=Z
710 FOR L=1 TO M
720 LET W=W+Y(L)*Q(L)
730 NEXT L
740 LET S(I)=W/W1
750 IF I-N4>=0 THEN 1090
760 IF I-M>=0 THEN 1090
770 LET E1=Z
780 FOR L=1 TO M
790 LET A9 = ABS(Q(L))
800 IF A9 < 09 THEN 860
801 LET X9 = ABS(X(L))
802 IF X9 < 06 THEN 844
803 REM: SO  X & Q  NOT TOO CLOSE TO 0.0 FOR LOG
810 LET L2 = FNL(X9) + 2*FNL(A9)
820 IF L2 < 38 THEN 850
821 REM: SO  X & Q  TOO BIG FOR  X*Q↑2
830 LET E1 = 07
831 LET L2 = 38
840 GO TO 870
844 IF A9 < 1  THEN 860
846 IF A9 < 08 THEN 850
847 REM: SO  Q  TOO BIG FOR  Q↑2
849 GO TO 830
850 LET E1 = E1 + X(L) * A9↑2
860 NEXT L
870 IF L2 - FNL(W1) > (-38) THEN 900
880 LET E1 = 0
890 GO TO 910
900 LET E1 = E1/W1
```

```
910 LET A(I+1)=E1
920 LET W=Z
930 FOR L=1 TO M
940 LET V=(X(L)-E1)*Q(L)-F1*P(L)
950 LET P(L)=Q(L)
960 LET Q(L)=V
970 LET V9 = ABS(V)
980 IF V9 < 09 THEN 1030
990 IF V9 < 08 THEN 1020
1000 LET W = 07
1010 GO TO 1040
1020 LET W=W+V*V
1030 NEXT L
1040 LET F1= W/W1
1050 LET B(I+2)=F1
1060 LET W1=W
1070 LET I=I+1
1080 GOTO 700
1090 FOR L = 1 TO 13
1100 LET G(L)=Z
1110 NEXT L
1120 REM:
1130 LET G(2) = 0
1140 FOR J=1 TO N
1150 LET S1 = Z
1160 FOR L = 2 TO N+1
1170 IF L=2 THEN 1190
1180 LET G(L) = G(L) - A(L-1)*G(L-1) - B(L-1)*G(L-2)
1190 LET S1 = S1 + S(L-1)*G(L)
1200 NEXT L
1210 LET U(J)=S1
1220 REM:
1230 LET L = N+1
1240 FOR I2=2 TO N
1250 LET G(L)=G(L-1)
1260 LET L=L-1
1270 NEXT I2
1280 LET G(2) = Z
1290 NEXT J
1300 REM:
1310 PRINT
1320 LET T=Z
1330 FOR L=1 TO M
1340 LET C(L)=Z
1350 LET J=N
1360 FOR I2=1 TO N
1370 LET C(L)=C(L)*X(L)+U(J)
1380 LET J=J-1
1390 NEXT I2
1400 LET T3=Y(L)-C(L)
1410 LET T=T+T3↑2
```

```
1420 NEXT L
1430 IF M<>N THEN 1460
1440 LET T5=0
1450 GOTO 1470
1460 LET T5=T/(M-N)
1470 LET Q7 = 1 - T/(T9*(M-1))
1490 PRINT "POLYFIT OF DEGREE";N-1;
1500 PRINT "INDEX OF DETERM =";FNR(Q7);
1510 GOSUB 2200
1540 IF R=0 THEN 9999
1550 IF R=3 THEN 1810
1555 PRINT"
          "
1560 PRINT TAB(8);  "TERM "," COEFFICIENT"
1570 PRINT
1580 FOR J=1 TO N
1590 LET I2=J-1
1600 PRINT I2,U(J)
1610 NEXT J
1620 IF R=1 THEN 1780
1630 PRINT
1640 PRINT "    X-ACTUAL","    Y-ACTUAL","       Y-CALC","         DIFF";
1650 PRINT "        PCT-DIFF"
1660 PRINT
1670 FOR L=1 TO M
1680 LET Q8=Y(L)-C(L)
1690 PRINT X(L),Y(L),C(L),Q8,
1700 IF C(L)=0 THEN 1730
1710 PRINT 100*Q8/C(L)
1720 GOTO 1740
1730 PRINT "INFINITE"
1740 NEXT L
1750 PRINT
1760 PRINT "           STD ERROR OF ESTIMATE FOR Y =";SQR(T5)
1770 IF K=N THEN 9999
1780 PRINT
1790 GOSUB 2200
1800 GOTO 1540
1810 LET N=N+1
1820 IF M<N THEN 2170
1830 GOTO 1090
1840 PRINT
1850 PRINT "PROGRAM SIZE LIMIT IS 100 DATA POINTS."
1860 GOTO 9999
1870 PRINT "ELEVENTH DEGREE IS THE LIMIT."
1880 GOTO 9999
1890 PRINT "MINIMUM INSTRUCTION = 0,  ALL = OTHER #, WHICH...";
1900 INPUT Q0
1910 IF Q0=0 THEN 2090
1920 PRINT
```

```
1940 PRINT
1950 PRINT "FITS LEAST-SQUARES POLYNOMIALS TO BIVARIATE DATA BY";
1960 PRINT " ORTHOGONAL POLYNMLS."
1970 PRINT
1980 PRINT "   LIMITS :   11-TH DEGREE FIT   &   MAX OF 100 DATA POINTS."
1985 PRINT"  BUT FITS HIGHER THAN DEGREE 5 MAY GIVE POOR RESULTS."
1990 PRINT "   POLFIT ALLOWS USER TO SPECIFY THE LOWEST DEGREE POLYNOM";
2000 PRINT "IAL TO BE FIT"
2010 PRINT"   AND THEN FITS POLYNOMIALS IN ORDER OF ASCENDING DEGREE."
2020 PRINT"   AT EACH STAGE, THE INDEX OF DETERMINATION IS PRINTED, AND"
2030 PRINT"   THEN THE USER HAS THE CHOICE OF:"
2040 PRINT"        GOING TO NEXT HIGHER FIT, OR"
2050 PRINT"        GETTING ONE OF TWO SUMMARIES, OR"
2060 PRINT"        STOPPING THE PROGRAM."
2070 PRINT
2080 PRINT "TO USE, TYPE:"
2090 PRINT
2100 PRINT "   10   DATA N,D"
2110 PRINT "          (WHERE N = NUMBER OF DATA POINTS TO BE READ"
2120 PRINT "             AND D = INITIAL [LOWEST] DEGREE TO BE FIT)"
2130 PRINT "   100 DATA X(1),Y(1),X(2),Y(2),....,X(N),Y(N)"
2140 PRINT "         (CONTINUATION ON LINES 101-299 AS NEEDED)"
2150 PRINT "   RUN"
2160 GO TO 9999
2170 PRINT
2180 PRINT "TOO FEW POINTS FOR FITTING DEGREE";N-1
2190 GOTO 9999
2200 PRINT "   WHAT NEXT";
2210 INPUT R
2220 RETURN
9999 END
```



REGRESSION:

DESCRIPTION

This program performs a regression analysis on data for up to six variables. This program may be used to fit a random sequence to an observed variable. This is one of the most versatile regression programs available.

USERS

Experimenters will find this program an invaluable aid in matching data to a particular system. One interesting example would be to put a microphone out in a rain storm and digitize the noise the rain makes. This will be the input data for the Y variable. The generating equation is:

Y = A + B X1 + C X2 + D X3

Then three (3) random number functions are used for inputs to the next three variables; X1, X2, and X3. When the program is run the three coefficients: B, C, and D can represent the gain of three noise amplifiers. When the outputs of these three amps are mixed the sound will resemble the initial rain storm noise. Five generators could have been used, however three is sufficient for this type of analog generation.

INSTRUCTIONS

If transforms are to be used see the following instruction sheet. Data however is entered starting in line 100, for the first variable. Then variable two starting in line 200, variable three in line 300 and so forth. The equation that is fitted is:

Y = A + B X1 + C X2 + D X3 + E X4 + F X5

where

Y - is variable one
X1 - is variable two
X2 - is variable three
and so forth.....

After your data has been entered type RUN.

LIMITATIONS

Lines 702 and 882 contain a Restore statement. Matrix statements are in line 711 and 712. Lines 718 and 721 contain MAT Read statements. Lines 732 to 738 contain MAT = ZER statements. Line 761 contains a MAT = INV( ) statement. The source code for this program will require 3K Bytes of memory space for storage. Regression will require 13K Bytes of available memory for execution. If more then 61 data points per variable are required the DIM statement in line 711 will need to be increased.

ADDITIONAL PROGRAM INFORMATION:

LINE NUMBERS 41-699 ARE FREE FOR USE AS DATA STATEMENTS, AS REQUIRED.
ALL VALUES FOR ONE VARIABLE ARE GIVEN IN DATA STATEMENTS, FOLLOWED BY
ALL DATA FOR A SECOND VARIABLE, AND SO ON.
ONE WAY TO ARRANGE INPUT IS TO PUT ALL VALUES OF VARIABLE 1 AT LINES
100-199, ALL FOR VARIABLE 2 AT LINES 200-299, ETC.
IF SEPARATE TAPES ARE MADE FOR EACH VARIABLE, THE FULL USE OF INPUT
FLEXIBILITY CAN BE OBTAINED.

ACCEPTS UP TO 61 OBSERVATIONS ON UP TO 6 VARIABLES. THE ORDER
OF VARIABLES IN THE INPUT IS IMPORTANT ONLY IF TRANSFORMATIONS
ARE NOT BEING USED. IF NO TRANSFORMED VARIABLES ARE GIVEN, THEN
THE PROGRAM FITS A LINEAR FUNCTION FOR THE FIRST DATA VARIABLE
IN TERMS OF THE FOLLOWING VARIABLES. BY USING TRANSFORMS, ANY
FUNCTION OF ANY (OR ALL) OF THE INPUT VARIABLES MAY BE USED AS
ANY VARIABLE IN THE ANALYSIS.
TRANSFORMS ARE ENTERED BEFORE THE RUN, AT LINES 1000-1099 AS:

10XX LET X(J) = ANY FUNCTION OF V(1), V(2),...,V(R)

WHERE X(J) IS THE J-TH VARIABLE IN THE ANALYSIS (J=1 IS THE
DEPENDENT VARIABLE), AND THE V(I) ARE THE INPUT DATA VARIABLES.
THUS, ONE MIGHT ENTER DATA FOR THREE VARIABLES V(1), V(2),
AND V(3), BUT RUN THE ANALYSIS WITH X(1)=V(2) AS THE DEPENDENT
VARIABLE, AND THREE 'INDEPENDENT' VARIABLES, BY ENTERING
TRANSFORMATIONS LIKE THE FOLLOWING AT LINES 1000-1003:

```
1000 LET X(1)=V(2)
1001 LET X(2)=LOG(V(1)+V(3))
1002 LET X(3)=V(1)+X(2)*EXP(-1-V(3))
1003 LET X(4)=(V(3)-1)*2
```

NOTE: IF TRANSFORMS ARE USED, THE PROGRAM EXPECTS AT LEAST
THREE SETS OF INPUT VARIABLES. IF THERE ARE ONLY TWO SETS OF
VALID DATA, A SET OF ZEROS MAY BE INPUT FOR THE THIRD VARIABLE.
HOWEVER, THIS DUMMY SET SHOULD NOT BE USED IN DEFINING THE
TRANSFORMATIONS.

```
25 PRINT
26 PRINT
27 PRINT
28 PRINT
700 READ E1
701 IF E1 = 1E38 THEN 1101
702 RESTORE
703 PRINT "
        '     DO YOU WANT INSTRUCTIONS (0=NO, 1=YES).... WHICH ";
704 INPUT O0
705 IF O0 = 0 THEN 709
707 PRINT"N=# VALUES, R=# IN VARIABLES, S=# OUT VARIABLES, D IS A CODE"
708 PRINT"VALUE (D: 1 = TRANSFORMS HAVE BEEN SPECIFIED, 0 = OTHERWISE.)"
709 PRINT " N, R, S, D = ";
710 INPUT P1,R1,N1,D
711DIMY(6,61),Z(6,61),S(6,6),T(6,6),W(5,5)
712DIMB(5,1),F(6,1),G(6),X(6),V(6)
713LETN=N1-1
714LETR=R1-1
715LETP=P1-1
716IFD=1THEN 721
717LETN=R
718MATREADZ(N+1,P+1)
719GOSUB 818
720GOTO 732
721MATREADY(R+1,P+1)
722GOSUB 818
723FORK=1TOP+1
724FORL=1TOR1
725LETV(L)=Y(L,K)
726NEXTL
727GOSUB 1000
728FORL=1TON1
729LETZ(L,K)=X(L)
730NEXTL
731NEXTK
732MATY=ZER(3,P+1)
733LETP2=1/P1
734MATF=ZER(N+1,1)
735MATW=ZER(N,N)
736MATS=ZER(N+1,N+1)
737MATT=ZER(N+1,N+1)
738MATB=ZER(N,1)
```

```
739FORK=1TOP+1
740FORI=1TON+1
741LETF(I,1)=F(I,1)+P2*Z(I,K)
742FORJ=1TON+1
743LETS(I,J)=S(I,J)+Z(I,K)*Z(J,K)
744NEXTJ
745NEXTI
746NEXTK
747FORI=1TON+1
748FORJ=1TON+1
749LETT(I,J)=S(I,J)-P1*F(I,1)*F(J,1)
750NEXTJ
751NEXTI
752FORJ=1TON+1
753LETG(J)=SQR(P2*T(J,J))
754NEXTJ
755MATS=ZER(N,N)
756FORI=1TON
757FORJ=1TON
758LETS(I,J)=T(I+1,J+1)
759NEXTJ
760NEXTI
761MATW=INV(S)
762FORI=1TON
763FORJ=1TON
764LETB(I,1)=B(I,1)+T(1,J+1)*W(I,J)
765NEXTJ
766NEXTI
767LETA=F(1,1)
768LETS7=0
769LETS8=0
770FORI=1TON
771LETA=A-B(I,1)*F(I+1,1)
772NEXTI
773FORK=1TOP+1
774LETY(1,K)=A
775FORJ=1TON
776LETY(1,K)=Y(1,K)+B(J,1)*Z(J+1,K)
777NEXTJ
778LETY(2,K)=Y(1,K)-Z(1,K)
779LETY(3,K)=100*Y(2,K)/Y(1,K)
780LETS7=S7+Y(2,K)
781LETS8=S8+Y(2,K)↑2
782NEXTK
783LETW8=P2*SQR(P1*S8-S7↑2)
784LETW7=P2*S7
785 PRINT
786 PRINT"      M U L T I V A R I A T E   C U R V E   F I T"
787 PRINT
788 PRINT
789 PRINT"VARIABLE","REGR COEFF","MEAN VALUE","STD DEV"
```

```
790 PRINT
791 PRINT" 1 (CONSTANT =";A;")";F(1,1),G(1)
792 PRINT
793FORI=1TON
794PRINTI+1,B(I,1),F(I+1,1),G(I+1)
795NEXTI
796PRINT
797PRINT
798PRINT"   STANDARD DEVIATION OF RESIDUALS = ";W8
799PRINT"     INDEX OF DETERMINATION (R-SQ) = ";1-(W8/G(1))↑2
800PRINT"       ZERO-CHECK ON MEAN RESIDUAL = ";W7
801PRINT
802PRINT
803PRINT"      A C T U A L   V S   C A L C U L A T E D"
804PRINT"                    "
805PRINT"ACTUAL","CALCULATED","DIFFERENCE"," PCT DIFFER"
806PRINT
807FORK=1TOP+1
808PRINTZ(1,K),Y(1,K),Y(2,K),
809LETQ=.1*SGN(Y(3,K))*INT(ABS(10*Y(3,K)))
810IFQ=0THEN 815
811IFQ<0THEN 813
812PRINT"      ";
813PRINTQ
814GOTO 816
815PRINT"    .0"
816NEXTK
817STOP
818READE1
819IFE1=1E38THEN 824
820PRINT
821PRINT"WRONG AMOUNT OF DATA SPECIFIED?  PLEASE CHECK."
822RESTORE
823GOTO 703
824RETURN
825STOP
1000 PRINT"NO TRANSFORMS"
1100RETURN
1101 PRINT

1103 PRINT"----"
1104 DATA 1E38, 0,0,0,0,0,0,0
1105 END
```

STAT 1:

DESCRIPTION

This statistical program calculates the mean, variance, and standard deviation for several sets of data distributions.

USERS

This program will be of use to individuals who work with data bases. Quality control engineers, reliability engineers, data gatherers, astronomers, experimenters, statisticians are a few of the types of individuals who could find use for this program.

INSTRUCTIONS

Enter all of the data starting in line 1, with data statements before running this program. For information about data format and entry, list lines 900 to 1390. For detailed program information list Stat 1.

LIMITATIONS

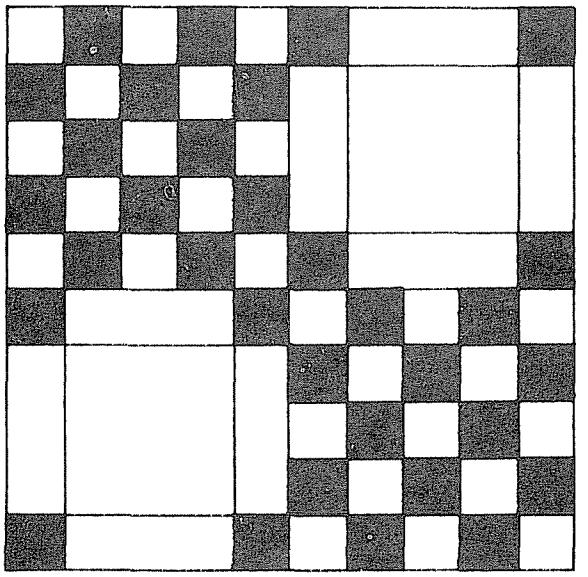
Line 345 contains a Restore statement. This program will store and execute in 5K Bytes of memory in most computer systems.

STAT 1

```
340 READ 00
345 RESTORE
350 IF 00=9E20 THEN 940
355 LET H=0
357 PRINT "

           ARITHMETIC MEAN, VARIANCE, AND STANDARD DEVIATION
                                                              "
360 READ C
370 LET H=H+1
380 IF C=9E20 THEN 1400
390 LET S1=0
400 LET S2 = 0
410 IF C<>0 THEN 590
430 PRINT "
           FOR GROUPED DATA SET";H,"
                                    "
450 READ Q
460 LET N = 0
470 PRINT "       X-VALUE","       FREQUENCY
                                              "
490 FOR P = 1 TO Q
500 READ X, M
510 LET N = N + M
520 LET S1 = S1 + M*X
530 LET S2 = S2 + M*X*X
540 PRINT X, M
550 NEXT P
560 GOTO 710
590 PRINT "

           INDIVIDUAL SET NUMBER";H
610 PRINT "
           SAMPLE VALUES:
                         "
630 LET N = C
640 FOR P = 1 TO N
650 READ X
660 LET S1 = S1 + X
670 LET S2 = S2 + X*X
680 PRINT X;
690 NEXT P
700 PRINT
710 LET A = S1/N
720 LET V=(S2-A*S1)/N
```

```
730 LET V1=V*N/(N-1)
740 PRINT
750 PRINT "MAXIMUM LIKELIHOOD";
760 GOSUB900
770 PRINT "       NUMBER OF VALUES = ";N
780 PRINT "         ARITHMETIC MEAN = ";A
790 PRINT "     STANDARD DEVIATION = ";SQR(V)
800 PRINT "         SAMPLE VARIANCE = ";V
810 PRINT
820 PRINT "UNBIASED";
830 GOSUB900
840 PRINT "         ARITHMETIC MEAN = ";A
850 PRINT "     STANDARD DEVIATION = ";SQR(V1)
860 PRINT "                VARIANCE = ";V1
870 PRINT
880 PRINT
890 GOTO 360
900 PRINT " ESTIMATES"
910 PRINT "OF POPULATION PARAMETERS"
920 PRINT
930 RETURN
940 PRINT
950 PRINT "THIS PROGRAM CALCULATES THE MEAN, VARIANCE, AND STANDARD"
960 PRINT "DEVIATION FOR EACH OF SEVERAL SETS OF INDIVIDUAL VALUES"
970 PRINT "OR FREQUENCY DISTRIBUTIONS."
980 PRINT
990 PRINT "DATA FOR EACH SET OF INDIVIDUAL VALUES IS ENTERED INTO"
1000 PRINT "THE PROGRAM AS FOLLOWS:"
1010 PRINT
1020 PRINT "    1 DATA  N, X(1), X(2), X(3),....., X(N)"
1030 PRINT
1040 PRINT "WHERE THE N VALUES OF THE SET ARE X(1) THRU X(N).  IF"
1050 PRINT "NEEDED, ADDITIONAL DATA STATEMENTS MAY BE USED TO GIVE"
1060 PRINT "THE ENTIRE LIST OF VALUES.   ADDITIONAL CASES MAY BE"
1070 PRINT "GIVEN IN SUBSEQUENT DATA STATEMENTS IN THE SAME FORMAT."
1080 PRINT
1090 PRINT "THE INPUT FOR GROUPED VALUES HAS THE FOLLOWING FORMAT:"
1100 PRINT
1110 PRINT "    1 DATA 0, N, X(1), F(1), X(2), F(2),..., X(N), F(N)"
1120 PRINT
1130 PRINT "WHERE THE INITIAL ZERO SIGNALS GROUPED DATA, THE N IS"
1140 PRINT "THE NUMBER OF DIFFERENT VALUES TO BE GIVEN, AND THE F(I)"
1150 PRINT "ARE THE NUMBER OF TIMES THE X(I) OCCUR.  DATA STATEMENTS"
1160 PRINT "FOLLOWING MAY BE USED TO EXTEND THE LIST AS NECESSARY,"
1170 PRINT "AND BLOCKS OF GROUPED DATA MAY BE INTERMIXED FREELY"
1180 PRINT "WITH STRAIGHT LISTS DESCRIBED ABOVE."
1190 PRINT
1200 PRINT "AS AN EXAMPLE, SUPPOSE WE WERE INTERESTED IN THE MEAN"
1210 PRINT "AND STANDARD DEVIATION OF THE NUMBERS 1,5,4,2,6,7,4,7"
1220 PRINT "AND ALSO FOR THE DISTRIBUTION CONSISTING OF 5-1'S,"
1230 PRINT "3-4'S, 6-7'S, AND 2-11'S.  THESE TWO CASES COULD BE RUN"
```

```
1240 PRINT "BY TYPING THE FOLLOWING:"
1250 PRINT
1260 PRINT "     1 DATA 8,1,5,4,2"
1270 PRINT "     2 DATA 6,7,4,7"
1280 PRINT "     3 DATA 0,4,1,5,4,3"
1290 PRINT "     4 DATA 7,6,11,2"
1300 PRINT "    RUN"
1310 PRINT
1320 PRINT "OR EQUIVALENTLY:"
1330 PRINT
1340 PRINT "     1 DATA 8,1,5,4,2,6,7,4,7,0,4,1,5,4,3,7,6,11,2"
1350 PRINT "    RUN"
1360 PRINT
1370 PRINT "NOTE THAT STATEMENT NUMBERS 1 THRU 299 ARE AVAILABLE"
1375 PRINT "                                        ----------"
1380 PRINT "FOR CONTINUATION OF THE DATA FIELD."
1390 DATA 9E20
1400 END
```



```
STAT 1


1 DATA 22,261.4,270.8,265.4,261.4,258.1,252.1,268.3,250.3
2 DATA 272.3,262.8,255.5,249.6,280.9,270.3,263.2,258.3
3 DATA 256.3,259.3,270.1,259.3,253.2,266.4



RUN



ARITHMETIC MEAN, VARIANCE, AND STANDARD DEVIATION


INDIVIDUAL SET NUMBER      1


MAXIMUM LIKELIHOOD ESTIMATES
OF POPULATION PARAMETERS

        NUMBER OF VALUES        =    22
        ARITHMETIC MEAN         =    262.0591
        STANDARD DEVIATION      =    7.783971
        SAMPLE VARIANCE         =    60.5902

UNBIASED ESTIMATES
OF POPULATION PARAMETERS

        ARITHMETIC MEAN         =    262.0591
        STANDARD DEVIATION      =    7.967148
        VARIANCE                =    63.47545
```

STAT 2:

DESCRIPTION

This program does a statistical analysis on one variable. Once the data for the variable is entered the program calculates 34 different statistical quantities for the data. The data may or may not be weighted. In addition the program calculates a ten class frequency distribution summary.

USERS

Anyone who gathers data for any purpose such as pollsters, surveyers, engineers, experimenters, even fishermen may find uses for this program. For example a fisherman may enter the time of the moon or tide or location versus the number of fish caught. The program will then determine the most likely conditions to catch the most fish.

INSTRUCTIONS

Data must be entered in one of two formats. The two formats are for either weighted or unweighted data. They are as follows:

(1.) FOR WEIGHTED DATA

```
10 DATA  1,X(1),F(1),X(2),F(2),X(3),.....
```

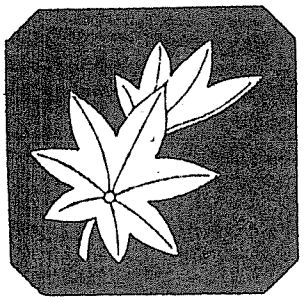
(2.) FOR UNWEIGHTED DATA

```
10 DATA  0,X(1),X(2),X(3),.....................
```

Run the program for additional instructions or list lines 1055 to 1255.

LIMITATIONS

The length of this program is 6K Bytes. The amount of space required for proper execution is a function of the number of data sets in the DIM statement in line 7. With the DIM set for 100 data points, 14K Bytes will be required for execution.



STAT 2

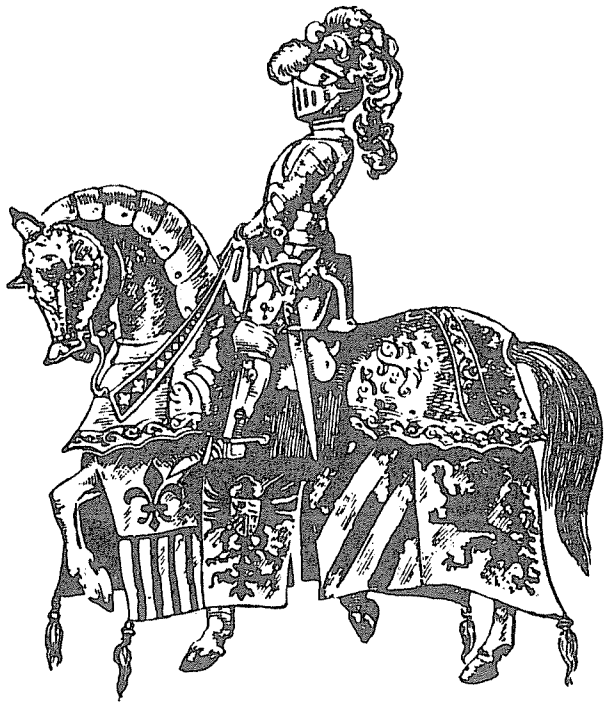
```
7 DIM X(100),W(100),D(110),Y(100)
10 GO TO 1055
135 DATA 1E35,3E33
136 PRINT
137 PRINT
140 PRINT "COMPUTATIONS ON THE DATA ARRAY:"
145 PRINT
150 READ N
155 LET J=0
160 LET I=1
165 LET J1=0
170 IF K=10 THEN 835
175 LET I1=0
180 LET K1=0
185 LET L1=0
190 READ X(I)
195 LET Y(I)=X(I)
200 IF X(I)=1E35 THEN 260
205 LET W(I)=1
210 IF N = 0 THEN 225
215 READ W(I)
220 IF W(I)=0 THEN 230
225 LET J=J+1
230 LET I=I+1
235 LET I1 = X(I-1) + I1
240 LET J1 = W(I-1) + J1
245 LET K1 = W(I-1)*X(I-1) +K1
250 LET L1=W(I-1)*X(I-1)↑2+L1
255 GOTO 190
260 LET I=I-1
265 PRINT "NUMBER OF VALUES =";I
270 PRINT "NUMBER OF NONZERO WEIGHTS =";J
275 PRINT "SUM OF WEIGHTS =";J1
280PRINT "SUM OF VALUES =";I1
285 LET X2 = K1/J1
290 PRINT "WEIGHTED MEAN =";X2
295 PRINT "UNWEIGHTED MEAN =";I1/I
300 LET M1=0
305 LET M2=0
310 LET D(1)=X(1)-X2
315 LET A=1
320 GOSUB 1020
325 LET S=D(1)↑2*W(1)
330 LET S1=D(1)
```

```
335 LET S2=D(1)↑G*W(1)
340LET S3=D(1)↑3
345LET S4=D(1)↑4
350 LET S5=0
355 LET S6=ABS(D(1))
360 LET X3=X(1)
365 LET X4=X(1)
370 FOR A = 2 TO I
375 IF X(A) >= X3 THEN 390
380 LET X3 = X(A)
385 GO TO 400
390 IF X(A)<= X4 THEN 400
395 LET X4=X(A)
400 LET D(A)=X(A) - X2
405 GOSUB 1020
410 LET S = D(A)↑2*W(A) + S
415 LET S5 = (X(A)-X(A-1))↑2 + S5
420 LET S1=A*D(A)+S1
425 LET S2=D(A)↑2*W(A)+S2
430LET S3=D(A)↑3+S3
435LET S4=D(A)↑4+S4
440 LET S6=ABS(D(A))+S6
445 NEXT A
450PRINT "MINIMUM VALUE=";X3
455PRINT "MAXIMUM VALUE=";X4
460 PRINT "RANGE =";X4-X3
465 PRINT "WEIGHTED SUM OF SQUARES =";L1
470 LET S=SQR(S/(J1-1))
475 PRINT "VARIANCE =";S↑2
480 PRINT "STANDARD DEVIATION =";S
485 PRINT "STANDARD DEVIATION OF MEAN =";S/SQR(J1)
490PRINT "COEFFICIENT OF VARIATION =";100*S/X2
495 PRINT "STUDENT'S T =";X2*SQR(J1)/S
500 LET S5 = S5/(I-1)
505 PRINT "MEAN SQUARE SUCCESSIVE DIFFERENCES =";S5
510 PRINT "(MEAN SQ SUCC DIFF)/(VARIANCE) =";S5/S↑2
515 LET N=1
520 LET N1=1
525 LET K=1
530 IF X(K)=X(K+1) THEN 560
535 IF X(K)>X(K+1) THEN 550
540 IF X(K+1)>X(K+2) THEN 555
545 GOTO 560
550 IF X(K+1)>=X(K+2) THEN 560
555 LET N=N+1
560IF D(K)=0THEN 610
565 IF D(K)>0 THEN 590
570 IF D(K+1)<0 THEN 610
575 IF D(K+1)>0 THEN 605
580 IF D(K+2)<=0 THEN 610
585 GOTO 605
```

```
590 IF D(K+1)<0 THEN 605
595 IF D(K+1)>0 THEN 610
600 IF D(K+2)<=0 THEN 610
605 LET N1=N1+1
610 LET K=K+1
615 IF K<= I-2 THEN 590
620 IF X(I-1)=0 THEN 650
625 IF X(I-1)>0 THEN 640
630 IF X(I)<=0 THEN 650
635 GOTO 645
640 IF X(I)>0 THEN 650
645 LET N1=N1+1
650 LET N2=(2*I-1)/3
655 FOR J=1 TO I-1
660 LET K=J+1
665 FOR L=K TO I
670 IF X(J)<=X(L) THEN 690
675 LET N3=X(J)
680 LET X(J)=X(L)
685 LET X(L)=N3
690 NEXT L
695 NEXT J
700 IF I/2=INT(I/2) THEN 720
705 LET J=INT((I+1)/2)
710 LET J=X(J)
715 GOTO 730
720 LET J=INT(I/2)
725 LET J=(X(J)+X(J+1))/2
730 PRINT "MEDIAN =";J
735 LET N3=SQR((16*I-29)/90)
740 PRINT "NUMBER OF RUNS UP AND DOWN =";N
745 PRINT "EXPECTED NUMBER OF RUNS =";N2
750 PRINT "STD DEV OF NUMBER OF RUNS =";N3
755 LET K=1
760 FOR J=1 TO 10
765 LET D(100+J)=0
770 NEXT J
775 PRINT "(ACTUAL RUNS - EXP RUNS)/(STD DEV) =";ABS(N-N2)/N3
780 PRINT
785 PRINT "FREQUENCY DISTRIBUTION (TEN EQUAL CLASSES):"
790 LET M3=(X4-X3)/10
795 LET M4=X3+M3
800 FOR J=1 TO I
805 IF X(J)<=M4 THEN 835
810 IF K>10 THEN 840
815 IF K=10 THEN 835
820 LET K=K+1
825 LET M4=X3+K*M3
830 IF X(J)>M4 THEN 810
835 LET D(100+K)=D(100+K)+1
840 NEXT J
```

```
845 FOR J=1 TO 10
850 PRINT D(100+J);
855 NEXT J
860 PRINT
865 PRINT
870 PRINT "COMPUTATIONS ON DEVIATIONS FROM MEAN:"
875 PRINT
880 PRINT "NUMBER OF + SIGNS IN DEVIATIONS =";M1
885 PRINT "NUMBER OF - SIGNS IN DEVIATIONS =";M2
890 PRINT "NUMBER OF RUNS (SIGN CHANGES + 1) =";N1
895 LET M3 = 1+(2*M1*M2/I)
900 LET M4 = SQR(2*M1*M2*(2*M1*M2-M1-M2))/SQR((M1+M2)↑2*(I-1))
905 PRINT "EXPECTED NUMBER OF RUNS =";M3
910 PRINT "STD DEVIATION OF NUMBER OF RUNS =";M4
915 PRINT "(ACTUAL RUNS - EXP RUNS)/(STD DEV) =";ABS(N1-M3)/M4
920 LET T = 12*S1/(I↑3-I)
925 PRINT "TREND VALUE =";T
930 LET M3 = SQR(12*S2/(I↑3-I)-T↑2)/(I-2)
935 PRINT "STD DEV OF TREND =";M3
940 PRINT "(TREND)/(STD DEV) =";T/M3
945LET M3=(S3/I)↑2/(S*S-S*S/I)↑3
950 LET M4=(S4/J1)/(S*S-S*S/J1)↑2
955 PRINT "BETA ONE =";M3
960 PRINT "BETA TWO =";M4
965 PRINT "MEAN DEVIATION =";ABS(S6)/I
970 PRINT
975 PRINT "RECAPITULATION OF INPUT:"
980 PRINT
985 PRINT TAB(7);"VALUE";TAB(17);"DEVIATIONS";TAB(35);"WEIGHTS";
990 PRINT TAB(44);"ORDERED ARRAY"
995 PRINT
1000 FOR I1=1 TO I
1005 PRINT Y(I1),D(I1),W(I1),X(I1)
1010 NEXT I1
1015 STOP
1020IF D(A)>0 THEN 1040
1025IF D(A)=0 THEN 1045
1030 LET M2=M2+1
1035 GOTO 1045
1040 LET M1=M1+1
1045 RETURN
1050 STOP
1055 PRINT"THESE ARE INSTRUCTIONS FOR USING THE PROGRAM       "
1065 PRINT
1075 PRINT" THE         PROGRAM PERFORMS A STATISTICAL ANALYSIS ON"
1085 PRINT" DATA FOR ONE VARIABLE.   IT COMPUTES 34 DIFFERENT MEASURES"
1095 PRINT" FOR AN ARRAY OF WEIGHTED (AS WITH FREQUENCIES) OR UNWEIGHTED"
1105 PRINT" VALUES OF THE VARIABLE.  IT ALSO GIVES A 10-CLASS FREQUENCY"
1115 PRINT" DISTRIBUTION SUMMARY, AND A RECAPITULATION OF THE INPUT"
1125 PRINT" DATA IN TERMS OF DEVIATIONS FROM THE MEAN AND AS AN ORDERED"
1135 PRINT" ARRAY.  EITHER OF TWO TYPES OF INPUT ARE SUPPLIED, DEPENDING"
```

```
1145 PRINT" ON WHETHER OR NOT WEIGHTS ARE TO BE GIVEN FOR THE VALUES:"
1155 PRINT"     1)  FOR UNWEIGHTED VALUES:"
1165 PRINT"         10 DATA 0, X(1), X(2), X(3),....."
1175 PRINT
1185 PRINT"     2)  FOR DATA WITH WEIGHTS OR FREQUENCIES:"
1195 PRINT"         10 DATA 1, X(1), F(1), X(2), F(2), X(3), F(3),..."
1205 PRINT
1215 PRINT" WHERE THE INITIAL ZERO OR ONE SIGNALS THE PRESENCE OR THE"
1225 PRINT" ABSENCE OF WEIGHTS.  LINES 11 THRU 99 ARE AVAILABLE FOR"
1235 PRINT"ADDITIONAL INPUT DATA."
1245 PRINT" NOTE:  THIS PROGRAM PRODUCES OUTPUT CORRESPONDING TO THE"
1255 PRINT" LISTING IN THE NATIONAL BUREAU OF STANDARDS HANDBOOK NO. 101."

99999 END
```



T-DISTRIBUTION:

DESCRIPTION

T-Distribution will perform several types of calculations for normal and T-distributions. If it is given 1 or 2 tailed probabilities it will calculate the corresponding limits. If it is given a variate or set of limits it will calculate the corresponding probabilities.

USERS

Individuals who would like to analyze most types of data will find T-Distribution very useful. In the included example this program is used to calculate the probability that monthly sales for the next period will be within certain limits. The input for the prediction is past sales histories. From this anyone in business or who does ordering for a business could benefit by use of this program.

INSTRUCTIONS

Data must be entered into line 100 prior to running the program. The format for this data and entry information for normal or T-Distributions is given in lines 1360 to 1600 of the program. Standard - t curves are based on a mean of Ø and a sigma of 1.

LIMITATIONS

Starting in line 70 and continuing to line 95 the DEF FN__ statement is used. These functions are used extensively throughout the program. The program will require 5K Bytes for storage and 6K Bytes of memory for operation.

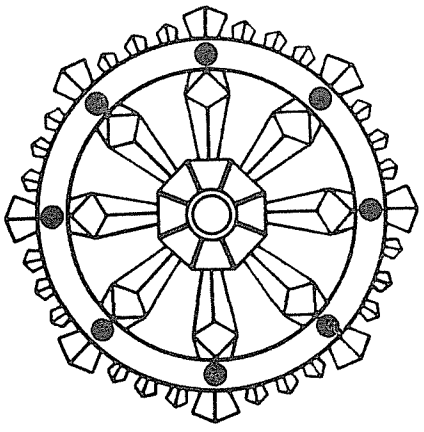
T-DISTRIBUTION

```
4REM
29 REM: LINES 30-60 VITAL; KEEP 'EM IN; THEY'RE FOR INTERPOLATION.
30 DATA 5000000,5398278,5792597,6179114,6554217,6914625,7257469
35 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
40 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
45 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
50 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
55 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
60 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
65 DIM X(49)
70 DEF FNW(U)=(U-M)/S
75 DEF FNQ(U)=M+U*S
80 DEF FND(U)=X(U)-X(U-1)
85 DEF FNB(U)=U-U*(U-1)*(D2/(2*D1)+(U-2)*D3/(6*D1))
90 DEF FNT(U)=1-((U↑2)+1)/(4*D)+(13*(U↑2)↑2+8*(U↑2)+3)/(96*D↑2)
95 DEF FNZ(U)=1+((U↑2)+1)/(4*D)+(((U↑2)+3)*(5*(U↑2)+1))/(96*D↑2)
100 GO TO 1360
200 DATA 99
210 FOR I = 1 TO 49
220 READ X(I)
230 NEXT I
240 READ D,M,S
250 PRINT
260 PRINT "   CALCULATIONS FOR A ";
270 IF D=0 THEN 300
280 PRINT "STUDENT'S T-";
290 GOTO 310
300 PRINT "NORMAL ";
310 PRINT "DISTRIBUTION"
320 PRINT "   HAVING A MEAN OF ";M;"AND A SIGMA OF ";S
330 IF D=0 THEN 350
340 PRINT "   AND HAVING ";D;"DEGREES OF FREEDOM."
350 PRINT
360 PRINT "CASE      THE PROBABILITY OF A VARIATE:"
370 PRINT
380 LET C=0
390 READ H
400 IF H=99 THEN  9999
410 LET C=C+1
420 IF H=4 THEN 800
430 IF H=3 THEN 720
440 IF H=2 THEN 550
450 IF H=1 THEN 480
```

```
460 PRINT C;"   ERROR...ILLEGAL BLOCK TYPE SPECIFIED"
470 GO TO 9999
480 READ V
490 LET B1=FNW(V)
500 IF D=0 THEN 520
510 LET B1=B1*FNT(B1)
520 GOSUB 890
530 PRINT C;"   EXCEEDING ";V;"IS";1-B2
540 GOTO 390
550 READ V1,V2
560 IF V1<V2 THEN 600
570 LET W=V1
580 LET V1=V2
590 LET V2=W
600 LET W1=FNW(V1)
610 LET W2=FNW(V2)
620 IF D=0 THEN 650
630 LET W1=W1*FNT(W1)
640 LET W2=W2*FNT(W2)
650 LET B1=W1
660 GOSUB 890
670 LET Q1=B2
680 LET B1=W2
690 GOSUB 890
700 PRINT C;"   BETWEEN ";V1;"AND ";V2;"IS";B2-Q1
710 GOTO 390
720 READ P
730 GOSUB 1330
740 LET A1=1-P
750 GOSUB 1030
760 IF D=0 THEN 780
770 LET A2=A2*FNZ(A2)
780 PRINT C;"   EXCEEDING ";FN0(A2);"IS";P
790 GOTO 390
800 READ P
810 GOSUB 1330
820 LET A1=0.5*(1+P)
830 GOSUB 1030
840 IF D=0 THEN 860
850 LET A2=A2*FNZ(A2)
860 PRINT C;"   BETWEEN ";FN0(-A2);"AND ";FN0(A2);"IS";P
870 GOTO 390
880 REM  FIND PROB FOR A GIVEN NORMAL DEVIATE
890 IF B1<-4.5 THEN 1000
900 IF B1<0 THEN 970
910 IF B1<4.5 THEN 940
920 LET B2=1
930 GOTO 1010
940 GOSUB 1120
950 LET B2=0
960 GOTO 1010
```

```
970 GOSUB 1120
980 LET B2=1-Q
990 GOTO 1010
1000 LET B2=0
1010 RETURN
1020 REM  FIND NORMAL DEVIATE FOR A GIVEN PROB
1030 IF A1>0.5 THEN 1080
1040 LET A1=1-A1
1050 GOSUB 1190
1060 LET A2=-Q
1070 GOTO 1100
1080 GOSUB 1190
1090 LET A2=Q
1100 RETURN
1110 REM  INTERPOLATE FOR NORMAL PROBABILITY
1120 LET Z=10*ABS(B1)
1130 LET K=INT(Z)
1140 LET D1=Z-K
1150 LET Q=X(K+1)+D1*FND(K+2)+(D1*(D1-1)/2)*(FND(K+3)-FND(K+1))
1160 LET Q=1E-5*INT(5E-1+1E-2*Q)
1170 RETURN
1180 REM  REVERSE INTERPOLATE FOR NORMAL DEVIATE
1190 LET Z=1E7*A1
1200 FOR I = 1 TO 46
1210 IF Z<X(I) THEN 1230
1220 NEXT I
1230 LET D1 = FND(I+1)
1240 LET E1 = FND(I+2)
1250 LET D2=E1-D1
1260 LET D3 = FND(I+3) - E1 - D2
1270 LET U = (Z-X(I)) / D1
1280 LET Q=FNB(FNB(U))
1290 LET Q=0.1*(Q+I-1)
1300 RETURN
1310 PRINT C;"   MUST BE SPECIFIED IN RANGE 0.000005 TO 0.999995"
1320 GOTO 390
1330 IF P>1-5E-6 THEN 1310
1340 IF P<5E-6 THEN 1310
1350 RETURN
1351 STOP
1360 PRINT
1370 PRINT"THESE ARE INSTRUCTIONS FOR USING THE PROGRAM         "
1380PRINT" IT     PERFORMS SEVERAL TYPES OF CALCULATIONS FOR THE"
1390 PRINT"NORMAL AND STUDENT'S-T DISTRIBUTIONS.  GIVEN ONE- OR"
1400 PRINT"TWO-TAILED PROBABILITIES, IT WILL PRODUCE THE CORRES-"
1410 PRINT"PONDING LIMITS; GIVEN A VARIATE OR A SET OF LIMITS, IT"
1420 PRINT"WILL PRODUCE THE CORRESPONDING PROBABILITIES.  DATA IS"
1430 PRINT"ENTERED AS FOLLOWS:"
1440 PRINT
1450 PRINT"   100 DATA  DF, MEAN, SIGMA, (BLOCK1), (BLOCK2),....."
1460 PRINT
```

```
1470 PRINT"WHERE DF IS DEGREES OF FREEDOM (SETTING DF=ZERO SIGNALS"
1480 PRINT"NORMAL DISTRIBUTION), MEAN AND SIGMA ARE THE MEAN AND"
1490 PRINT"STANDARD DEVIATION OF THE DISTRIBUTION, AND THE DATA"
1500 PRINT"BLOCKS MAY BE ANY MIXTURE OF THE FOUR TYPES SHOWN BELOW:"
1510 PRINT
1520 PRINT"   BLOCK(INPUT)              OUTPUT PRODUCED"
1530 PRINT
1540 PRINT"    1,U              PROB OF X EXCEEDING THE VALUE U"
1550 PRINT"    2,L,U            PROB OF X INSIDE LIMITS L AND U"
1560 PRINT"    3,P              VALUE WHICH IS EXCEEDED WITH PROB P"
1570 PRINT"    4,P              LIMITS CONTAINING X WITH PROB P"
1571 PRINT
1572 PRINT" FOR EXAMPLE, THE LINE OF DATA SHOWN BELOW WOULD PRODUCE"
1573 PRINT"AS OUTPUT THE PROBABILITY OF A NORMALLY DISTRUBUTED"
1574 PRINT"VARIABLE WITH MEAN 100 AND STANDARD DEVIATION 10 BEING"
1575 PRINT"IN THE RANGE 95-105, AND ALSO THE VALUE OF THE VARIABLE"
1576 PRINT"THAT WILL BE EXCEEDED ONLY 2.5 PERCENT OF THE TIME:"
1578 PRINT
1579 PRINT"       100 DATA 0,100,2,95,105,3,0.025"
1580 PRINT
1581 PRINT
1582 PRINT"NOTE THAT DATA FOR ADDITIONAL CASES MAY BE CONTINUED"
1583 PRINT"ON LINES 101-199 AS NEEDED."
1600 PRINT"END OF INSTRUCTIONS            "
1700 STOP
9999END
```



T-DISTRIBUTION

```
100 DATA  0,1550000,6000,1,1650000,1,1700000,1,1450000,2,1500000
101 DATA 1600000,3,.75,4,.5

RUN


CALCULATIONS FOR A NORMAL DISTRIBUTION
HAVING A MEAN OF     1550000 AND A SIGMA OF    6000

CASE    THE PROBABILITY OF A VARIATE:

  1       EXCEEDING       1650000 IS 0.0476
  2       EXCEEDING       1700000 IS 0.00621
  3       EXCEEDING       1450000 IS 0.9524
  4       BETWEEN        1500000 AND    1600000 IS 0.595586
  5       EXCEEDING       1500953 IS 0.75
  6       BETWEEN        1500953 AND    1599047 IS 0.5
```

EXAMPLE #2

```
100 DATA 17,0,1,3,.025

RUN


CALCULATIONS FOR A STUDENT'S T-DISTRIBUTION
HAVING A MEAN OF      0 AND A SIGMA OF    1
AND HAVING      17 DEGREES OF FREEDOM.

CASE    THE PROBABILITY OF A VARIATE:
     .
  1       EXCEEDING    2.109843 IS  0.025
```

UNPAIRED:

DESCRIPTION

This program compares various statistical quantities between two groups of unpaired data. These two groups may have unequal variances.

USERS

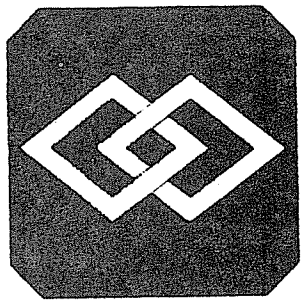
Again users of this program will be individuals who analyze data, regardless of the source.

INSTRUCTIONS

Enter the data for group one (1) starting in line 900 and enter the corresponding data for group two (2), starting in line 950, before program execution. For additional information list the program.

LIMITATIONS

This program should store and execute in 3K Bytes of memory in most computer systems.



UNPAIRED

```
20 REM  ***DESCRIPTION:  COMPUTES THE MEANS, VARIANCES, AND T-RATIO
30 REM  FOR TWO GROUPS OF UNPAIRED DATA.  THIS PROGRAM ASSUMES
40 REM  THAT THE GROUPS MAY HAVE UNEQUAL VARIANCES.
50 REM  ***INSTRUCTIONS FOR USE:  PUT DATA IN LINE 900 AND FOLLOWING.
60 REM  MAKE SURE THE DATA LINE NUMBERS DO NOT EXCEED 998.
70 REM  END THE FIRST SERIES OF DATA WITH 999999, AND THEN
80 REM  TYPE IN THE SECOND SERIES, AGAIN ENDING WITH 999999.
90 DIM S(10),Z(10),N(10),M(10),V(10),D(10)
100 LET P = 1
110 LET S = 0
120 LET S2 = 0
130 LET N = 0
140 READ X
150 IF X = 999999 THEN 210
160 LET S = S + X
170 LET S2 = S2 + X*X
180 LET N = N + 1
190 GO TO 140
210 LET S(P) = S
220 LET Z(P) = S2
230 LET N(P) = N
240 IF P=2 THEN 280
250 LET P = 2
260 GO TO 110
270 REM  NEW WE PRINT THE ANSWERS
280 PRINT TAB(7);"GROUP";TAB(21);"NUMBER"TAB(38);"MEAN";
290 PRINT TAB(49);"VARIANCE";TAB(63);"STD. DEV."
300 PRINT
310 FOR I = 1 TO 2
320 LET M(I) = S(I)/N(I)
330 LET V(I) = (N(I)*Z(I) - S(I)*S(I))/N(I)/(N(I) - 1)
340 LET D(I) = SQR(V(I))
350 PRINT I,N(I),M(I),V(I),D(I)
360 NEXT I
370 LET Q = V(1) / N(1) + V(2) / N(2)
380 LET W = SQR(Q)
390 390 LET R = M(1) - M(2)
400 LET K = V(1) / N(1) / Q
410 LET D = 1/( K*K/(N(1) - 1) + (1-K)*(1-K)/(N(2) - 1) )
420 PRINT
430 PRINT "MEAN DIFF.","VAR.DIFF.","STD. DEV. DIFF."
440 PRINT R, Q, W
450 PRINT
```

```
460 PRINT "T RATIO"; R/W ; "ON"; D;"DEGREES OF FREEDOM."
470 STOP
480 PRINT R,Q,W
490 PRINT
500 PRINT "T RATIO"; R/W; "ON"; N(1);N(2)-2; "DEGREES OF FREEDOM."
510 STOP
9999 END
```

```
UNPAIRED
```

```
900 DATA 73,43,47,53,58,47,52,38,61,56,56,34,55,65,75,999999
910 DATA 51,41,43,41,47,32,24,43,53,52,57,44,57,40,68,999999
```

```
RUN
```

```
     GROUP          NUMBER     MEAN      VARIANCE          STD. DEV.

       1              15        54.2      134.0286          11.57707
       2              15        46.2      116.0286          10.77166

MEAN DIFF.          VAR. DIFF.          STD. DEV. DIFF.
     8               16.67048              4.082949

T RATIO              1.959368    ON     27.85566  DEGREES OF FREEDOM.
```

VARIANCE 1:

DESCRIPTION

This program performs a one way analysis of variances with equal sample sizes. There may be several different samples as long as each sample has the same number of data points.

USERS

Metalurgists, experimenters, and persons who would like to compare system operating characteristics could all use this program effectively. In the example given, the program must analyze the effects of an impurity on the tensile properties of three sample alloys.

INSTRUCTIONS

Starting in line 100 enter the data in Data Statements before the program is run. Use the following format for entering the data:

```
100 DATA  M,N
200 DATA  X11,X12,X13,.........
201 DATA  X21,X22,X23,.........
          and so forth until all the samples are entered
```

where

M = The number of samples
N = The number of data per sample (all samples must be the same size)
XIJ = The sample data (the data may be contained on additional lines if necessary)

List Variance 1 for additional information.

LIMITATIONS

In line 27 there is a DEF FND statement. This program will require 3K Bytes of memory for storage and will execute in 4K Bytes.

```
11 DATA 5000000,5398278,5792597,6179114,6554217,6914625,7257469
13 DATA 7580363,7881446,8159399,8413447,8643339,8849303,9031995
15 DATA 9192433,9331928,9452007,9554345,9640697,9712834,9772499
17 DATA 9821356,9860966,9892759,9918025,9937903,9953388,9965330
19 DATA 9974449,9981342,9986501,9990324,9993129,9995166,9996631
21 DATA 9997674,9998409,9998922,9999277,9999519,9999683,9999793
23 DATA 9999867,9999915,9999946,9999966,9999979,9999987,9999992
25 DIM X(49)
27 DEF FND(U) = X(U+1) - X(U)
100 GO TO 630
300 FOR I = 1 TO 49
305 READ X(I)
310 NEXTI
320 DIM U(20,50),T(20)
325 READ M,N
330 FORI=0TO20
335 LETT(I)=0
340 NEXTI
345 LET G=0
350 LET R=0
355 LET W=0
360 FOR I=1 TO M
365 FOR J=1 TO N
370 READ U(I,J)
375 LET T(I)=T(I)+U(I,J)
380 LET R=R+U(I,J)↑2
385 NEXT J
390 LET G=G+T(I)
395 LET W=W+T(I)↑2
400 NEXT I
405 LET A=(W-(G↑2)/M)/(N*(M-1))
410 LET B=(R-W/N)/(M*(N-1))
415 LET F=A/B
420 LET F1=F↑(1/3)
425 LET A1=2/(9*(M-1))
430 LET A2=2/(9*M*(N-1))
435 LET B1=(F1*(1-A2)+(A1-1))/SQR(A2*F1↑2+A1)
440 GOSUB 535
445 PRINT
450 PRINT "A N A L Y S I S   O F   V A R I A N C E"
455 PRINT "      ";M;"SAMPLES OF SIZE ";N
```

```
460 PRINT
465 PRINT "SAMPLE","SAMPLE TOTAL","SAMPLE MEAN"
470 PRINT
475 FOR I=1 TO M
480 PRINT I,T(I),T(I)/N
485 NEXT I
490 PRINT
495 PRINT "MEAN SQUARE (BETWEEN SAMPLES) = ";A
500 PRINT "MEAN SQUARE (WITHIN SAMPLES)  = ";B
505 PRINT "CALCULATED VALUE OF F-RATIO    = ";F
510 PRINT "CORRESPONDING NORMAL DEVIATE   = ";B1
515 PRINT
520 PRINT " THE PROBABILITY OF AN F-RATIO THIS LARGE"
525 PRINT " OCCURRING BY CHANCE ALONE IS";1-B2
530 STOP
535 IF B1<-4.5 THEN 590
540 IF B1<0 THEN 575
545 IF B1<4.5 THEN 560
550 LET B2=1
555 RETURN
560 GOSUB 600
565 LET B2=Q
570 RETURN
575 GOSUB 600
580 LET B2=1-Q
585 RETURN
590 LET B2=0
595 RETURN
600 LET Z=10*ABS(B1)
605 LET K=INT(Z)
610 LET D1=Z-K
615 LET Q=X(K+1)+D1*FND(K+1)+(D1*(D1-1)/2)*(FND(K+2)-FND(K+1))
620 LET Q=1E-3*INT(.5+1E-4*Q)
625 RETURN
630 PRINT
635 PRINT "THIS PROGRAM WILL PERFORM A ONE-WAY ANALYSIS OF"
640 PRINT "VARIANCE WITH EQUAL SAMPLE SIZES.  TO USE, TYPE:"
645 PRINT
650 PRINT "    100 DATA  M,N"
655 PRINT "    200 DATA  X11,X12,X13,...,X1N"
660 PRINT "    201 DATA  X21,X22,X23,...,X2N"
665 PRINT "     ETC UNTIL"
670 PRINT "    2XX DATA  XM1,XM2,XM3,...,XMN"
675 PRINT
680 PRINT "WHERE M IS THE NUMBER OF SAMPLES, N IS THE COMMON"
685 PRINT "SAMPLE SIZE, AND THE XIJ ARE THE OBSERVATIONS."
690 END
```

```
VARIANCE 1


100 DATA 3,4
200 DATA 181.4,168.6,174,183.2
201 DATA 166.3,172.1,179.2,159.7
202 DATA 176.4,183.2,189.4,177.5



RUN


ANALYSIS OF VARIANCE

       3 SAMPLES OF SIZE      4

SAMBLE          SAMPLE TOTAL     SAMPLE MEAN

      1             707.2            176.8
      2             677.3            169.325
      3             726.5            181.625

MEAN SQUARE (BETWEEN SAMPLES)=     153.6328
MEAN SQUARE (WITHIN SAMPLES) =     50.16145
CALCULATED VALUE OF F-RATIO  =     3.062766
CORRESPONDING NORMAL DEVIATE =     1.305796

THE PROBABILITY OF AN F-RATIO THIS LARGE
OCCURRING BY CHANCE ALONE IS     0.096
```

VARIANCE 2:

DESCRIPTION

This program calculates the analysis of a variance table for a one-way design. The design may be completely randomized. The input data must be in the form of a table where different events and their observations are listed.

USERS

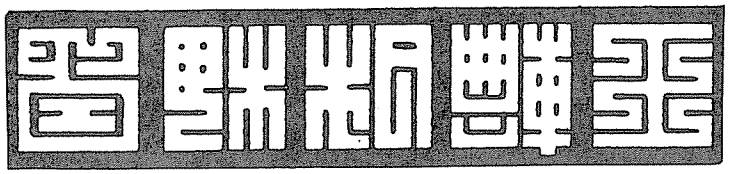
Statisticians will find the most use for this program.

INSTRUCTIONS

Before the program is run the problem data must be entered, starting in line 900. For a complete set of instructions for program operation list the program. After the data has been entered type RUN.

LIMITATIONS

Line 91 contains a Restore statement and line 115 contains a MAT Read statement. In the DIM statement in line 100, provision is made for a two dimensional array. The source code for this program requires 3K Bytes of memory for program storage and it will require 6K Bytes of memory for storage and execution.



VARIANCE 2

```
10  REM  DESCRIPTION: COMPUTES THE ANALYSIS OF VARIANCE TABLE
20  REM  FOR A ONE-WAY COMPLETELY RANDOMIZED DESIGN.
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) AND FINALLY, THE OBSERVATIONS THEMSELVES, FIRST
83  REM     FOR TREATMENT 1, THEN TREATMENT 2, ETC.
85  REM  IF ANY N(J) > 20, CHANGE THE DIMS IN LINE 100.
86  REM  IF M > 10, CHANGE THE DIMS IN LINE 100.
90 READ 00
91 RESTORE
92 IF 00 <> 999999 THEN 100
93 PRINT "LIST LINES 10 TO 86 FOR INSTRUCTIONS"
94 STOP
100 DIM X(20,10),N(10),T(10),S(10)
110 READ A, M
115    MAT READ N(M)
120 FOR J = 1 TO M
130 FOR I = 1 TO N(J)
140 READ X(I,J)
150 NEXT I
160 NEXT J
170 FOR J = 1 TO M
180 FOR I = 1 TO N(J)
190 LET T(J) = T(J) + X(I,J)
200 LET S(J) = S(J) + X(I,J)*X(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";TAB(25);"SS";TAB(40);"DF";TAB(55);"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 LET B1=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(1E5*P + .5)/1E5
827 PRINT "EXACT PROB. OF F=";G;"WITH ( "M;", "N;" ) D.F. IS ";P
828 PRINT
829 RETURN
9999 END
```

XY:

DESCRIPTION

This plotting program plots single valued functions in X on a standard teletype. The X axis is the vertical axis. The plots that are drawn are linear plots. Log plots may be done by setting Y = LOG(f(X)) but the axis will still indicate a linear grid.

USERS

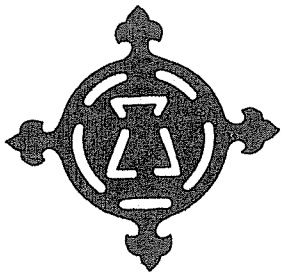
Anyone who would like to plot functions whether for business or pleasure will find this program easy to use.

INSTRUCTIONS

The function that is to be plotted will have to be entered in line 2030 before the program is run. After the function has been entered type RUN. For additional information list the program.

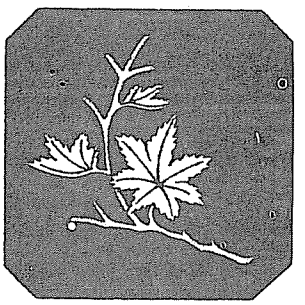
LIMITATIONS

This program should execute without incident in most systems that have 3K Bytes of available memory space.

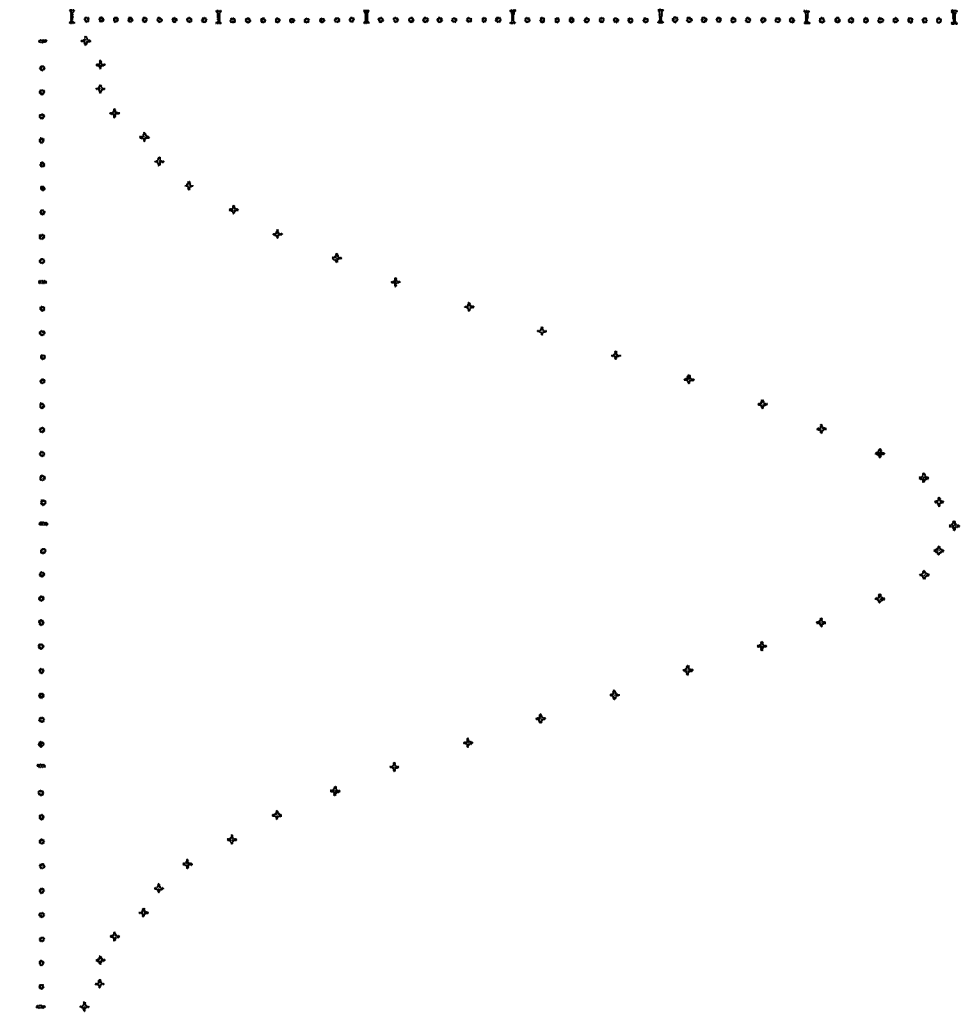


```
XY



2015 LET J=0
2030 GO TO 2810
2045 IF J=0 THEN 2090
2060 RETURN
2075 PRINT "



                                                XY



                                                                "
2090 PRINT
2105 PRINT "WHAT ARE  YMIN,YMAX,XMIN,XMAX,DELX";
2120 INPUT Q0,Q1,Q2,Q3,Q4
2135 LET Q5=(Q1-Q0)/60
2150 LET Q6=0
2165 LET J=2
2180 FOR X = Q2 TO Q3 STEP Q4
2195 GOSUB 2030
2210 IF Q6 = 0 THEN 2600
2225 IF Q6 = 20 THEN 2270
2240 PRINT " . ";
2255 GOTO 2300
2270 PRINT " - ";
2285 LET Q6 = 10
2300 IF Y > Q1 THEN 2555
2315 IF Y < Q0 THEN 2555
2330 LET Q7 = Q0 + 2*Q5
2345 LET Z=Q7+0.5*Q5
2360 IF Z<Y THEN 2510
2375 LET Q6 = Q6 + 1
2390 IF Z-Y>=2*Q5 THEN 2480
2405 IF Z-Y>=Q5 THEN 2450
2420 PRINT "  +"
2435 GOTO 2690
2450 PRINT " +"
2465 GOTO 2690
2480 PRINT "+"
2495 GOTO 2690
2510 LET Q7 = Q7 + 3*Q5
2525 PRINT "   ";
```

```
2540 GOTO 2345
2555 PRINT "OFF SCALE:  (X,Y) =  ";X;",  ";Y
2570 LET 06 = 06 + 1
2585 GOTO 2690
2600 PRINT
2615 PRINT "FOR X:     TOP = ";02;" BOTTOM = ";03;" INCREMENT = ";04
2630 PRINT "FOR Y:   LEFT = ";00;"  RIGHT = ";01;" INCREMENT = ";05
2645 PRINT
2660 PRINT "   I.........I.........I.........I.........I.........I.........I.........I"
2675 GOTO 2270
2690 NEXT X
2705 PRINT
2720 PRINT "TYPE '0' TO STOP OR '1' TO CHANGE LIMITS.  WHICH";
2735 INPUT 08
2750 IF 08 = 0 THEN 3005
2765 IF 08 = 1 THEN 2090
2780 PRINT
2795 GOTO 2705
2810 PRINT
2825 PRINT "THIS PROGRAM WILL PLOT SINGLE-VALUED FUNCTIONS OF X,"
2840 PRINT "WITH X ON THE VERTICAL AXIS.  TO USE, TYPE:"
2855 PRINT
2870 PRINT " 2030 LET Y= (THE FUNCTION TO BE PLOTTED)"
2885 PRINT "   RUN"
2900 PRINT
2915 PRINT "DURING RUNNING, THE PROGRAM WILL ASK FOR YMIN AND"
2930 PRINT "YMAX (THE LIMITS ON THE HORIZONTAL Y-AXIS), FOR"
2945 PRINT "XMIN AND XMAX (THE LIMITS ON THE VERTICAL X-AXIS),"
2960 PRINT "AND FOR DELX, THE INCREMENT TO BE USED ALONG THE"
2975 PRINT "X-AXIS.  NOTE: LINES 70-99 OF THE PROGRAM MAY BE"
2990 PRINT "USED AS DESIRED TO EXPRESS COMPLICATED FUNCTIONS."
3005 END
```



```
*2030   Y=EXP(-(X↑2))
*RUN


WHAT ARE  YMIN,YMAX,XMIN,XMAX,DELX ?0,1,-2,2,.1

FØR X:    TØP =    -2  BØTTØM =     2  INCREMENT =   0.1
FØR Y:   LEFT =     0   RIGHT =     1  INCREMENT =   0.0166667
```



```
TYPE '0' TØ STØP ØR '1' TØ CHANGE LIMITS.  WHICH ?0

READY
```

APPENDIX A

BASIC

STATEMENT

DEFINITIONS

SEE APPENDIX B FOR CONVERSION ALGORITHMS FOR OTHER BASICS

STATEMENT DEFINITION

Each BASIC statement consists of the following elements arranged in the order given:

Statement (or line) number - by its ascending order, indicates the processing sequence of the statement.

BASIC word - specifies the computer operation to be performed.

Parameters - in most statements are variables, expressions, and numbers used in or to direct the operation performed by the statement.

MATHEMATICAL NOTATION AND OPERATIONS WITHIN A STATEMENT

Variable Representation

In the BASIC language, a variable can be represented by

1. a letter
2. a letter and a digit
3. either of the above, followed by the character $

For example A,Z,K6, and X may represent variables, but AR, Z12, 6K, and 22 can not. The inadvertent use of the digit 0 for the letter O (and vice versa) in a variable will cause errors in a program; use of the letter O or the digit 0 in variable representation is not recommended. The user may find choice of a letter as a mnemonic for a variable helpful; for example, P for price, S for sales, and N for numbers.

Variables with $'s are restricted to the assignment of strings (alphanumeric data) and are referred to as "string variables", in contrast to variables without the $ that are referred to as "numeric variables". Numeric variables, when used as a starting point in calculations (e.g., for a counter), have an initial value of zero. String variables have an initial value of zero when used for character count.

A BASIC variable is assigned a value, during the execution of a program, from the numbers given in a related LET, FOR, READ, or INPUT statement. It retains this value during the processing, unless it is reassigned a new value by another of these statements.

List and Table Variables

Subscripted variables are represented in BASIC as

variable name (subscript)
or
variable name (subscript, subscript)

where the subscript can be an integer, variable, or an arithmetic expression such as (1+K) or (A(3,7),B-C). The subscript must always be enclosed by parentheses. Subscript values should begin at 1 (i.e., not 0).

A list variable designates an element of a one-dimensional array that can be represented by such as P(15), P(H) or L(20). Before a list variable can be used in any statement, the maximum value of its subscript (i.e., size of list) must be specified in a DIM statement; otherwise a list of 10 or less is implied.

A table variable designates an element of a two-dimensional array that can be represented by such as S(15,17) or T(20,30). Before a table variable can be referenced in any statement, the maximum value of its subscripts must be specified in a DIM statement; otherwise, subscripts of 10 or less are implied.

Use of Numbers

A number may be positive or negative, may contain up to nine digits, and must be in decimal form. BASIC would accept 0.01, 2, -3.675, 123456789, -.987654321, and 483.4156 as numbers, but would reject 14/3 (this is an expression) or 32,437 (as representing 32437). Numbers are stored as single-precision floating-point values.

A number can also be expressed in "E notation", which is equivalent to expressing it as a power of 10. For example, in E notation,

0.00123456789	may be expressed as	0.123456789E-2 or 12.3456789E-4
1967		1.967E3 or 19.67E2
10,000,000		1E7 or 100E5

The decimal point can be positioned anywhere within the number as long as the integer following the E indicates its correct position. Note that E and an exponent alone cannot represent a number. For example, E7 cannot be written as a number to represent 10,000,000; it must be written as 1E7 to indicate 1 multiplied by 10 to the 7th power.

Arithmetic Operations

Five arithmetic operations can be performed by BASIC. Each of the following symbols represents an arithmetic operation that can be included in an expression.

Operator symbol	denotes	as illustrated by
+	addition	A + B
-	subtraction	A - B
*	multiplication	A * B
/	division	A / B
↑ or **	raise to a power	A ↑ B or A ** B

Relational Symbols

Six relational tests can be made with BASIC. Symbols representing these relationships can be used in statements when comparisons are required. The symbols and illustration of their use follow.

Relational symbol	denotes	as illustrated by
=	is equal to	A = B
<	is less than	A < B
<= or =<	is less than or equal to	A <= B or A =< B
>	is greater than	A > B
>= or =>	is greater than or equal to	A >= B or A => B
<> or ><	is not equal to	A <> B or A >< B

Function	means find the
SIN(X)	sine of X
COS(X)	cosine of X
TAN(X)	tangent of X
COT(X)	cotangent of X
ATN(X)	arctangent of X
EXP(X)	e to the power X
LOG(X)	natural logarithm of X
CLG(X)	common logarithm of X
ABS(X)	absolute value of X
SQR(X)	square root of X

In these definitions, the letter X represents an expression, which, for the trigonometric functions, implies an angle measured in radians. If the value of X in LOG(X), CLG(X), or SQR(X) is negative, then the negative sign is ignored, the positive value is used, and an error message is printed.

Four additional mathematical functions are included in BASIC.

Function	means
INT(X)	truncate X
RND(X)	produce a random number
SGN(X)	sign determination
DET(X)	provide determinant of last matrix inverted

In addition, the user may employ the DEF statement to define one or more of his own functions.

Miscellaneous Functions

A set of miscellaneous functions is available for use to provide a variety of non-mathematical operations. These are as follows:

Function	means obtain
TIM(X)	elapsed processor time
CLK$	time of day
DAT$	calendar date
NUM(X)	count of matrix data elements
SST(X$,Y,Z)	selected characters of a string (substring)
TAB(X)	character print position
SPC(X)	space print position
LEN(X$)	number of characters in string
LIN(X)	last line number encountered in reading/writing file

BASIC WORDS

BASIC words are short, distinctive, easily recognizable words that are either valid words or abbreviations of words. When formatted into a statement, a word becomes an explicit instruction to the computer to perform some operation. Some statements can be made by the use of a BASIC word alone; other statements require other information in addition to the BASIC word.

BASIC words may be grouped by type of statements in which they occur. The words and their associated functional statement categories are as follows:

Arithmetic Statements

BASIC Word	When formatted into a statement
DEF	- defines a repeatedly used function
LET	- requests a computation or manipulation upon an arithmetic variable
MAT	- requests a computation or manipulation upon a matrix

Specification Statements

BASIC WORD	When formatted into a statement
CHANGE	- converts string characters to numerical code or vice versa
DATA	- specifies numeric values for variables listed in a READ statement
DIM	- reserves space for list or table

Input/Output Statements

BASIC Word	When formatted into a statement
INPUT	- delays input of values to variables until program is in execution; program will request input of data by terminal user or a user's file when statement is executed
PRINT	- prints computed results; prints text - prints computed results and text - skips lines - formats output data
PRINT USING	- formats output line
READ	- reads values from a DATA statement or user's file and assigns them to designated variables
RESTORE	- restores previously processed blocks of input data from DATA statements

Loop and Subroutine Statements

BASIC Word	When formatted into a statement
CALL	- directs processing sequence to a subroutine previously saved
FOR	- is first statement of a loop and sets conditions of loop
NEXT	- is last statement of loop
GOSUB	- directs processing sequence to a subroutine
RETURN	- returns processing sequence from a subroutine

Logic Statements

BASIC Word	When formatted into a statement
GOTO	- unconditionally transfers the processing sequence to a designated statement
IF-----THEN or IF---GOTO-	conditionally transfers the processing sequence to a designated statement

ON-----THEN
or
ON---GOTO- conditionally transfers the processing sequence to designated statements

STOP - stops the execution of the program

END - indicates end of program

Documentation Statement

BASIC Word	When formatted into a statement
REM	- inserts a remark into the statement sequence

For all Matrices operations the subscripts are defined as follows:

```
     DIM    A(3,4)
where
     3 = # of rows
     4 = # of columns
```

ie: If the following data were input to the above A matrix it would be stored as:

Input Data = 1,2,3,4,5,6,7,8,9,10,11,12

Matrix				
	1	2	3	4
	5	6	7	8
	9	10	11	12

Function ASC(X)

Purpose: To provide the numeric value of a specified character or, for the case of non-printing characters, an abbreviation.

Format: ASC $\left\{\begin{matrix}\text{(character)}\\\text{(abbreviation)}\end{matrix}\right\}$

Function SPC(X)

Purpose: To insert spaces at indicated positions within an output line.

Format: SPC(expression); <data to be printed >

Function TAB(X)

Purpose: To position data field at indicated character position within an output line.

Function NUM(X)

Purpose: To supply count of number of data elements in response to request from MAT INPUT statement.

Format: NUM (any alphanumeric character)

File read statements

```
READ # file designator, input list
INPUT # file designator, input list
```

File write statements

```
WRITE # file designator, output list
PRINT # file designator, output list
PRINT # file designator, USING statement number, output list
```

Matrix input statements

```
MAT READ # file designator, matrix input list
MAT INPUT # file designator, matrix input list
```

Matrix output statements

```
MAT WRITE # file designator, matrix output list
MAT PRINT # file designator, matrix output list
```

File manipulation statements

```
SCRATCH # file designator
RESTORE # file designator
BACKSPACE # file designator
```

Function LEN(X$)

Purpose: To determine the number of characters in a specified string variable.

Format: LEN(string variable)

Function VAL(S$)

Purpose: To produce a numeric value corresponding to the value of a string represented by a string variable.

Format: VAL (string variable)

Function SST(X$,Y,Z)

Purpose: To extract selected characters of a string.

Format: SST(string variable, beginning character, number of characters)

Function STR$(N)

Purpose: To produce a string corresponding to a value of a number represented by an expression.

Format: STR$ (expression)

Use of STR$ implies placement of the string right-justified in the smallest zone into which it will fit. Blanks will occupy the remaining character positions of the zone.

CHANGE

The change statement may be used to convert string characters to equivalent numeric code or vice versa.

The process involves two lists, one numeric, the other a string variable. When converting numeric codes to a character string, the numeric list is to contain the valid numeric equivalent of a single character in each element. Given the desired number of items to convert, the CHANGE command will perform the conversion and concatenate the resulting characters into the string variable.

In changing from a character string, the command stores the related numeric code for each character into the elements of the numeric array.

String concatenation is limited within one LET statement to two string variables or one string constant and one string variable.

The statements

```
*10 LET A$ = "JOHN DOE "
*20 LET B$ = "EMPLOYEE NUMBER 12345"
*30 LET C$ = A$ & B$
*40 PRINT C$
```

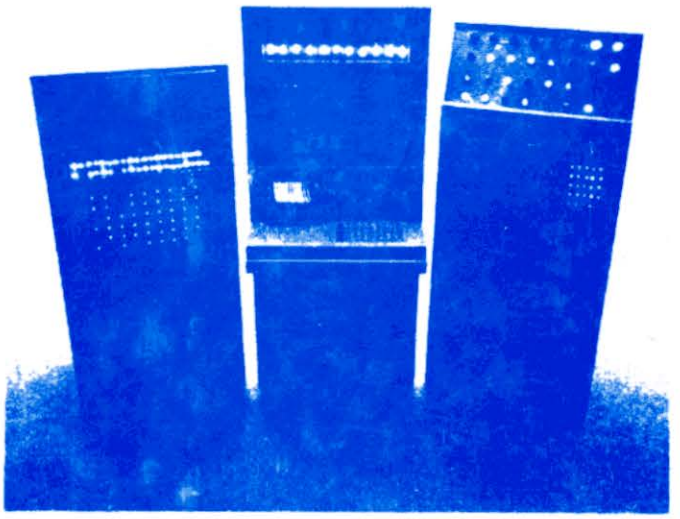
or

```
*10 LET A$ = "JOHN DOE "
*20 LET C$ = A$ & "EMPLOYEE NUMBER 12345"
*30 PRINT C$
```

when executed, will produce the printout

```
JOHN DOE EMPLOYEE NUMBER 12345
```

RELIABLE COMPUTER SOFTWARE



FOR YOUR DOWN TO EARTH TASKS