

\$ 9.95

BASIC SOFTWARE

LIBRARY

VOLUME IV

GENERAL PURPOSE PROGRAMS

by R. W. BROWN



BASIC

SOFTWARE

LIBRARY

VOLUME IV

GENERAL

PURPOSE

PROGRAMS

THIS BASIC SOFTWARE LIBRARY IS MADE AVAILABLE, FREE OF RESTRICTIONS AND ROYALTIES TO SCHOOLS, COLLEGES, UNIVERSITIES, INDIVIDUALS, HOBBYIST & BUSINESS CONCERNS FOR USE ON THEIR OWN COMPUTERS AND OR COMPUTING 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 EXPRESSED WRITTEN PERMISSION OF SCIENTIFIC RESEARCH.

1ST. PRINTING - - JANUARY 1977

COPYRIGHT UNDER UCC 1977 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

<u>NAME</u>	<u>DESCRIPTION</u>
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

<u>NAME</u>	<u>DESCRIPTION</u>
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; between 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

<u>NAME</u>	<u>DESCRIPTION</u>
	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

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

TABLE OF CONTENTS

VOLUME TWO (CONT.)

Part 4 - Plotting & Statistics Programs

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

APPENDIX A - BASIC STATEMENT DEFINITIONS

TABLE OF CONTENTS

VOLUME THREE

Part 5 - Advanced Business Programs

<u>NAME</u>	<u>DESCRIPTION</u>
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

PAGE

General Purpose Programs

<u>NAME</u>	<u>DESCRIPTION</u>	
Bingo	An age old favorite. "B9, C23, D4, E13, F21, BINGO!	684
Bonds	Computes the yields for a bond for different periods.	690
Bull	If you ever dreamed of being a Matador, here's your chance.	696
Enterprise	Take charge of the Enterprise while Capt. Kirk is on leave.	702
Football	Authentic NFL version of this well known sport.	710
Funds 1	Calculates long-term predictions of funds.	722
Funds 2	Plots the results of Funds 1.	728
Go-Moku	Ancient Chinese game of chance.	736
Jack	Plays Blackjack, Las Vegas style.	742
Life	Life is truly a battle for survival, a real challenger!	750
Loans	Calculates annuities, loans and mortgages.	763
Mazes	Generates unique maze puzzles for you to solve.	771
Poker	Five card draw - for up to 5 players.	777
Popul	Performs population projections for defined areas.	784
Profits	Determines the profitability of a firms various depts.	787
Qubic	3-Dimensional Tic-Tac-Toe.	791
Rates	Calc. the effective annual interest rate for stated interest.	799
Retire	Calculates your Civil Service Retirement benefits.	803
Savings	Computes savings plan profiles.	808
SBA	Calculates repayment schedules for SBA loans.	811
Tic-Tac-Toe	An all time favorite for young and old alike.	817

TABLE OF CONTENTS

VOLUME FIVE

Experimenter's Programs

<u>NAME</u>	<u>DESCRIPTION</u>
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

<u>NAME</u>	<u>DESCRIPTION</u>
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

<u>NAME</u>	<u>DESCRIPTION</u>
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.

P R E F A C E

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-tempted" was used as no work is ever considered complete by everyone regardless 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 presented here are all written in the computer language called BASIC. Each program 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 program, tells who might be interested in using it, a brief set of instructions or how to get them and then any limitations in the program are noted. 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 occupied 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 buying, 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 overlay 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 pages, 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, engineers, astronomers, airplane pilots, etc. Most of these programs are very

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

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

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

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

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

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

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

ACKNOWLEDGMENTS

MY WIFE MARY AND MY FAMILY

DONALD ALVAREZ

GETIMESHARING

DAVE BEETLE

BILL JONES

MORTON BERGER

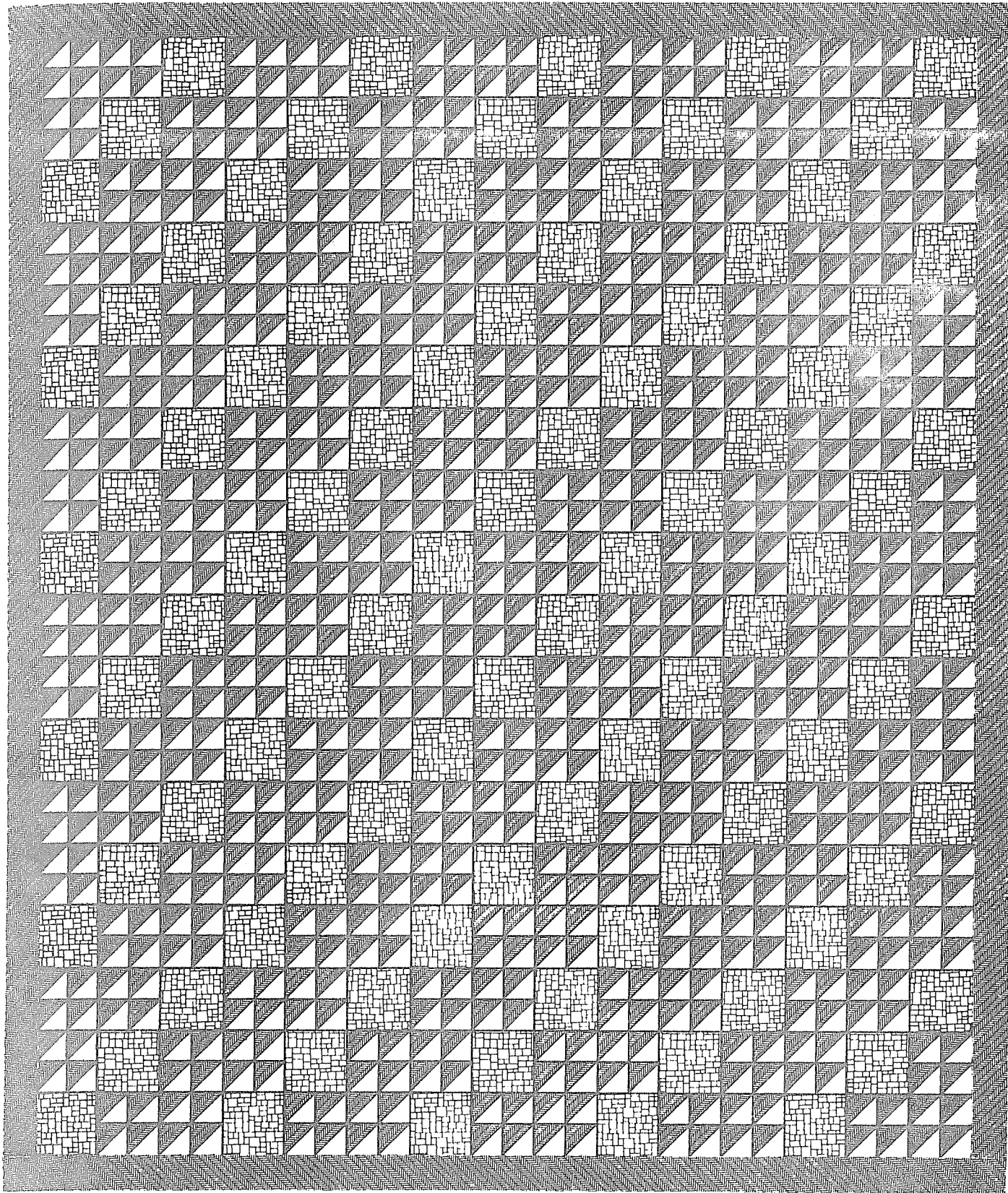
GEORGE LONG

COPYCAT INC

TOM ROSE

ARTWORK COURTESY OF

MELISSA



GENERAL

PURPOSE

PROGRAMS

BINGO:

DESCRIPTION

This program simulates that old family favorite "Bingo". The game is played between you and the computer, although another may play using the computer's card. The computer will tumble the balls and select the numbers at random.

USERS

Individuals who enjoy matching their luck against chance will like playing this game. Bingo will also appeal to persons who enjoy a leisure game of chance without the tension of house betting.

INSTRUCTIONS

After the program is loaded into memory type RUN. All necessary instructions and inputs will be prompted as this program is self instructing. For a detailed operation of the game, list the program.

LIMITATIONS

This program uses two dimensional arrays; see line 210. Line 230 contains a Restore statement and line 490 contains a SPC () statement. The SPC () statement is used to set the print position for the Bingo cards and could be replaced by a TAB () statement, if your instruction set does not include the SPC () instruction. A definition of the SPC () statement can be found in Appendix A, in Volume II. The source code requires 6K Bytes of memory for storage and the program will execute in most 8K Basic compilers in only 7K Bytes of memory, this is in addition to the amount of memory required to store your Basic compiler.

BINGO

```
0090 REM
0100 REM  INSTRUCTIONS--TYPE "RUN" AND FOLLOW INSTRUCTIONS.
0110 REM
0120 REM
0130 REM  * * * * * MAIN PROGRAM * * * * *
0140 REM
0150 F=RND(-1)
0160 PRINT
0170 PRINT"YOU ARE NOW GOING TO PLAY A COMPUTERIZED VERSION OF BINGO--"
0180 LET F=1
0190 LET V=0
0200 LET W=0
0210 DIM B(10,10)
0220 DIM L(75),A$(5)
0230 RESTORE
0240 READ A$(1),A$(2),A$(3),A$(4),A$(5)
0250 DATA B , I , N , G , "0 "
0260 REM  THIS SEQUENCE GENERATES THE CARD NUMBERS (LINES 240-430)
0270 FOR K1=1 TO 75
0280 LET L(K1)=0
0290 NEXT K1
0300 PRINT " ", "      HERE'S ";
0310 IF F>1 THEN 340
0320 PRINT"YOUR CARD"
0330 GOTO 350
0340 PRINT"MY CARD"
0350 LET M=16
0360 LET G=F+4
0370 FOR Y=F TO G
0380 FOR X=F TO G
0390 LET R=INT(M*RND(0))
0400 IF R<M-15 THEN 390
0410 IF L(R)<>0 THEN 390
0420 LET B(X,Y)=R
0430 LET L(R)=1
0440 NEXT X
0450 LET M=M+15
0460 NEXT Y
0470 REM  THIS SEQUENCE PRINTS THE CARD (LINES 440-670)
0480 PRINT
```

```

0490 PRINT TAB(9);"-B-";SPC(12);"-I-";SPC(12);"-N-";SPC(12);"-G-";SPC(12);"-O-"
0500 FOR X=F TO G
0510 PRINT
0520 PRINT
0530 PRINT
0540 FOR Y=F TO G
0550 IF B(X;Y)=B(F+2;F+2) THEN 580
0560 PRINT B(X;Y);TAB(1);
0570 GOTO 590
0580 PRINT TAB(39);"FREE";
0590 NEXT Y
0600 NEXT X
0610 PRINT
0620 PRINT
0630 PRINT
0640 PRINT TAB(9);"-----"

0650 PRINT"      [TEAR OFF AFTER MACHINE AUTOMATICALLY ADVANCES PAPER]"
0660 FOR K9=1 TO 9
0670 PRINT
0680 NEXT K9
0690 IF F=6 THEN 720
0700 LET F=6
0710 GOTO 270
0720 PRINT
0730 PRINT"NOW WE'RE ALL SET TO PLAY THE GAME.  USE A PENCIL TO MARK OFF"
0740 PRINT"THE NUMBERS ON YOUR CARD AS THEY ARE CALLED."
0750 PRINT" (PLEASE PLAY MY CARD AS WELL AS YOUR OWN)"
0760 PRINT"DON'T GET SMART AND TRY TO CHEAT AS"
0770 PRINT"BIG BROTHER LOVES YOU."
0780 PRINT
0790 PRINT "ARE YOU READY";
0800 INPUT R$
0810 IF R$<>"NO" THEN 850
0820 PRINT
0830 PRINT"*****:HURRY UP:*****"
0840 GOTO 780
0850 REM
0860 FOR K1=1 TO 75
0870 LET L(K1)=0
0880 NEXT K1
0890 PRINT
0900 LET B(3,3)=0
0910 LET B(8,8)=0
0920 REM THIS SEQUENCE GENERATES THE BINGO NUMBERS (LINES 880-1100)
0930 LET U=INT(75*RND(0))+1
0940 IF L(U)<>0 THEN 930
0950 LET L(U)=1

```

```

0960 PRINT
0970 IF RND(0) > .2 THEN 1000
0980 PRINT"IT COMES UP:",TAB(20),
0990 GOTO 1100
1000 IF RND(0) > .4 THEN 1030
1010 PRINT"THE NUMBER IS:",TAB(20),
1020 GOTO 1100
1030 IF RND(0) > .6 THEN 1060
1040 PRINT"WE HAVE A:",TAB(20),
1050 GOTO 1100
1060 IF RND(0) > .8 THEN 1090
1070 PRINT"IT'S A:",TAB(20),
1080 GOTO 1100
1090 PRINT"THE NEXT IS:",TAB(20),
1100 PRINT A$(INT((U-1)/15)+1);U
1110 REM THIS IS THE "NUMBER ON CARD?" SEQUENCE (LINES 1110-1260)
1120 FOR Y=1 TO 10
1130 FOR X=1 TO 10
1140 IF B(X,Y)=U THEN 1260
1150 NEXT X
1160 NEXT Y
1170 LET F=1
1180 GOSUB 1310
1190 LET F=6
1200 GOSUB 1310
1210 IF U=0 THEN 1240
1220 IF W=1 THEN 1950
1230 GOTO 2050
1240 IF W=0 THEN 930
1250 GOTO 2130
1260 LET B(X,Y)=0
1270 GOTO 1150
1280 REM THIS IS THE BINGO DETERMINING SEQUENCE (LINES 1270-1950)
1290 PRINT
1300 REM THIS IS THE VERTICAL CHECK FOR BINGO*** (LINES 1270-1430)
1310 LET G=F+4
1320 FOR Y=F TO G
1330 FOR X=F TO G
1340 IF B(X,Y)<>0 THEN 1460
1350 IF X<F+4 THEN 1450
1360 IF F=6 THEN 1410
1370 PRINT
1380 PRINT"YOU'VE GOT A B I N G O * * * *"
1390 LET W=1
1400 RETURN
1410 PRINT
1420 PRINT"I'VE GOT A B I N G O * * * * *"
1430 LET U=1

```

```

1440 GOTO 1470
1450 NEXT X
1460 NEXT Y
1470 REM THIS IS THE HORIZONTAL CHECK FOR BINGO*** (LINES 1450-1610)
1480 LET G=F+4
1490 FOR X=F TO G
1500 FOR Y=F TO G
1510 IF B(X;Y)<>0 THEN 1630
1520 IF Y<F+4 THEN 1620
1530 IF F=6 THEN 1580
1540 PRINT
1550 PRINT"YOU'VE GOT A  B I N G O * * * "
1560 LET W=1
1570 RETURN
1580 PRINT
1590 PRINT"I'VE GOT A  B I N G O * * * * "
1600 LET U=1
1610 GOTO 1650
1620 NEXT Y
1630 NEXT X
1640 REM THIS IS THE SLANT CHECK (M=-1) FOR BINGO*** (LINES 1620-1770)
1650 LET X=F
1660 LET Y=F
1670 IF B(X;Y)<>0 THEN 1780
1680 LET X=X+1
1690 LET Y=Y+1
1700 IF Y<F+5 THEN 1670
1710 IF Y=11 THEN 1760
1720 LET W=1
1730 PRINT
1740 PRINT"YOU'VE GOT A  B I N G O * * * "
1750 RETURN
1760 PRINT"I'VE GOT A  B I N G O * * * * "
1770 LET U=1
1780 RETURN
1790 REM THIS IS THE SLANT CHECK (M=1) FOR BINGO**** (LINES 1780-1950)
1800 LET X=F+4
1810 LET Y=X
1820 IF B(X;Y)<>0 THEN 1930
1830 LET X=X-1
1840 LET Y=Y+1
1850 IF Y<F+5 THEN 1820
1860 IF Y=11 THEN 1910
1870 PRINT
1880 PRINT"YOU'VE GOT A  B I N G O * * * "
1890 LET W=1
1900 RETURN
1910 PRINT

```

```
1920 LET U=1
1930 RETURN
1940 REM THIS THE TIE PRINTOUT SEQUENCE
1950 PRINT
1960 PRINT" ***** IT'S A TIE *****"
1970 REM THIS IS THE "PLAY AGAIN?" SEQUENCE
1980 PRINT
1990 PRINT "DO YOU WANT TO PLAY AGAIN";
2000 INPUT E$
2010 PRINT
2020 IF E$="YES" THEN 180
2030 STOP
2040 REM THIS IS THE "I WIN" SEQUENCE
2050 PRINT
2060 FOR S=1 TO 6
2070 PRINT"I WIN. ";
2080 NEXT S
2090 PRINT
2100 PRINT
2110 GOTO 1980
2120 REM THIS IS THE "YOU WIN" SEQUENCE
2130 PRINT
2140 PRINT" YOU WIN..... YOU WIN..... YOU WIN....."
2150 GOTO 1980
2160 REM
2170 END
```



BONDS:

DESCRIPTION

Bonds calculates bond yields to maturity for a bond. The yields are computed before and after taxes. The user must supply the tax rates, the redemption value, price, annual coupon and the maturity for the bond before any calculations are made. This program is more comprehensive than the program BOND.

USERS

Persons or companies interested in investing in bonds, or already having bond investments could benefit from using BONDS.

INSTRUCTIONS

The input data for the bond may be entered in data statements prior to program execution or it may be entered after execution through the prompting provided in the program. After the program has been loaded into memory, type RUN. Bonds will then ask if you want instructions. There are three valid responses to this question: YES, NO and DATA. If you answer YES, the program will prompt for the required inputs. If your response is NO, then you are expected to enter the seven (7) required data inputs. If the response is DATA, the program will expect to find the data previously entered in Data Statements prior to operation. Data for a sample problem is contained in line 190.

LIMITATIONS

The program should execute in most Basic compilers with 5K Bytes of available memory. The source code requires 4K Bytes of memory for storage.

801105

```
15REM
20REM DESCRIPTION-----COMPUTES BOND YIELDS
25REM
40REM INSTRUCTIONS----
100REM THIS BASIC PROGRAM COMPUTES THE BEFORE AND AFTER TAX
110REM YIELD TO MATURITY OF A BOND WITH ANNUAL COUPON OF C
120REM DOLLARS, REDEMPTION AT R DOLLARS, AND A MATURITY OF
130REM M1 YEARS AND M2 MONTHS (M2 NEED NOT BE AN INTEGER), AT
140REM A PRICE P. THE TAX RATE APPLIED TO INTEREST RECEIPTS
150REM IS T1; THE RATE APPLIED TO CAPITAL GAINS IS T2;
160REM PREMIUM ON PURCHASE ABOVE THE REDEMPTION AMOUNT IS
165REM AMORTIZED OVER THE MATURITY AND DEDUCTED AT T1; DISCOUNT
170REM AT PURCHASE BELOW REDEMPTION AMOUNT IS TAXED AT MATURITY
175REM AT RATE T2.
178REM
180REM      LINE 190 CONTAINS DATA FOR A SAMPLE PROBLEM
182REM
184REM
190 DATA 5,100,6,5.25,97,.5,.25
200REM
210 LET K2=15
220 LET K3=45
230 PRINT "DO YOU WANT INSTRUCTIONS FOR ENTERING DATA?";
240 INPUT A$
250 PRINT
260 IF A$="DATA" THEN 500
270 IF A$="NO" THEN 460
280 PRINT "ENTER THE FOLLOWING DATA ITEMS"
290 PRINT
300 PRINT "ITEM 1";TAB(K2);"ANNUAL COUPON (IN DOLLARS)";TAB(K3);
310 INPUT C
320 PRINT
330 PRINT "ITEM 2";TAB(K2);"PROCEEDS ON REDEMPTION";TAB(K3);
340 INPUT R
350 PRINT
360 PRINT "ITEMS 3 & 4";TAB(K2);"MATURITY IN YEARS, MONTHS";TAB(K3);
370 INPUT M1,M2
380 PRINT
390 PRINT "ITEM 5";TAB(K2);"PRICE";TAB(K3);
400 INPUT P
410 PRINT
420 PRINT "ITEMS 6 & 7";TAB(K2);"TAX RATES ON INCOME, CAP GNS";TAB(K3);
430 INPUT T1,T2
440 PRINT
```

```

450 GO TO 510
460 PRINT "ENTER ITEMS 1 THROUGH 7";
470 INPUT C,P,M1,M2,P,T1,T2
480 PRINT
490 GO TO 510
500 READ C,P,M1,M2,P,T1,T2
510 PRINT "COUPON";C,"REDEMPTION AT";R,"PRICE";P
520 PRINT "MATURITY",M1;"YEARS",M2;"MONTHS"
530 PRINT "TAX RATES",T1;"ON INCOME",T2;"ON CAPITAL GAINS"
540 PRINT
545REM   SEMI-ANNUAL COUPON
550 LET C=C/2
555REM   TIME TO MATURITY IN SEMI-ANNUAL COUPON PERIODS
560 LET M3=M1*2+M2/6
565REM   NUMBER OF COUPONS INCLUDING FINAL COUPON TO MATURITY
570 LET M4=INT(M3)
575REM   TIME TO NEXT COUPON PERIOD
580 LET M5=M3-M4
585REM   IS BOND SELLING AT A PREMIUM
590 IF P<R THEN 620
595REM   SEMI-ANNUAL AMORTIZATION OF PREMIUM
600 LET D=(P-R)/(M4+1)
610 GO TO 630
620 LET D=0
625REM   AFTER TAX AND AMORTIZATION COUPON
630 LET C2=C-(C-D)*T1
640 LET B=P-D*(M4+1)
645REM   BOOK VALUE AT MATURITY
650 LET G=T2*(R-B)
655REM   MARKET VALUE OF BOND PLUS ACCRUED INTEREST
660 LET U1=P+C*(1-M5)
665REM   REDEMPTION PRICE
670 LET U2=R
675REM   IS TIME TO MATURITY GREATER THAN THREE YEARS
680 IF M3>6 THEN 710
690 GOSUB 1010
700 GO TO 720
710 GOSUB 860
715REM   ANNUAL RATE OF RETURN
720 LET Y=2*(EXP(Y)-1)
730 PRINT "YIELD BEFORE TAX",100*Y;"PERCENT"
740 PRINT
745REM   AFTER TAX MARKET VALUE PRESENT
750 LET U1=P+C*(1-M5)*(1-T1)
755REM   AFTER TAX SEMI-ANNUAL COUPON
760 LET C=C2
765REM   AFTER TAX MARKET VALUE AT MATURITY

```

```

770 LET U2=R-G
775REM   IS TIME TO MATURITY GREATER THAN THREE YEARS
780 IF M3>6 THEN 810
790 GOSUB 1010
800 GO TO 820
810 GOSUB 860
815REM   ANNUAL RATE OF RETURN
820 LET Y=2*(EXP(Y)-1)
830 PRINT "YIELD AFTER TAX",100*Y;"PERCENT"
840 STOP
850REM   ALGORITHM FOR DETERMINING EXACT RATE OF RETURN
860 LET Y = .0001
870 GOSUB 930
880 LET D = Q-(U1+U2+C)
890 IF ABS(D/U3)<.000000005 THEN 920
900 LET Y = Y-D/U3
910 GO TO 870
920 RETURN
930 LET Q=U1*EXP(Y*M5)
940 LET U1 =C*(1-EXP(-M4*Y))/(EXP(Y)-1)
950 LET U2=U2*EXP(-Y*M4)
960 LET U3 = U2*M4-P*M5
970 LET U3=U3+(C/(EXP(Y)-1)+2)*(EXP(Y)-(M4/EXP(Y*(M4-1))))
980 LET U3=U3+(C/(EXP(Y)-1)+2)*(M4-1)/(EXP(Y*(M4-1)))
990 RETURN
1000REM   ALGORITHM FOR DETERMINING EXACT RATE OF RETURN
1010 LET Y=.0001
1020 GOSUB 1080
1030 LET D=Q-(U1+U2)
1040 IF ABS(D/U3)<.000000005 THEN 1070
1050 LET Y=Y-D/U3
1060 GO TO 1020
1070 RETURN
1080 LET U1=U3=0
1090 LET Q=U1
1100 LET U2=U2*EXP(-Y*M3)
1110 FOR I=0 TO M4
1120 LET U1=U1+C*EXP(-Y*(M5+I))
1130 LET U3=U3+(M5+I)*C*EXP(-Y*(M5+I))
1140 NEXT I
1150 LET U3=U3+M3*U2*EXP(-Y*M3)
1160 RETURN
1170 END

```

SAMPLE PROBLEM

TO FIND THE BEFORE AND AFTER TAX YIELDS FOR A BOND, ASSUMING THE FOLLOWING DATA: 1 - A \$5 ANNUAL COUPON; 2 - REDEMPTION VALUE OF \$100; 3 - MATURITY OF 6 YEARS 5-1/4 MONTHS; 4 - A PRICE OF \$97; AND TAX RATES OF 50% ON INTERST RECEIPTS AND 25% ON CAPITAL GAINS.

EXAMPLE #1

RUN

DO YOU WANT INSTRUCTIONS FOR ENTERING DATA ?YES

ENTER THE FOLLOWING DATA ITEMS

ITEM 1	ANNUAL COUPON (IN DOLLARS)	?5
ITEM 2	PROCEEDS ON REDEMPTION	?100
ITEM 3 & 4	MATURITY IN YEARS, MONTHS	?6, 5.25
ITEM 5	PRICE	?97
ITEM 6 & 7	TAX RATES ON INCOME, CAP GNS	? .5, .25

COUPON	5	REDEMPTION AT	100	PRICE	97
MATURITY	6 YEARS 5.25 MONTHS				
TAX RATES	.5 ON INCOME		.25 ON CAPITAL GAINS		

YIELD BEFORE TAX 5.560054 PERCENT

YIELD AFTER TAX 2.907394 PERCENT

EXAMPLE #2

RUN

DO YOU WANT INSTRUCTIONS FOR ENTERING DATA ?NO
ENTER ITEMS 1 THROUGH 7 ?5;100;6;5.25;97;.5;.25

COUPON	5	REDEMPTION	100	PRICE	97
MATURITY	6 YEARS	5.25 MONTHS			
TAX RATES	.5 ON INCOME		.25 ON CAPITAL GAINS		
YIELD BEFORE TAX		5.560054 PERCENT			
YIELD AFTER TAX		2.907394 PERCENT			

EXAMPLE #3

RUN

DO YOU WANT INSTRUCTIONS FOR ENTERING DATA ?DATA

COUPON	5	REDEMPTION	100	PRICE	97
MATURITY	6 YEARS	5.25 MONTHS			
TAX RATES	.5 ON INCOME		.25 ON CAPITAL GAINS		
YIELD BEFORE TAX		5.560054 PERCENT			
YIELD AFTER TAX		2.907394 PERCENT			

BULL:

DESCRIPTION

In this program you will be transformed into a Matador. Have you ever been a Matador? Well, in any event you will have the opportunity to test your skill and courage in killing a bull or in the reverse being killed by a Bull!

USERS

Anyone who is looking for a good, fast, moving game where the probabilities for "loosing" or "winning" are Equal, will enjoy playing Bull. This game is extremely unique and is full of unexpected events.

INSTRUCTIONS

All instructions necessary for execution are included in the program. After the program has been loaded, type RUN; you will then be asked if you want instructions. If you type NO to this question the game will commence, any other response will yield the full set of instructions. The game is self contained and provides for all necessary inputs automatically.

LIMITATIONS

This program uses multiple line statements throughout, i.e.: lines 460, 730, 780, 970, 1030, etc. In addition, line 413 contains a Restore statement. The source code requires 5K Bytes of memory for storage and 6K Bytes for execution.

BULL

```
200 REM THIS IS A BULL FIGHT GAME
210 PRINT "DO YOU WANT INSTRUCTIONS ?"
212 INPUT Z$
214 IF Z$="NO" GOTO 400
219 PRINT
220PRINT"HERE IS YOUR BIG CHANCE TO KILL A BULL"
230PRINT
240PRINT"ON EACH PASS OF THE BULL, YOU MAY TRY"
250PRINT"0=VERONICA(DANGEROUS INSIDE MOVE OF THE CAPE),"
260PRINT"1=LESS DANGEROUS OUTSIDE MOVE OF THE CAPE, OR"
270PRINT"2=ORDINARY SWIRL OF THE CAPE"
280PRINT
290PRINT"INSTEAD OF THE ABOVE, YOU MAY TRY TO KILL THE BULL"
300PRINT"ON ANY TURN: 4=OVER THE HORNS, 5=IN THE CHEST;"
310PRINT"BUT IF I WERE YOU,"
320PRINT"I WOULDN'T TRY IT BEFORE THE SEVENTH PASS"
330PRINT
340PRINT"THE CROWD WILL DETERMINE WHAT AWARD YOU DESERVE"
350PRINT"POSTHUMOUSLY IF NECESSARY"
360PRINT"THE BRAVER YOU ARE, THE BETTER THE AWARD YOU RECEIVE"
370PRINT
380PRINT"THE BETTER A JOB THE PICADORES AND TOREADORES DO,"
390PRINT"THE BETTER YOUR CHANCES ARE"
400 PRINT
410PRINT
411 DIM Z(50)
412 REM
413 RESTORE
414 LET Z=D(3)=0
417 PRINT "*****"
420LET D(5)=1
430LET D(4)=1
440REM TYPE OF BULL
450DIM L$(5)
460 FOR X=1 TO 5\READ L$(X)\NEXT X
470 DATA SUPERB, EXCELLENT, GOOD, FAIR, STINKY
480 A=INT(RND(-1)*5)+1
490PRINT"YOU HAVE DRAWN A "; L$(A) ; " BULL"
500IF A>4 THEN 530
510IF A<2 THEN 550
520GOTO 560
530PRINT"YOU'RE LUCKY"
540GOTO 560
```



```

550PRINT"GOOD LUCK. YOU'LL NEED IT."
560REM PICADORES
570PRINT
580PRINT
590LET A$="PICADORES"
600GOSUB 1610
610LETD(1)=C
620REM TOREADORES
630LET A$="TOREADORES"
640GOSUB 1610
650LET D(2)=C
660REM PASSES
670 PRINT
680IF Z=1 THEN 1310
690LET D(3)=D(3)+1
700PRINT"PASS NUMBER "; D(3)
710IF D(3)<3 THEN 760
720PRINT"HERE COMES THE BULL! TRY FOR A KILL";
730 INPUT Z$;IF Z$="YES" THEN 1130
740PRINT "CAPE MOVE";
750GOTO800
760PRINT"THE BULL IS CHARGING AT YOU! YOU ARE THE MATADOR--"
770PRINT"DO YOU WANT TO TRY TO KILL THE BULL";
780 INPUT Z$;IF Z$="YES" THEN 1130
790PRINT"WHAT MOVE DO YOU MAKE WITH THE CAPE";
800INPUT E
810IF E<>INT(E)THEN 830
820IF E<3 THEN 850
830PRINT"DON'T PANIC, YOU IDIOT! PUT DOWN A CORRECT NUMBER";
840GOTO 800
850REM CHANCES OF BEING GORED
860IF E=0 THEN 920
870IF E=1 THEN 900
880LET M=1/2
890GOTO 930
900LET M=2
910GOTO 930
920LETM=3
930LET L=L+M
940 F=(6-A+M/10)*RND(-1)/((D(1)+D(2)+D(3)/10)*5)
950IF F<.51THEN 660
960PRINT"THE BULL HAS GORED YOU"
970D7=INT(RND(-1)*2)+1\ON D7 THEN 980,1010
980PRINT"YOU ARE DEAD"
990LET D(4)=1.5
1000GOTO 1310
1010PRINT"YOU ARE STILL ALIVE"
1020PRINT"DO YOU RUN FROM THE RING";

```

```

1030 INPUT Z$ \IF Z$="NO" THEN 1070
1040 PRINT "COWARD"
1050 LET D(4)=0
1060 GOTO 1310
1070 PRINT "YOU ARE BRAVE. STUPID, BUT BRAVE"
1080 F1=INT(RND(-1)*2)+1 \ON F1 GOTO 1090,1110
1090 LET D(4)=2
1100 GOTO 0660
1110 PRINT "YOU ARE GORED AGAIN"
1120 GOTO 970
1130 REM THE MOMENT OF TRUTH
1140 LET Z=1
1150 PRINT "IT IS THE MOMENT OF TRUTH. HOW DO YOU TRY TO KILL THE BULL?";
1160 INPUT H
1170 FOR J=4 TO 5
1180 IF H=J THEN 1230
1190 IF J=H THEN 1230
1200 NEXT J
1210 PRINT "YOU PANICKED. THE BULL GORED YOU."
1220 GOTO 970
1230 K=(6-A)*10*RND(-1)/((D(1)+D(2)*5*D(3)))
1240 IF J=4 THEN 1290
1250 IF K>.2 THEN 960
1260 PRINT "YOU KILLED THE BULL"
1270 LET D(5)=2
1280 GOTO 1310
1290 IF K>.8 THEN 960
1300 GOTO 1260
1310 REM CROWD HONORS
1320 PRINT
1330 PRINT
1340 IF D(4)<>0 THEN 1390
1350 PRINT "THE CROWD BOOS FOR TEN MINUTES. IF YOU EVER DARE TO SHOW"
1360 PRINT "YOUR FACE IN A RING AGAIN, THEY SWEAR THEY WILL KILL YOU--"
1370 PRINT "UNLESS THE BULL DOES FIRST"
1380 GOTO 1580
1390 DEF FNC(X)=((4.5+1/6-(D(1)+D(2))*2.5+4*D(4)+2*D(5)-D(3)**2)/120-A)*RND(-X)
1400 IF D(4)<>2 THEN 1430
1410 PRINT "THE CROWD CHEERS WILDLY"
1420 GOTO 1450
1430 IF D(5)<>2 THEN 1450
1440 PRINT "THE CROWD CHEERS"
1450 PRINT "THE CROWD AWARDS YOU ";
1460 D7=FNC(1) \IF D7<2.4 THEN 1570
1470 D7=FNC(1) \IF D7<4.9 THEN 1550
1480 D7=FNC(1) \IF D7<7.4 THEN 1520
1490 PRINT "BOTH EARS AND THE TAIL OF THE BULL"
1500 PRINT "OLE! YOU ARE 'MUY HOMBRE'! OLE! OLE!"

```

```

1510GOTO1580
1520PRINT"BOTH EARS OF THE BULL"
1530PRINT"OLE!"
1540GOTO 1580
1550PRINT"ONE EAR OF THE BULL"
1560GOTO 1580
1570PRINT"NOTHING"
1580PRINT
1590 PRINT "DO YOU WANT TO TRY AGAIN";
1591 INPUT Z$;IF Z$="YES" GOTO 410
1600GOTO2030
1610 LET B=3/A*AND(-1)
1620IF B<.37 THEN 1740
1630IF B<1/2 THEN 1720
1640IF B<.63THEN 1700
1650IF B<.87 THEN 1680
1660LET C=1/10
1670GOTO 1750
1680LET C=2/10
1690GOTO1750
1700LET C=3/10
1710GOTO1750
1720LET C =4/10
1730 GOTO 1750
1740LET C=1/2
1750LET T=INT(10*C+2/10)
1760PRINT"THE "; A$; " DID A "; L$(T) ; " JOB"
1770IF 4>T THEN 1900
1780IF 5=T THEN 1870
1790 F1=INT(AND(-1)*2)+1\ON F1 GOTO 1800,1900
1800IF A$="TOREADORES" THEN 1820
1810PRINT"ONE OF THE HORSES OF THE ";A$;" WAS GORED"
1820 F1=INT(AND(-1)*2)+1\ON F1 GOTO 1830,1850
1830PRINT"ONE OF THE ";A$;" WAS KILLED"
1840GO TO 1900
1850PRINT "NO ";A$; " WERE KILLED"
1860GOTO 1900
1870IF A$="TOREADORES" THEN 1890
1880 F1=INT(AND(-1)*2)+1\PRINT F1;" OF THE HORSES OF THE ";A$;" GORED"
1890 F1=INT(AND(-1)*2)+1\PRINT F1;" OF THE ";A$;" KILLED"
1900PRINT
1910RETURN
2030 END

```

DO YOU WANT INSTRUCTIONS?YES

HERE IS YOUR BIG CHANCE TO KILL A BULL

ON EACH PASS OF THE BULL, YOU MAY TRY
0=VERONICA(DANGEROUS INSIDE MOVE OF THE CAPE),
1=LESS DANGEROUS OUTSIDE MOVE OF THE CAPE, OR
2=ORDINARY SWIRL OF THE CAPE

INSTEAD OF THE ABOVE, YOU MAY TRY TO KILL THE BULL
ON ANY TURN: 4=OVER THE HORNS, 5=IN THE CHEST;
BUT IF I WERE YOU,
I WOULDN'T TRY IT BEFORE THE SEVENTH PASS

THE CROWD WILL DETERMINE WHAT AWARD YOU DESERVE
POSTHUMOUSLY IF NECESSARY
THE BRAVER YOU ARE, THE BETTER THE AWARD YOU RECEIVE

THE BETTER A JOB THE PICADORES AND TOREADORES DO,
THE BETTER YOUR CHANCES ARE

YOU HAVE DRAWN A GOOD BULL

THE PICADORES DID A EXCELLENT JOB

THE TOREADORES DID A EXCELLENT JOB

PASS NUMBER 1

THE BULL IS CHARGING AT YOU! YOU ARE THE MATADOR--

DO YOU WANT TO TRY TO KILL THE BULL?YES

IT IS THE MOMENT OF TRUTH. HOW DO YOU TRY TO KILL THE BULL?1

YOU PANICKED. THE BULL GORED YOU.

YOU ARE DEAD

THE CROWD AWARDS YOU NOTHING

DO YOU WANT TO TRY AGAIN?NO

ENTERPRISE:

DESCRIPTION

The federation is in dire need of a commander to takeover the Enterprise while Captain Kirk is on leave. Prior military experience helpful as the Enterprise and a Klingon battle cruiser face each other in a simulated battle mission on your computer. Designed to test your abilities, this version of the game will require considerable skill on your part to avoid total destruction. GOOD LUCK!

USERS

Starfleet Captains and other individuals interested in keeping the Star Trek tradition alive and well, as well as having a lot of fun, will find this simulation space battle both exciting and fascinating. ****WARNING**** This game may be habit forming — players have been known to become addicted!

INSTRUCTIONS

Type RUN after the program is loaded and the program will ask you if you want instructions. The program is self instructing and easy to use.

LIMITATIONS

Line 230 contains a TAB () statement, 670 an ON_GOTO statement and 2270 a Restore statement. Lines 330 and 2810 contain Multiple line statements. The source code is 7K Bytes long and the program will execute in 8K Bytes of memory. For those of you experiencing extreme difficulty in bringing back the Enterprise in one piece, the weapon control constraints are located in lines: 700, 800, 870, 2010 and 2020.

ENTERPRISE

```
0005 REM THIS IS ENTERPRISE
0010 S4=1
0020 DIM I(25)
0030 PRINT " DO YOU WANT INSTRUCTIONS (YES OR NO) ";
0040 INPUT A$
0050 IF A$="NO" THEN 350
0060 IF A$="No" THEN 350
0070 Q=67
0080 PRINT "WHILE ON PATROL NEAR TRISKELLION-7..."
0090 PRINT "THE USS ENTERPRISE IS CONFRONTED BY "
0100 PRINT "A K L I N G O N B A T T L E C R U I S E R"
0110 REM
0120 PRINT TAB(20);" WEAPONRY"
0130 REM
0140 PRINT "2 FORWARD PHASER BANKS; RANGE: 300,000 KM."
0150 PRINT "1 REAR PHASER; RANGE: SAME"
0160 REM
0170 REM
0180 PRINT "2 FORWARD PHOTON TORPEDO BANKS; RANGE: 20,000 TO 600,000 KM."
0190 REM
0200 PRINT "1 REAR PHOTON TORPEDO BANK; RANGE: SAME"
0210 REM
0220 REM
0230 PRINT TAB(15);"M A N E U V E R S "
0240 REM
0250 PRINT "1-FIRE FORWARD PHASERS"
0260 PRINT "2-FIRE REAR PHASER"
0270 PRINT "3-FIRE FORWARD PHOTON TORPEDO"
0280 PRINT "4-FIRE REAR PHOTON TORPEDO"
0290 PRINT "5-ACTIVATE AUTO DISTRUCT"
0300 PRINT "6-SURRENDER"
0310 PRINT "7-ATTEMPT TO BREAK CONTACT--GO INTO WARP DRIVE"
0320 PRINT "8-MOVE CLOSER TO THE KLINGON"
0330 PRINT\PRINT "ALL VECTOR HEADINGS WILL BE BETWEEN 0 AND 180 DEGREES"
0340 REM
0350 PRINT "KLINGON COMING INTO RANGE--SHIELDS ON"
0360 READ R,H,P,X,A
0370 DATA 85000.,0,0,-1,97.3304
0380 REM
0390 REM
0400 PRINT "WHAT IS YOUR COMPUTER DEFENSE NUMBER (1-25) ";
0410 INPUT N2
0420 IF N2<1 GOTO 400
```

```

0430 IF N2>25 GOTO 400
0440 REM
0450 REM
0460 FOR I=1 TO N2
0470 B=RND(X)
0480 NEXT I
0490 GOSUB 1370
0500 PRINT "THE RANGE=";R;" KM. AT A VECTOR HEADING OF ";A;" DEGREES"
0510 GOSUB 620
0520 IF H>8 THEN 2560
0530 IF E=6 GOTO 2700
0540 IF E=5 GOTO 2610
0550 IF R>1.00000E+6 THEN 2690
0560 GOSUB 1100
0570 IF P>8 GOTO 2610
0580 IF K=7 GO TO 500
0590 GOTO 490
0600 REM
0610 REM
0620 W=2
0630 PRINT "WHAT IS YOUR MOVE";
0640 INPUT E
0650 GOSUB 1490
0660 IF Q>0 THEN 630
0670 ON E GOTO 680,790,810,850,900,940,980,1030
0680 IF A>90 THEN 730
0690 B=RND(X)
0700 IF B>.2 GOTO 770
0705 REM THIS IS FOR THE FRONT PHASERS
0710 PRINT "COMPLETE MISS, YOUR AIM IS OFF"
0720 RETURN
0730 PRINT "MISS, INCORRECT VECTOR ANGLE"
0740 PRINT "REMEMBER THAT BETWEEN 0 AND 90 DEGREES IS FORWARD"
0750 PRINT " AND THAT BETWEEN 90 AND 180 DEGREES IS REARWARD"
0760 RETURN
0770 GOSUB 2000
0780 RETURN
0790 IF A<90 THEN 730
0800 IF B>.4 GOTO 770
0803 REM THIS IS FOR THE REAR PHASERS
0805 GOTO 710
0810 IF A>90 THEN 730
0820 B=RND(X)
0830 IF B>.1 GOTO 770
0835 REM THIS IS FOR THE FRONT PHOTON TORPEDOES
0840 GOTO 710
0850 IF A<90 THEN 730
0860 B=RND(X)

```

```

0870 IF B>.2 GOTO 770
0875 REM THIS IS FOR THE REAR PHOTON TORPEDOES
0880 GOTO 710
0890 REM
0900 PRINT TAB(20);"A C T I V A T E D"
0910 REM
0920 PRINT "10   9   8   7   6   5   4   3   2   1   0"
0930 RETURN
0940 PRINT "ON BEHALF OF THE KLINGON EMPIRE, I ACCEPT YOUR SURRENDER"
0950 REM
0960 PRINT TAB(20);"PREPARE TO BE BOARDED"
0970 RETURN
0980 R=R+(200000.*RND(X))
0990 IF R<1.00000E+06 THEN 1010
1000 RETURN
1010 PRINT "CONTACT NOT BROKEN, NEW DISTANCE=";R;" KM."
1020 RETURN
1030 IF R<200000. THEN 1060
1040 R=R-200000.*RND(X)
1050 RETURN
1060 R=R/2
1070 RETURN
1080 REM
1090 REM
1100 W=1
1110 K=INT((7*RND(X))+1)
1120 GOSUB 1490
1130 IF N>0 THEN 1110
1140 ON K GOTO 1150,1150,1180,1180,1180,1230,1290
1150 PRINT "KLINGON FIRES PHOTON TORPEDO"
1160 GOSUB 2000
1170 RETURN
1180 PRINT "KLINGON FIRES PHASERS AT ENTERPRISE"
1190 LET B=RND(X)
1200 IF B>.2 THEN 1160
1210 PRINT "YOU OUTMANEUVERED HIM, MISS."
1220 RETURN
1230 PRINT "KLINGON ATTEMPTING TO BREAK CONTACT....";
1240 R=R+(200000.*RND(X))
1250 IF R<1.00000E+06 THEN 1270
1260 RETURN
1270 PRINT "CONTACT NOT BROKEN"
1280 RETURN
1290 PRINT "KLINGON APPROACHING"
1300 IF R<200000. THEN 1330
1310 R=R-200000.*RND(X)
1320 RETURN
1330 R=R/2

```



```

1340 RETURN
1350 REM
1360 REM
1370 B=RND(X)
1380 IF B>.5 THEN 1410
1390 R=R+(10000*RND(X))
1400 GOTO 1420
1410 R=ABS(R-(10000*RND(X)))
1420 A=A+(100*RND(X))
1430 IF A>180 THEN 1450
1440 RETURN
1450 A=A-180
1460 GOTO 1430
1470 REM
1480 REM
1490 IF W=1 THEN 1750
1500 IF E<5 THEN 1530
1510 Q=0
1520 RETURN
1530 IF E<3 THEN 1700
1540 IF R>600000. THEN 1590
1550 IF D=5 THEN 1650
1560 IF R<20000. THEN 1590
1570 Q=0
1580 RETURN
1590 PRINT "THE RANGE OF A PHOTON TORPEDO IS 20,000 TO 600,000 KM."
1600 GOTO 1670
1610 PRINT "YOUR PHASERS HAVE BEEN DESTROYED ALREADY"
1620 GOTO 1670
1630 PRINT "THE RANGE OF A PHASER IS 300,000 KM."
1640 GOTO 1670
1650 PRINT "YOUR WEAPONS HAVE BEEN DESTROYED ALREADY"
1660 GOTO 1670
1670 PRINT "MOVE IMPOSSIBLE, TRY AGAIN"
1680 Q=1
1690 RETURN
1700 IF R>300000. THEN 1630
1710 IF D=4 THEN 1610
1720 IF D=5 THEN 1650
1730 Q=0
1740 RETURN
1750 IF K>5 THEN 1880
1760 IF K>2 THEN 1840
1770 IF R>600000. THEN 1820
1780 IF R<20000. THEN 1820
1790 IF H>7 THEN 1820
1800 N=0
1810 RETURN

```

```

1820 N=1
1830 RETURN
1840 IF R>300000. THEN 1820
1850 IF H>6 THEN 1820
1860 N=0
1870 RETURN
1880 IF P>H+1 THEN 1950
1890 IF P<H-1 THEN 1920
1900 N=0
1910 RETURN
1920 IF K=7 THEN 1820
1930 N=0
1940 RETURN
1950 IF K=6 THEN 1820
1960 N=0
1970 RETURN
1980 REM
1990 REM
2000 B=RND(X)
2010 IF B>.75 THEN 2120
2020 IF B>.1 GOTO 2050
2030 PRINT "NEAR MISS"
2040 RETURN
2050 IF W=1 THEN 2085
2060 PRINT "HIT ON KLINGON BATTLE CRUISER"
2070 H=H+1
2080 GOTO 2380
2085 IF B<.4 GOTO 2030
2090 PRINT "HIT ON THE USS ENTERPRISE, DAMAGE REPORT-"
2100 P=P+1
2110 GOTO 2180
2120 IF W=1 THEN 2160
2130 PRINT "DIRECT HIT ON KLINGON VESSEL, DAMAGE REPORT-"
2140 H=H+2
2150 GOTO 2380
2160 PRINT "DIRECT HIT ON ENTERPRISE, DAMAGE REPORT-"
2170 P=P+2
2180 IF P<9 THEN 2210
2190 D=1
2200 RETURN
2210 IF P=6 THEN 2290
2220 IF P=7 THEN 2310
2230 IF P=8 THEN 2340
2240 IF P>3 THEN 2270
2250 PRINT "SHIELDS HOLDING - NO DAMAGE"
2260 RETURN
2270 PRINT "SHIELDS WEAKENING - MINOR DAMAGE IN MIDSHIPS"
2280 RETURN
2290 PRINT "ALL SHIELDS DESTROYED - DAMAGE TO OUTER HULL"
2300 RETURN
2310 D=4

```

```

2320 PRINT "PHASERS DEACTIVATED - DILITHIUM CRYSTALS OVERHEATING"
2330 RETURN
2340 D=5
2350 PRINT "ALL WEAPONS DESTROYED - POWER DROPPING"
2360 PRINT "MAJOR DAMAGE TO WARP ENGINES"
2370 RETURN
2380 IF H<9 THEN 2410
2390 D=8
2400 RETURN
2410 IF H=8 THEN 2530
2420 IF H=7 THEN 2510
2430 IF H=6 THEN 2490
2440 IF H>3 THEN 2470
2450 PRINT "SHIELDS HOLDING - NO DAMAGE"
2460 RETURN
2470 PRINT "SHIELDS WEAKENING - MINOR DAMAGE"
2480 RETURN
2490 PRINT "ALL SHIELDS DESTROYED - HULL DAMAGE"
2500 RETURN
2510 PRINT "PHASER BANKS DESTROYED - MAJOR DAMAGE FORE AND AFT"
2520 RETURN
2530 D=7
2540 PRINT "ALL WEAPONS ON KLINGON VESSEL DEACTIVATED, POWER DROPPING"
2550 RETURN
2560 PRINT TAB (15);"C O N G R A T U L A T I O N S"
2570 PRINT
2580 PRINT "KLINGON VESSEL DESTROYED"
2590 PRINT
2600 GOTO 2700
2610 PRINT TAB (20);"ENTERPRISE DESTROYED"
2620 PRINT
2630 IF E=5 THEN 2650
2640 D=1000*RND(X)\GOTO 2660
2650 D=100000.*RND(X)
2660 PRINT "RADIUS OF MATTER-ANTIMATTER EXPLOSION=";D;" KM."
2670 IF R<D THEN 2580
2680 GOTO 2700
2690 PRINT "CONTACT BROKEN, RETURN TO BASE"
2700 PRINT "AGAIN (YES OR NO)";\GOTO 2760
2710 PRINT "STAR FLEET HAS DETERMINED THAT YOU ARE TOO BLOOD-THIRSTY"
2720 PRINT "THEREFORE, IT HAS BEEN DECIDED THAT YOU WILL NOT BE "
2730 PRINT "ALLOWED TO CONTINUE ON THIS RAMPAGE OF KILLING POOR"
2740 PRINT "LITTLE DEFENSELESS KLINGONS.ADMIRAL WESTLAKE,"
2750 PRINT "                STAR FLEET COMMAND"\GOTO 2820
2760 INPUT A$
2770 RESTORE
2780 IF S4>=4 THEN 2710
2790 IF A$="YES" THEN 2810
2800 IF A$<>"YES" THEN 2820
2810 S4=S4+1\GOTO 350
2820 PRINT ""
2830 END

```

DO YOU WANT INSTRUCTIONS (YES OR NO)?YES
WHILE ON PATROL NEAR TRISKELLION-7...
THE USS ENTERPRISE IS CONFRONTED BY
A KLINGON BATTLE CRUISER
WEAPONRY

2 FORWARD PHASER BANKS, RANGE: 300,000 KM.
1 REAR PHASER, RANGE: SAME
2 FORWARD PHOTON TORPEDO BANKS, RANGE: 20,000 TO 600,000 KM.
1 REAR PHOTON TORPEDO BANK, RANGE: SAME

MANEUVERS

1-FIRE FORWARD PHASERS
2-FIRE REAR PHASER
3-FIRE FORWARD PHOTON TORPEDO
4-FIRE REAR PHOTON TORPEDO
5-ACTIVATE AUTO DISTRUCT
6-SURRENDER
7-ATTEMPT TO BREAK CONTACT--GO INTO WARP DRIVE
8-MOVE CLOSER TO THE KLINGON

ALL VECTOR HEADINGS WILL BE BETWEEN 0 AND 180 DEGREES
KLINGON COMING INTO RANGE--SHIELDS ON
THE RANGE= 91332.36 KM. AT A VECTOR HEADING OF 144.9378 DEGREES
WHAT IS YOUR MOVE?4
COMPLETE MISS, YOUR AIM IS OFF
KLINGON APPROACHING
THE RANGE= 45666.18 KM. AT A VECTOR HEADING OF 144.9378 DEGREES
WHAT IS YOUR MOVE?4
COMPLETE MISS, YOUR AIM IS OFF
KLINGON FIRES PHASERS AT ENTERPRISE
NEAR MISS
THE RANGE= 53897.81 KM. AT A VECTOR HEADING OF 62.60793 DEGREES
WHAT IS YOUR MOVE?3
DIRECT HIT ON KLINGON VESSEL, DAMAGE REPORT-
SHIELDS HOLDING - NO DAMAGE
KLINGON FIRES PHOTON TORPEDO
NEAR MISS
THE RANGE= 63323.14 KM. AT A VECTOR HEADING OF 100.1475 DEGREES
WHAT IS YOUR MOVE?4
COMPLETE MISS, YOUR AIM IS OFF
KLINGON FIRES PHASERS AT ENTERPRISE
HIT ON THE USS ENTERPRISE, DAMAGE REPORT-
SHIELDS HOLDING - NO DAMAGE
THE RANGE= 61206.09 KM. AT A VECTOR HEADING OF 109.9991 DEGREES
WHAT IS YOUR MOVE?2
DIRECT HIT ON KLINGON VESSEL, DAMAGE REPORT-
SHIELDS WEAKENING - MINOR DAMAGE
KLINGON FIRES PHASERS AT ENTERPRISE
HIT ON THE USS ENTERPRISE, DAMAGE REPORT-
SHIELDS HOLDING - NO DAMAGE
THE RANGE= 57123.84 KM. AT A VECTOR HEADING OF 154.9717 DEGREES
WHAT IS YOUR MOVE?

FOOTBALL:

DESCRIPTION

If you're a football fan this program should be right up your 50 yard line. This is an authentic football game simulation. The game adheres to the NFL rules and the play generated is as stimulating as you're up too.

USERS

NFL and AFL fans will find this game very enjoyable. The game will appeal to anyone who has an interest in football, regardless of their expertise.

INSTRUCTIONS

The program contains all necessary prompts and instructions for playing the game. To use -- load the program and then type RUN. For detailed program operations list the program before playing.

LIMITATIONS

This program uses the DEF FNX () function and the FNX () function extensively, starting in program line 1090. Line 2030 contains an ABS () statement and line 6080 contains a Restore statement. The source code requires 11K Bytes for storage and 13K Bytes of memory for storage and execution.

FOOTBALL

```
100 REM THIS IS FOOTBALL
130 REM THIS PROGRAM SIMULATES A FOOTBALL GAME
140 REM THE USER CHOOSES HIS OWN TEAM AND HIS OPPONENT AND THEN
150 REM ACTS AS THE QUARTERBACK AND DEFENSIVE SIGNAL-CALLER FOR HIS TEAM
154 REM THIS PROGRAM SAMPLES THE OPPOSITIONS PLAYS FROM A NON-LINEAR DIST.
210 PRINT"      THIS IS PRO-STYLE FOOTBALL. YOU WILL BE THE"
220 PRINT"QUARTERBACK AND DEFENSIVE SIGNAL-CALLER FOR YOUR"
230 PRINT"TEAM. YOU HAVE 14 OFFENSIVE PLAYS (8 RUNS; 6 PASSES)"
240 PRINT"AND 4 DEFENSIVE ALIGNMENTS."
250 PRINT"      THERE IS A TWO-MINUTE WARNING BEFORE THE END"
260 PRINT"OF EACH HALF. FIVE TO SEVEN PLAYS ARE LEFT IN THE"
270 PRINT"HALF AT THAT POINT."
280 PRINT
350 REM
360 DIM A$(5)
370 DIM B(15), D$(10)
390 FOR I=1 TO 5
400 READ A$
410 LET D$(I)=A$
420 NEXT I
430 DATA POINTS, FIRST DOWNS, YDS-RUSH, YDS-PASS, YDS-PENALTY
440 PRINT
471 IF I8=1 GOTO 630
472 PRINT"DO YOU WANT A LIST OF PLAYS (YES OR NO)";
474 INPUT I9$
476 IF I9$="NO" GOTO 630
477 IF I9$="N" GOTO 630
480 PRINT"CALL PLAYS AS FOLLOWS:"
490 PRINT
500 PRINT"RUNS:"
510 PRINT"      1=DIVE; 2=OFF TACKLE; 3=SCISSORS; 4=TRAP"
520 PRINT"      5=SWEEP; 6=OPTION; 7=REVERSE; 8=DRAW"
530 PRINT
540 PRINT"PASSES:"
550 PRINT"      9=SIDELINE; 10=LOOK-IN; 11=ROLLOUT"
560 PRINT"      12=SCREEN; 13='FLY'; 14='POST'"
570 PRINT
580 PRINT"KICKS:"
590 PRINT"      15=FIELD GOAL; 16=PUNT; 17=QUICK KICK"
600 PRINT
610 PRINT"CALL DEFENSES AS FOLLOWS:"
620 PRINT"      1='PRO'(4-3); 2='OKIE'(5-2); 3='SHORT-YDG'; 4='PREVENT'"
630 PRINT
680 PRINT"YOUR TEAM";
```

```

690 INPUT A$
700 LET O$(1)=A$
710 PRINT
720 PRINT"OPPONENT";
730 INPUT B$
740 PRINT
750 LET O$(3)=B$
760 IF B$(<>)A$ THEN 800
770 PRINT"YOU ARE "A$";". TRY AGAIN."
780 GOTO 730
800 REM
820 DIM H(69),U(69),E$(50),L(15),G(15)
840 FOR I=1 TO 45
850 READ D$
860 LET E$(I)=D$
870 NEXT I
880 DATA YARD,KICKOFF,TOUCHBACK,RUNBACK,BALL ON
890 DATA YARDLINE,DIVE,OFF TACKLE,SCISSORS,TRAP
900 DATA SWEEP,OPTION,REVERSE,DRAW,SIDELINE
910 DATA LOOK-IN,ROLLOUT,SCREEN,FLY,POST
920 DATA FIELD GOAL,PUNT,QUICK KICK,EXTRA POINT
930 DATA PRO,OKIE,GOAL-LINE,PREVENT,PENALTY
940 DATA OFFSIDE,HOLDING,INCOMPLETE,BATTED DOWN
950 DATA INTERCEPTION,LOSS,NO GAIN,GAIN,FUMBLE
960 DATA TOUCHDOWN,SAFETY,GOAL TO GO
970 DATA FIRST,SECOND,THIRD,FOURTH
980 FOR I=1 TO 14
990 READ L(I),G(I)
1000 NEXT I
1010 DATA 2,7,3,10,5,15,5,20,3,10,7,15,9,25
1020 DATA 5,15,5,10,5,10,7,15,10,20,10,50,10,40
1030 H9=25
1040 E9=50
1090 DEF FNF(X)=SIN(3.141*RND(-X))
1100 T5=0=1-2*INT(RND(-1)*2)
1110 IF Q>0 THEN 1150
1120 PRINT O$(3);" WON THE TOSS AND WILL RECEIVE."
1130 PRINT
1140 GOTO 1200
1150 PRINT O$(1);" WON THE TOSS."
1160 PRINT"DO YOU ELECT TO KICK OR RECEIVE";
1170 INPUT C$
1180 PRINT
1190 IF C$="KICK" THEN 1220
1200 Q=-1*Q
1210 IF Z9>0 THEN 5990
1220 X=50-(10*Q)
1230 Y=30+INT(40*FNF(1))
1240 PRINT Y;E$(1);" "E$(2)
1250 X=X+(Q*Y)
1260 IF Q=1 THEN 1290

```

```

1270 IF X>0 THEN 1350
1280 GOTO 1300
1290 IF X<100 THEN 1350
1300 PRINT E$(3)
1310 D=0
1320 X=50+(30*Q)
1330 Q=-1*Q
1340 GOTO 1400
1350 IF P>14 THEN 1370
1360 R=INT(40*FNF(1))
1370 Q=-1*Q
1380 PRINT R; E$(1); " ";E$(4)
1390 X=X+(Q*R)
1400 R=P1=0
1410 GOSUB 1450
1420 GOTO 1550
1450 PRINT"BALL ON ";
1460 IF X>50 THEN 1490
1470 PRINT Q$(1); " ";X;
1480 GOTO 1500
1490 PRINT Q$(3); " ";ABS(X-100);
1500 PRINT " YARD LINE"
1510 RETURN
1550 D=D+1
1560 IF D>1 THEN 1590
1570 DEF FNT(X,Q)=ABS(X-(((Q+1)/2)*100))
1580 DEF FNU(X,Q)=ABS(X-(((Q-1)/2)*100))
1590 IF D<>1 THEN 1660
1600 IF P8<>0 THEN 1620
1610 S=0
1620 PRINT E$(42); " DOWN ";Q$(ABS(Q-2));
1630 IF FNT(X,Q)>10 THEN 1680
1640 G1=1
1650 GOTO 1710
1660 PRINT E$(41+D); " DOWN";
1670 IF G1>0 THEN 1710
1680 PRINT " YARDS TO GO:";ABS(10-S)
1690 PRINT
1700 GOTO 1730
1710 PRINT " ";E$(41)
1720 PRINT
1730 REM
1780 P8=0
1790 P1=P
1800 GOSUB 5490
1810 IF Z>0 THEN 5390
1820 IF Q>0 THEN 2260
1830 GOTO 2940
1840 IF D>1 THEN 1890
1870 F=1+INT(3.5*RND(-1))*4
1880 GOTO 2330

```



```

1890 IF D>2 THEN 2000
1920 IF ABS(10-S)>3 THEN 1950
1930 P=12+INT(3*RND(-1))
1940 GOTO 2330
1950 IF P1>12 THEN 1870
1960 IF P1>8 THEN 1990
1970 P=P1+1
1980 GOTO 2330
1990 ON (INT(2*RND(-1))+1) GOTO 1870,1970
2000 IF D>3 THEN 2160
2030 IF ABS(10-S)>3 THEN 2060
2040 P=1+4*INT(2*RND(-1))+INT(2*RND(-1))
2050 GOTO 2330
2060 IF ABS(10-S)>6 THEN 2080
2070 GOTO 1970
2080 IF ABS(10-S)>11 THEN 2110
2090 P=P1+1
2100 GOTO 1970
2110 IF ABS(10-S)<25 THEN 1930
2120 P=17
2130 GOTO 2330
2160 IF FNT(X,0)>40 THEN 2220
2170 IF ABS(10-S)>3 THEN 2190
2180 GOTO 2040
2190 IF FNT(X,0)>25 THEN 2220
2200 P=15
2210 GOTO 2330
2220 P=16
2230 GOTO 2330
2260 PRINT"YOUR PLAY";
2270 INPUT P
2280 IF P>17 GOTO 2286
2282 IF P<1 GOTO 2286
2284 GOTO 2330
2286 PRINT"WHAT KINDA PLAY'S THAT? TRY AGAIN, TWINKLETOES."
2288 GOTO 2260
2330 PRINT E$(P+6)
2340 IF P<15 THEN 2360
2350 ON P-14 GOTO 2410,2660,2730
2360 L=L(P)
2370 G=G(P)
2380 IF Q<0 THEN 3010
2390 GOTO 2840
2410 REM
2420 Y=INT(FNF(1)*50)
2430 IF Y>.6*FNT(X,0) THEN 2460
2440 PRINT E$(21);" IS BLOCKED"
2450 GOTO 2520
2460 IF Y<(FNT(X,0)+10) THEN 2510
2470 IF .05+10/FNT(X,0)<RND(-1) THEN 2510
2480 PRINT E$(21);" IS GOOD!"

```

```

2490 P9=3
2500 GOTO 5200
2510 PRINT E$(21);" FAILED"
2520 D=S=0
2530 IF Y>FNT(X,0) THEN 1300
2540 X=X+(0*Y)
2550 R=INT(FNF(1)*Y)
2560 GOTO 2780
2590 IF RND(-1)>.20 THEN 2630
2600 PRINT E$(24);" FAILED"
2610 P9=0
2620 GOTO 5390
2630 PRINT E$(24);" IS GOOD!"
2640 GOTO 5190
2660 REM
2670 Y=20+INT(30*FNF(1))
2680 PRINT Y;" YARD PUNT"
2690 X=X+(0*Y)
2700 R=INT(35*FNF(1))
2710 GOTO 2780
2730 REM
2740 Y=30+INT(20*FNF(1))
2750 PRINT Y;" YARD KICK"
2760 X=X+(0*Y)
2770 R=INT(10*FNF(1))
2780 GOTO 3960
2840 IF Q<0 THEN 2940
2850 IF FNT(X,0)>20 THEN 2880
2860 D9=3
2870 GOTO 3010
2880 IF ABS(10-S)<20 THEN 2910
2890 D9=4
2900 GOTO 3010
2910 D9=1+INT(2*RND(-1))
2920 GOTO 3010
2940 PRINT"DEFENSE";
2950 INPUT D9
2960 IF D9>4 GOTO 2966
2962 IF D9<1 GOTO 2966
2964 GOTO 1840
2966 PRINT"YA PRIMADONNA! CALL A REAL DEFENSE, OKAY?"
2968 GOTO 2940
3010 REM
3020 IF P>8 THEN 3230
3030 IF P>4 THEN 3050
3040 ON P GOTO 3230,3080,3120,3120
3050 ON P-4 GOTO 3080,3160,3160,3200
3080 IF P1<>1 THEN 3230
3090 GOTO 3210
3120 IF P1<>2 THEN 3230
3130 GOTO 3210

```

```

3160 IF P1<>5 THEN 3230
3170 GOTO 3210
3200 IF P1<9 THEN 3230
3210 U=1.25
3220 GOTO 3260
3230 U=1
3260 IF D9<>1 THEN 3290
3270 IF P>5 THEN 3410
3280 ON P GOTO 3430,3410,3390,3410,3430
3290 IF D9<>2 THEN 3320
3300 IF P>5 THEN 3410
3310 ON P GOTO 3410,3430,3410,3390,3430
3320 IF D9<>3 THEN 3360
3330 IF P>5 THEN 3350
3340 ON P GOTO 3430,3430,3410,3410,3430
3350 IF P<>9 THEN 3410
3360 IF P<13 THEN 3380
3370 GOTO 3430
3380 IF P<>8 THEN 3410
3390 M=1.25
3400 GOTO 3480
3410 M=1
3420 GOTO 3480
3430 M=.8
3480 Y=INT(((G*RND(-1))-(L*RND(-1)))*(U*M))
3490 IF R2<>0 THEN 5130
3500 IF RND(-1)<.98 THEN 3530
3510 Y=FNT(X,0)+1
3520 GOTO 4570
3530 IF P>8 THEN 3790
3580 IF RND(-1)<.93 THEN 4030
3590 Y=INT(.5*FNF(1)*Y)
3600 PRINT E$(38);" AFTER ";
3610 IF Y<=0 THEN 3640
3620 PRINT Y;" ";E$(1);" ";E$(37)
3630 GOTO 3690
3640 IF Y<>0 THEN 3670
3650 PRINT E$(36)
3660 GOTO 3690
3670 PRINT ABS(Y);" ";E$(1);" ";E$(35)
3690 IF 1-INT(2*RND(-1))=0 THEN 3720
3700 PRINT E$(38);" RECOVERED"
3710 GOTO 4030
3720 PRINT E$(38);" LOST"
3730 X=X+(0*Y)
3740 Q=Q*(-1)
3750 GOTO 5320
3790 IF RND(-1)<.07 THEN 3930
3800 IF Y>0 THEN 3830
3810 PRINT"QB TACKLED FOR ";
3820 GOTO 4570

```

```

3830 IF ABS(RND(-1)-.5)<.23 THEN 4570
3840 IF INT(RND(-1)*2)=0 THEN 3890
3860 PRINT E$(33)
3870 GOTO 3900
3890 PRINT E$(32)
3900 Y=0
3910 GOTO 4030
3930 PRINT E$(34)
3940 X=X+(Q*Y)
3950 R=INT(15*FNF(1))
3960 P=18
3970 D=S=G1=0
3980 GOTO 4570
4030 IF ABS(RND(-1)-.5)>.04 THEN 4570
4040 P8=1
4050 GOTO 4710
4060 IF INT(2*RND(-1))=0 THEN 4130
4070 W=-1
4080 IF INT(2*RND(-1))=0 THEN 4110
4090 P8=15
4100 GOTO 4150
4110 P8=5
4120 GOTO 4170
4130 W=1
4140 GOTO 4080
4150 PRINT E$(31);
4160 GOTO 4180
4170 PRINT E$(30);
4180 PRINT " ";E$(29);" ON ";Q$(ABS(W-2))
4190 IF W<0 THEN 4350
4200 IF Q<0 THEN 4300
4210 IF P8=15 THEN 4230
4220 IF Y<=0 THEN 4320
4230 GOSUB 4390
4240 Y=P8*W
4250 PRINT E$(29);" ACCEPTED: ";P8;" YARDS"
4260 D=D-1
4270 X=X+(Q*Y)
4280 B(14-W)=B(14-W)+P8
4290 GOTO 5280
4300 IF S+Y>ABS(10-S) THEN 4320
4310 GOTO 4230
4320 P8=0
4330 GOTO 4570
4340 IF W>0 THEN 4210
4350 PRINT"DO YOU ACCEPT THE PENALTY (YES OR NO)";
4360 INPUT C$
4370 IF C$="NO" THEN 4320
4375 IF C$="N" THEN 4320
4380 GOTO 4230
4390 IF Q>0 THEN 4470

```

```

4400 IF M) 1 THEN 4440
4410 IF P P8<FNU(X;0) THEN 4520
4420 P8=INT(FNU(X;0)/2)
4430 GOTO 4520
4440 IF 2*P8<FNT(X;0) THEN 4520
4450 P8=INT(FNT(X;0)/2)
4460 GOTO 4520
4470 IF M>0 THEN 4500
4480 M=1
4490 GOTO 4440
4500 M=-1
4510 GOTO 4410
4520 RETURN
4570 IF P>14 THEN 4680
4580 X=X+(Q*Y)
4590 GOSUB 5720
4600 IF Q>0 THEN 4640
4610 IF X<1 THEN 4940
4620 IF X>99 THEN 4850
4630 GOTO 4680
4640 IF X>99 THEN 4940
4650 IF X<1 THEN 4850
4680 IF P>14 THEN 5350
4690 IF R=0 THEN 4710
4700 GOTO 1380
4710 IF Y<=0 THEN 4740
4720 PRINT Y;" YARD GAIN"
4730 GOTO 4780
4740 IF Y<>0 THEN 4770
4750 PRINT E$(36)
4760 GOTO 4780
4770 PRINT ABS(Y);" YARD LOSS"
4780 IF P8<>0 THEN 4060
4790 GOTO 5280
4850 PRINT E$(40);" US "$0$(ABS(Q-2))
4860 P9=2
4870 Q=-1*Q
4880 S8=1
4890 GOSUB 5660
4900 Q=-1*Q
4910 GOTO 5380
4940 PRINT E$(39);" "$0$(ABS(Q-2));"!!!"
4950 P9=6
4960 GOSUB 5660
4970 IF Q>0 THEN 5030
4980 IF S2+1=S1 THEN 2590
4990 IF S2+2=S1 THEN 5010
5000 GOTO 2590
5010 P=4
5020 GOTO 5100
5030 PRINT"KICK EXTRA POINT (YES OR NO)";

```

```

5040 INPUT C$
5050 IF C$="YES" THEN 2590
5055 IF C$="Y" THEN 2590
5060 PRINT"PLAY";
5070 INPUT P
5080 IF P>14 THEN 5060
5090 IF P>12 THEN 5220
5100 R2=1
5110 D9=2
5120 GOTO 3020
5130 R2=0
5140 IF Y<4.1 THEN 5220
5150 P9=2
5160 PRINT E$(24);"S ARE GOOD!"
5170 GOSUB 5660
5180 GOTO 5380
5190 P9=1
5200 GOSUB 5660
5210 GOTO 5380
5220 PRINT E$(24);" ATTEMPT FAILED"
5230 GOTO 5380
5280 S=S+Y
5290 IF S>9 THEN 5320
5300 IF D<4 THEN 5350
5310 Q=Q*(-1)
5320 D=S=G1=0
5330 IF P>14 THEN 5350
5340 B(5+Q)=B(5+Q)+1
5350 IF P=18 THEN 1260
5360 GOSUB 1450
5370 GOTO 1550
5380 PRINT
5390 PRINT Q$(1);": ";S1;"      ";Q$(3);": ";S2
5400 PRINT
5410 D=S=Z=P1=G1=P=P9=0
5420 IF S8<>0 THEN 5450
5430 Q=Q*(-1)
5440 GOTO 1200
5450 X=50-(30*Q)
5460 PRINT E$(2);" FROM ";Q$(ABS(Q-2));" 20 ";E$(6)
5470 S8=0
5480 GOTO 1230
5490 T=T+1
5500 IF T=H9-5 THEN 5590
5510 IF T=E9-5 THEN 5590
5520 IF T>H9 THEN 5540
5530 GOTO 5650
5540 IF T1>0 THEN 5630
5550 IF RND(-1)<.35 THEN 5650
5560 GOSUB 5810
5570 Q=T5

```

```

5580 GOTO 5390
5590 PRINT
5600 PRINT"TWO-MINUTE WARNING"
5610 PRINT
5620 GOTO 5650
5630 IF T<E9 THEN 5650
5640 IF RND(-1)>.50 THEN 5830
5650 RETURN
5660 B(2+0)=B(2+0)+P9
5670 IF Q>0 THEN 5700
5680 S2=S2+P9
5690 GOTO 5710
5700 S1=S1+P9
5710 RETURN
5720 IF P>8 THEN 5750
5730 B(8+0)=B(8+0)+Y
5740 GOTO 5760
5750 B(11+0)=B(11+0)+Y
5760 RETURN
5810 PRINT"HALFTIME"
5820 GOTO 5850
5830 PRINT"END OF GAME"
5840 Z9=1
5850 PRINT"DO YOU WANT GAME STATISTICS (YES OR NO)";
5860 INPUT C$
5870 IF C$="NO" THEN 5950
5875 IF C$="N" THEN 5950
5880 PRINT
5890 PRINT"STAT",A$,B$
5900 PRINT
5910 FOR I=0 TO 4
5920 PRINT D$(I+1),B(3+I*3),B(1+3*I)
5930 NEXT I
5940 PRINT
5950 Z=T1=1
5960 T=H9
5970 RETURN
5990 REM
6010 PRINT"ANOTHER GAME (YES OR NO)";
6020 INPUT C$
6030 IF C$="NO" THEN 6100
6035 IF C$="N" THEN 6100
6040 Z9=D=T=S1=S2=P1=P=S=G1=T1=Z=0
6050 FOR I=1 TO 15
6060 B(I)=0
6070 NEXT I
6080 RESTORE
6083 I8=1
6090 GOTO 350
6100 END

```

THIS IS PRO-STYLE FOOTBALL. YOU WILL BE THE
QUARTERBACK AND DEFENSIVE SIGNAL-CALLER FOR YOUR
TEAM. YOU HAVE 14 OFFENSIVE PLAYS (8 RUNS; 6 PASSES)
AND 4 DEFENSIVE ALIGNMENTS.

THERE IS A TWO-MINUTE WARNING BEFORE THE END
OF EACH HALF. FIVE TO SEVEN PLAYS ARE LEFT IN THE
HALF AT THAT POINT.

DO YOU WANT A LIST OF PLAYS (YES OR NO)?YES
CALL PLAYS AS FOLLOWS:

RUNS:

1=DIVE; 2=OFF TACKLE; 3=SCISSORS; 4=TRAP
5=SWEEP; 6=OPTION; 7=REVERSE; 8=DRAW

PASSES:

9=SIDELINE; 10=LOOK-IN; 11=ROLLOUT
12=SCREEN; 13='FLY'; 14='POST'

KICKS:

15=FIELD GOAL; 16=PUNT; 17=QUICK KICK

CALL DEFENSES AS FOLLOWS:

1='PRO' (4-3); 2='OKIE' (5-2); 3='SHORT-YDG'; 4='PREVENT'

YOUR TEAM?JETS

OPPONENT?RAMS

JETS WON THE TOSS.

DO YOU ELECT TO KICK OR RECEIVE?KICK

69 YARD KICKOFF
TOUCHBACK
BALL ON RAMS 20 YARD LINE
FIRST DOWN RAMS YARDS TO GO: 10

DEFENSE?4

SWEEP

2 YARD LOSS

HOLDING PENALTY ON JETS

PENALTY ACCEPTED: 15 YARDS

BALL ON RAMS 35 YARD LINE
FIRST DOWN RAMS YARDS TO GO: 10

DEFENSE?2

SIDELINE

INCOMPLETE

NO GAIN

BALL ON RAMS 35 YARD LINE
SECOND DOWN YARDS TO GO: 10

DEFENSE?2

DIVE

1 YARD GAIN

BALL ON RAMS 36 YARD LINE
THIRD DOWN YARDS TO GO: 9

FUNDS 1:

DESCRIPTION

Funds 1 performs financial projections for an investor. Funds 2 is used to plot the data from this program. The program is normalized to allow various investment plans to be compared.

USERS

Individuals and companies engaged in financial speculation and investment would have use for this program. In addition, it could be related to any homeowner, as a home is an investment. With this program one could do equity projections on a home for the purpose of loans or additional investment value.

INSTRUCTIONS

The program is self prompting and requires no maintenance prior to execution. The program normalizes all investments to \$100 and makes projections for a 50 year period. In order to correct for other investments the investment return need only be scaled by the ratio of \$100 to the true \$ investment. For example: if the investment was \$14,100 then the investment return projections should be multiplied by 141, and so forth. The program can be listed for additional operating information.

LIMITATIONS

Funds 1 uses two dimensional arrays, starting in line 100. Line 910 contains a TAB () statement and starting in line 950 Print Using statements are used. The source code requires 4K Bytes for storage. Execution is a function of the length of the projection as set in the DIM statement in line 100. With the present setting the program will require 37K Bytes of memory for execution. This amount can be significantly reduced if the A(32,100) table is reduced, however this will reduce the length of the projection. The program is presently set for operation without a storage file for the tables. If the results are not to be plotted with Funds 2, then there is no loss, however if Funds 2 is to be utilized, it would be advantageous to convert this program to use a storage file. Instructions for utilizing a storage file are given under Funds 2. This program has been written without file statements as most small system compilers do not have this statement and the object of this library is to be as universally compatible as possible.

FUNDS 1

20REM DESCRIPTION--SIMULATES LONG-RUN PERFORMANCE OF FUNDS.
25REM
40REM INSTRUCTIONS TYPE RUN TO USE .

45REM
50REM

60REM*****
100 DIM R(32,100),T(60),B(32)

130 PRINT "WHAT ARE EXPECTED RATE OF RETURN AND STANDARD DEVIATION";
140 INPUT R,S
150 PRINT "AT WHAT RATE DO YOU EXPECT NEW MONEY TO BE ADDED";
160 INPUT E2
170 PRINT "WHAT RATE OF SPENDING IS BUDGETED";
180 INPUT P2
190 REM USER SPECIFIES EXPECTED RATE OF RETURN R AND STD DEVIATION S
200 REM $U = \text{LOG}(1+R)$ HAS DISTRIBUTION WITH EXPECTED VALUE $U2$ AND
210 REM VARIANCE $S2$, WHILE $R1$ HAS MEAN R AND VARIANCE $S1$.
220 REM DEDUCE $S2$ AND $U2$ FROM R AND $S1$
230 LET $S1 = S^2$
240 LET $S2 = \text{LOG}(S1 / (1+R)^2 + 1)$
250 LET $U2 = \text{LOG}(1+R) - (1/2) * \text{LOG}(S1 / (1+R)^2 + 1)$
260 REM NUMBER OF PERIODS N
270 LET N=50
280 REM NUMBER OF SIMULATIONS N1
290 LET N1=100
300 REM SIZE OF FUND THREE PRECEDING YEARS
310 READ Y1,Y2,Y3
320 DATA 100,100,100
330 PRINT "AT WHAT INTERVALS DO YOU WISH TO'S";
340 INPUT I1
350 IF INT(N/I1)≠N/I1 THEN 380
360 PRINT "
 MUST BE POSSIBLE TO DIVIDE INTERVALS EVENLY INTO";N
370 GOTO 330
380 IF I1>4 THEN 410
390 PRINT "
 INTERVALS MUST BE GREATER THAN OR EQUAL TO 5"
400 GOTO 330
410 LET N2=N/I1
420 FOR I=1 TO N1
430 REM LAST PERIOD FUND SIZE Y3
440 LET T(0)=Y3
450 REM INITIALIZE E3,G1,H1

```

460 LET E3=0
470 LET G1=-1
480 LET H1=0
490 REM IF W RATE OF RETURN AND COMPOUND FORWARD N YEARS
500 FOR K=1 TO N
510 GOSUB 1020
520 LET T(K)=T(K-1)*(R1+1)
530 REM COMPUTE CONTRIBUTIONS AND WITHDRAWALS
540 GOSUB 1100
550 LET T(K)=T(K)-E1+C
560 REM PLACES SIZE AND WITHDRAWALS IN MATRIX. A(X,Y) IS FOR INTERVAL X,
570 REM SIMULATION Y.
580 IF K<5 THEN 600
590 IF K/I1<>INT(K/I1) THEN 610
600 GOSUB 1245
610 NEXT K
620 NEXT I
630 REM PUT MATRIX IN ASCENDING ORDER. A(X,Y) IS YTH LARGEST FUND SIZE,
640 REM IN SIMULATION X, IF X IS ODD, YTH LARGEST PAYOUT IN SIMULATION
650 REM X, IF X IS EVEN
660 FOR P=1 TO (N2+4)*2
670 FOR I=1 TO N1-1
680 FOR J=I+1 TO N1
690 IF A(P,I)<=A(P,J) THEN 730
700 LET Q1=A(P,I)
710 LET A(P,I)=A(P,J)
720 LET A(P,J)=Q1
730 NEXT J
740 NEXT I
750 NEXT P
760 PRINT "N",N,"N1",N1,"I1",I1
770 FOR I=1 TO (N2+4)*2
780 FOR J=1 TO N1
790 PRINT "A(";I;J;") = ";A(I,J)
800 NEXT J
810 NEXT I
820 FOR I=1 TO (N2+4)*2
830 LET B(I)=0
840 FOR J=1 TO N1
850 LET B(I)=B(I)+A(I,J)
860 NEXT J
870 LET B(I)=B(I)/N1
880 NEXT I
890 PRINT"

"
900 PRINT "TOTAL RETURN FUND AT END OF:"
910 PRINT TAB(61);"AVERAGE"
920 PRINT "PERIOD","UPPER BOUND"," MEAN","LOWER BOUND","SPENDING"
930 FOR I=1 TO (N2+4)*2 STEP 2

```

```

940 IF I>8 THEN 969
947 LET I9=I9+1
948 IF B(I)>999999.99 THEN 954
950 PRINT USING 1280,I9,A(I,N1),B(I),A(I,1),B(I+1)
952 GO TO 980
954 PRINT USING 1290,I9,A(I,N1),B(I),A(I,1),B(I+1)
960 GOTO 980
969 LET T8=T8+11
970 IF B(I)>999999.99 THEN 976
972 PRINT USING 1280,T8,A(I,N1),B(I),A(I,1),B(I+1)
974 GO TO 980
976 PRINT USING 1290,T8,A(I,N1),B(I),A(I,1),B(I+1)
980 NEXT I
990 PRINT
1000 PRINT

```

```

1020 LET R8=0
1030 FOR B8=1 TO 12
1040 LET R8=R8+RND(X)
1050 NEXT B8
1060 REM U2 IS EXPECTED VALUE OF U AND U=LOG(1+R1)
1070 LET R1=EXP(U2+SQR(S2)*(R8-6))
1080 LET R1=R1-1
1090 RETURN
1100 IF K<>1 THEN 1140
1110 LET C=E2*((Y1+Y2+Y3)/3)
1120 LET E1=((Y1+Y2+Y3)/3)*P2
1130 GO TO 1240
1140 IF K<>2 THEN 1180
1150 LET C=E2*((Y2+Y3+T(K-1))/3)
1160 LET E1=((Y2+Y3+T(K-1))/3)*P2
1170 GO TO 1240
1180 IF K<>3 THEN 1220
1190 LET C=E2*((Y3+T(K-1)+T(K-2))/3)
1200 LET E1=((Y3+T(K-1)+T(K-2))/3)*P2
1210 GO TO 1240
1220 LET C=E2*((T(K-1)+T(K-2)+T(K-3))/3)
1230 LET E1=((T(K-3)+T(K-2)+T(K-1))/3)*P2
1240 RETURN
1245 LET G1=G1+2
1250 LET A(G1,1)=T(K)
1255 LET H1=H1+2
1260 LET A(H1,1)=E1
1270 RETURN

```

```

1280:##          #####.##          #####.##          #####.##          #####.##
1290:##          #.#####↑↑↑↑↑          #.#####↑↑↑↑↑          #.#####↑↑↑↑↑          #.#####↑↑↑↑↑
1300 END

```

RUN

WHAT ARE EXPECTED RATE OF RETURN AND STANDARD DEVIATION ? .15, .13
AT WHAT RATE DO YOU EXPECT NEW MONEY TO BE ADDED ? 0
WHAT RATE OF SPENDING IS BUDGETED ? 0
AT WHAT INTERVALS DO YOU WISH TU'S ? 5

TOTAL RETURN FUND AT END OF:

PERIOD	UPPER BOUND	MEAN	LOWER BOUND	AVERAGE SPENDING
1	141.15	115.35	89.41	.00
2	210.33	134.16	84.03	.00
3	235.50	152.57	87.51	.00
4	290.92	176.12	97.47	.00
5	406.55	202.09	100.82	.00
10	1024.97	421.27	115.70	.00
15	2298.53	842.70	223.54	.00
20	5068.67	1716.58	245.37	.00
25	16643.58	3798.55	499.89	.00
3000	33823.20	7677.73	736.18	.00
35	87232.97	15754.43	1117.45	.00
40	157535.17	30699.46	2294.71	.00
45	333111.65	61627.30	6005.92	.00
50	640792.34	126456.81	8064.77	.00

EXAMPLE #2

RUN

WHAT ARE EXPECTED RATE OF RETURN AND STANDARD DEVIATION ? .08, .07
AT WHAT RATE DO YOU EXPECT NEW MONEY TO BE ADDED ? .01
WHAT RATE OF SPENDING IS BUDGETED ? .045
AT WHAT INTERVALS DO YOU WISH TO'S ? 10

TOTAL RETURN FUND AT END OF:

PERIOD	UPPER BOUND	MEAN	LOWER BOUND	AVERAGE SPENDING
1	118.18	104.69	84.57	4.50
2	144.53	110.23	83.37	4.57
3	149.62	114.60	82.01	4.72
4	163.51	120.25	84.33	4.94
10	280.73	160.99	74.11	6.51
20	556.70	254.66	79.37	10.51
30	1103.35	424.64	106.62	17.33
40	1936.71	668.20	145.23	27.75
50	3111.87	1063.14	212.58	43.30

FUNDS 2:

DESCRIPTION

This program is the plotting routine for Funds 1. It may also be used to plot other data bases, if they are formatted for this program's input.

USERS

Users of Funds 1 will have the most use for this program, however Funds 2 can be used to plot the results of any financial analysis.

INSTRUCTIONS

This program is also self prompting and will ask for all necessary inputs. For additional information list Funds 1 and Funds 2.

LIMITATIONS

As in Funds 1 this program uses two dimensional arrays, starting in line 100. In line 1331 the TAB () statement appears and is used throughout the program. The source code stores in 5K Bytes of memory. The execution length is a function of the reserved table size. With the DIM statements as presently set, it will require 38K Bytes of memory for storage and execution. This amount may be significantly reduced by reducing the "A" table. Due to the length of the tables it may be more convenient to store the data from Funds 1 in a use file and then have Funds 2 read this use file. If your compiler has file statement capability, the mods would be:

```
FUNDS 1      120  FILES: PLOT 1
              128  Scratch #1
              760  Write #1: N, N1, 11
              790  Write #1: A(I,J)
```

```
FUNDS 2      120  FILES: PLOT 1
              125  Read #1: N, N1, 11
              150  Read #1: A(I,J)
              155  Rem
```

FUNDS 2

25REM DESCRIPTION--PLOTS SIMULATION RESULTS OF FUNDS 1.

30REM

40REM INSTRUCTIONS - CALL FUNDS 1 AND LIST FOR INSTRUCTIONS

45REM

50REM

55REM

60REM*****

100 DIM B(35),A(35,100),U(35)

120 PRINT "INPUT N, N1, I1"

125 INPUT N,N1,I1

130 LET N2=N/I1

132 FOR I= 1 TO (N2+4)*2

140 FOR J=1 TO N1

150 PRINT "INPUT A(I,J)"

155 INPUT A(I,J)

160 NEXT J

170 NEXT I

180 PRINT "WHICH FUND";

190 INPUT K\$

200 PRINT "AT WHICH PERIOD DO YOU WANT A HISTOGRAM";

210 INPUT B

220 IF B<=0 THEN 1860

222 IF B<5 THEN 230

224 IF B/I1=INT(B/I1) THEN 230

226 PRINT "MUST BE ONE OF THE PERIODS FOR WHICH DATA ARE PROVIDED ABOVE."

228 GOTO 200

230 PRINT "DO YOU WISH TO RESPACE";

240 INPUT U\$

250 IF U\$="NO" THEN 280

260 PRINT "WHAT ENDPOINTS";

270 INPUT U5,U4,D5,D4

280 LET Z=1

290 PRINT "

"

292 IF B>4 THEN 300

293 IF Z=2 THEN 296

294 LET C1=2*B-1

295 GOTO 370

296 LET C1=C=2*B

297 GOTO 360

300 LET C=B/I1+4

310 IF Z=2 THEN 340

320 LET C1=2*C-1


```

330 GO TO 370
340 LET C1=2*C
350 LET C=2*C
360 IF A(C1,100)=0 THEN 1520
365 IF B=1 THEN 1520
370 IF U$="NO" THEN 420
380 IF Z=2 THEN 410
390 LET V1=(U4-U5)/10
400 GO TO 420
410 LET V1=(D4-D5)/10
420 LET D=A(C1,100)-A(C1,1)
430 LET D=D/10
440 IF D>10000 THEN 490
450 IF D>5000 THEN 470
460 GO TO 500
470 LET D=D+.05
480 GO TO 500
490 LET D=D+1
500 LET E8=U8=U7=E7= 10 E+7
510 IF U$="NO" THEN 630
520 IF Z=2 THEN 590
530 IF A(C1,1)<U5 THEN 570
540 LET E=U5
550 LET E8=U5
560 GO TO 650
570 LET E=A(C1,1)
580 GO TO 650
590 IF A(C1,1)<D5 THEN 630
600 LET E=D5
610 LET E7=D5
620 GO TO 680
630 LET E=A(C1,1)
640 GO TO 680
650 IF U4<A(C1,100) THEN 700
660 LET U8=U4
670 GO TO 700
680 IF D4<A(C1,100) THEN 700
690 LET U7=U4
700 LET F=H=0
710 LET F=F+1
720 LET B(F)=0
730 IF U$="YES" THEN 760
740 LET E=E+D
750 IF U$="NO" THEN 1190
760 IF F<>1 THEN 880
770 IF Z=2 THEN 830
780 IF E<>U5 THEN 810
790 LET E=E+V1
800 GO TO 820
810 LET E=U5
820 GO TO 870

```

```

830 IF E<>D5 THEN 860
840 LET E=E+U1
850 GO TO 870
860 LET E=D5
870 GO TO 890
880 LET E=E+U1
890 IF F<>1 THEN 1100
900 IF Z=2 THEN 1010
910 IF E8=10 E+7 THEN 960
920 LET G6=11
930 IF U8=10 E+7 THEN 1100
940 LET G6=10
950 GO TO 1100
960 IF U8=10 E+7 THEN 990
970 LET G6=11
980 GO TO 1100
990 LET G6=12
1000 GO TO 1100
1010 IF E7=10 E+7 THEN 1060
1020 LET G6=11
1030 IF U7=10 E+7 THEN 1100
1040 LET G6=10
1050 GO TO 1100
1060 IF U7=10 E+7 THEN 1090
1070 LET G6=11
1080 GO TO 1100
1090 LET G6=12
1100 IF F<>G6 THEN 1190
1110 IF Z=2 THEN 1150
1120 IF U8=10 E+7 THEN 1180
1130 LET E=U4
1140 GO TO 1170
1150 IF U7=10 E+7 THEN 1180
1160 LET E=D4
1170 GO TO 1190
1180 LET E=A(C1,100)
1190 FOR I=H+1 TO 100
1200 IF A(C1,I)>E+.01 THEN 1230
1210 LET B(F)=B(F)+1
1220 LET H=I
1230 NEXT I
1240 LET U(F)=E
1250 IF U$="YES" THEN 1280
1260 IF F <>10 THEN 710
1270 GO TO 1300
1280 IF F<>G6 THEN 710
1290 GO TO 1540
1300 IF Z=2 THEN 1350
1310 PRINT "          FUND ";K$;" DISTRIBUTION OF FUND VALUES AT";B;" YEARS"
1320 PRINT
1330 LET G=INT(A(C1,1))

```

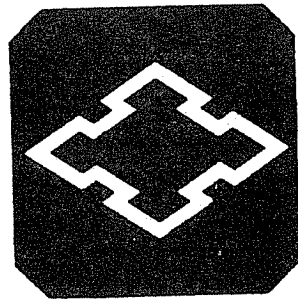
```

1331 PRINT G;TAB(12);":"
1340 GO TO 1380
1350 PRINT"          FUND ";K$;" DISTRIBUTION OF WITHDRAMALS AT";B;"YEARS"
1360 PRINT
1370 LET G=INT(A(C1,1)*100)/100
1371 PRINT G;TAB(12);":"
1380 FOR I= 1 TO 10
1390 PRINT TAB(12);":";
1400 FOR T=1 TO B(I)
1410 PRINT TAB(13);"*";
1420 NEXT T
1430 PRINT
1440 IF Z=2 THEN 1470
1450 LET G=G+D
1451 PRINT INT(G);TAB(12);":"
1460 GO TO 1480
1470 LET G=G+D
1471 PRINT INT(G*100)/100;TAB(12);":"
1480 NEXT I
1490 IF Z=2 THEN 1520
1500 LET Z=2
1510 GO TO 290
1520 PRINT "
"
1530 GO TO 200
1540 IF Z=2 THEN 1630
1550 PRINT"          FUND ";K$;" DISTRIBUTION OF FUND VALUES AT";B;"YEARS"
1560 PRINT
1570 IF E8=10 E+7 THEN 1600
1580 LET G=INT(E8)
1581 PRINT G;TAB(12);":"
1590 GO TO 1710
1600 LET G=INT(A(C1,1))
1601 PRINT G;TAB(12);":"
1610 GO TO 1710
1620 IF INT(A(C1,1)*100)/100 = 0 THEN 1860
1630REM
1640 PRINT"          FUND ";K$;" DISTRIBUTION OF WITHDRAWALS AT";B;"YEARS"
1650 PRINT
1660 IF INT (A(C1,1)*100)/100= 0 THEN 1850
1670 IF E7=10 E+7 THEN 1700
1680 LET G=INT(E7)
1681 PRINT G;TAB(12);":"
1690 GO TO 1710

1690 GO TO 1710
1700 LET G=INT(A(C1,1)*100)/100

```

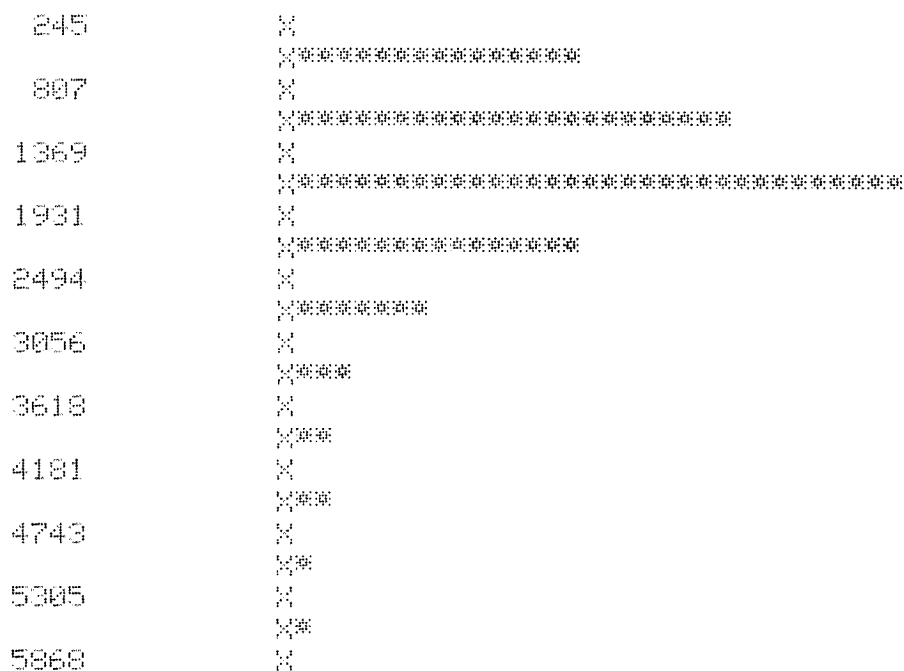
```
1701 PRINT G;TAB(12);":"
1710 FOR I=1 TO G6
1720 PRINT TAB(12);":";
1730 FOR T=1 TO B(I)
1740 PRINT TAB(13);"*";
1750 NEXT T
1760 PRINT
1770 PRINT INT(U(I));TAB(12);":"
1780 GO TO 1800
1790 PRINT INT(U(I)*100+.05)/100;TAB(12);":"
1800 NEXT I
1810 IF Z=2 THEN 1840
1820 LET Z=2
1830 GO TO 290
1840 PRINT "
"
1850 GO TO 200
1860 END
```



This example plots the results of the first problem from the program FUNDS 1.

WHICH FUND? ?ABC
AT WHICH PERIOD DO YOU WANT A HISTOGRAM ?20
DO YOU WISH TO RESPACE ?NO

FUND ABC DISTRIBUTION OF FUND VALUES AT 20 YEARS

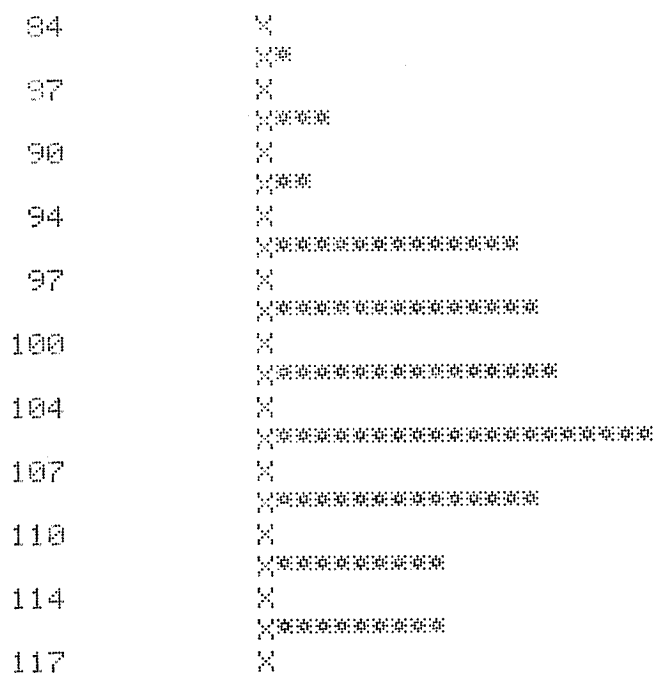


AT WHICH PERIOD DO YOU WANT A HISTOGRAM ?0

This example plots the results of the second problem from the program FUNDS 1.

WHICH FUND ?BCD
AT WHICH PERIOD DO YOU WANT A HISTOGRAM ?1
DO YOU WISH TO RESPACE ?NO

FUND BCD DISTRIBUTION OF FUND VALUES AT 1 YEAR



GO-MOKU:

DESCRIPTION

This program is a simulation of an old Chinese game. It is played on a 9 x 9 board and the object of the play is to place 5 in a row without being blocked by your opponent. Your opponent is the computer.

USERS

If you like a game offering a fair degree of skill you will enjoy playing GO-MOKU. The game can be analyzed and the methodology may be reduced to an algorithm as in TIC-TAC-TOE for those so inclined.

INSTRUCTIONS

Upon execution the program prints out all necessary instructions and game rules. All inputs will be prompted by the game program. The playing board will be printed after each move.

LIMITATIONS

This program uses two dimensional arrays; see line 90. The source code is 5K Bytes long and the game will execute in 11K Bytes of memory.

```

10 REM THIS IS THE GAME GO-HOKU
20 PRINT "YOUR OBJECTIVE IS TO PLACE 5 MARKS"
30 PRINT "IN A STRAIGHT LINE (EITHER HORIZONTAL, VERTICAL, OR DIAG-"
40 PRINT "ONAL) ON A 9X9 BOARD, PLAYING AGAINST THE COMPUTER. ENTER"
50 PRINT "YOUR MOVE AS 2 NUMBERS: THE POSITIONS ACROSS AND"
60 PRINT "THEN DOWN FROM THE UPPER LEFT CORNER OF THE BOARD. THE COM-"
70 PRINT "PUTER WILL THEN TELL YOU ITS MOVE AND PRINT THE BOARD. 'Y'"
80 PRINT "IS YOU; 'C' IS THE COMPUTER. GOOD LUCK."
90 DIM A(27,27),P(50)
100 LET L9=200
110 FOR J=1 TO 27
120 FOR K=1 TO 27
130 LET A(J,K)=3
140 NEXT K
150 NEXT J
160 LET S=0
170 FOR J=10 TO 18
180 FOR K=10 TO 18
190 LET A(J,K)=0
200 NEXT K
210 NEXT J
220 IF S=1 THEN 2400
230 LET S=1
240 GOTO 260
250 GOSUB 2100
260 PRINT
270 PRINT
280 INPUT J1,K1
290 IF J1+K1 =0 THEN 170
300 LET J1=J1+9
310 LET K1=K1+9
320 IF A(J1,K1)<>0 THEN 2300
330 LET A(J1,K1)=2
340 LET D2=2
350 LET L3=0
360 LET D4=0
370 LET L2=81
380 GOSUB 840
390 LET D5=D4
400 IF D4>79 THEN 1930
410 LET D2=1
420 LET L2=65
430 LET D4=0
440 GOSUB 840
450 LET D6=D4

```



```

460 IF D5>31 THEN 480
470 IF D6<31 THEN 770
480 IF L3>0 THEN 1950
490 IF D5<65 THEN 520
500 LET D2=2
510 GOTO 550
520 LET D2=1
530 LET L2=55
540 IF D6<55 THEN 580
550 GOSUB 840
560 IF L3=3 THEN 2010
570 GOTO 2030
580 IF D5<55 THEN 660
590 LET D2=2
600 GOSUB 840
610 LET L2=31
620 LET L9=55
630 GOSUB 840
640 GOSUB 1680
650 GOTO 2010
660 LET L2=31
670 GOSUB 840
680 IF L3<2 THEN 700
690 GOSUB 1680
700 LET L3=0
710 LET D2=2
720 GOSUB 840
730 IF L3=0 THEN 780
740 IF L3=1 THEN 1990
750 GOSUB 1680
760 GOTO 2030
770 LET D2=2
780 LET L2=D5-5
790 LET L3=0
800 GOSUB 840
810 IF L3>0 THEN 2030
820 PRINT "DRAW, ";
830 GO TO 2060
840 FOR J=1 TO 9
850 GOSUB 1610
860 FOR K=1 TO 10
870 GOSUB 1120
880 NEXT K
890 NEXT J
900 FOR K=1 TO 9
910 GOSUB 1610
920 FOR J=1 TO 10
930 GOSUB 1120
940 NEXT J
950 NEXT K
960 FOR L=4 TO -4 STEP -1

```

```

970 GOSUB 1610
980 FOR J=1 TO 10
990 LET K=J+L
1000 GOSUB 1120
1010 NEXT J
1020 NEXT L
1030 FOR L=5 TO 14
1040 GOSUB 1610
1050 FOR K=1 TO 10
1060 LET J=L-K
1070 GOSUB 1120
1080 NEXT K
1090 NEXT L
1100 LET L9=200
1110 RETURN
1120 LET B=A(J+9,K+9)
1130 IF B=0 THEN 1460
1140 IF B=D2 THEN 1350
1150 IF U1=0 THEN 1180
1160 IF C1=66 THEN 1180
1170 LET C1=C1-10
1180 LET S9=0
1190 LET U1=0
1200 IF C1<=C2 GO TO 1220
1210 LET C2=C1
1220 IF C5>=5 THEN 1250
1230 GOSUB 1610
1240 RETURN
1250 LET C5=0
1260 IF C2<=D4 THEN 1280
1270 LET D4=C2
1280 IF C2<L2 THEN 1240
1290 IF C2>L9 THEN 1240
1300 IF L1=0 THEN 1240
1310 LET P(L3)=Q(L1-1)
1320 LET L3=L3+1
1330 LET L1=L1-1
1340 GOTO 1300
1350 IF U1=1 THEN 1400
1360 LET U1=1
1370 IF S9=0 THEN 1400
1380 GOSUB 1760
1390 LET S9=0
1400 LET C6=C7+C6+1
1410 LET C7=0
1420 LET C5=C5+1
1430 IF C6>5 THEN 1240
1440 LET C1=C1+20
1450 GOTO 1240
1460 IF U1=1 THEN 1490
1470 LET S9=10*J+K

```

```

1480 GOTO 1510
1490 GOSUB 1760
1500 LET U1=0
1510 LET C5=C5+1
1520 IF C1<=0 THEN 1540
1530 LET C7=C7+1
1540 IF C1<=C2 THEN 1580
1550 LET C2=C1
1560 LET C1=C1-4
1570 GOTO 1240
1580 IF C1>=0 THEN 1560
1590 LET C1=0
1600 GO TO 1240
1610 LET L1=0
1620 LET C1=-10
1630 LET C2=0
1640 LET C5=0
1650 LET C6=0
1660 LET C7=0
1670 RETURN
1680 FOR H=0 TO L3-2
1690 FOR I=H+1 TO L3-1
1700 IF P(H)=P(I) THEN 1740
1710 NEXT I
1720 NEXT H
1730 RETURN
1740 LET P(0)=P(H)
1750 GOTO 1990
1760 IF C1<65 THEN 1820
1770 IF L1=0 THEN 1820
1780 IF L1=1 THEN 1870
1790 LET Q(0)=Q(1)
1800 LET L1=L1-1
1810 GOTO 1780
1820 IF S9=0 THEN 1850
1830 LET Q(L1)=S9
1840 GOTO 1860
1850 LET Q(L1)=10*J+K
1860 LET L1=L1+1
1870 RETURN
1880 LET J=INT(P(0)/10)+9
1890 LET K=P(0)-10*(J-9)+9
1900 LET A(J,K)=1
1910 PRINT J-9;K-9
1920 RETURN
1930 PRINT "YOU WIN"
1940 GOTO 2060
1950 GOSUB 1880
1960 GOSUB 2180
1970 PRINT "COMPUTER WINS!"
1980 GOTO 2060

```

```

1990 GOSUB 1880
2000 GOTO 250
2010 LET P(0)=P(1)
2020 GOTO 1990
2030 LET L9=INT(L3*RND(X))
2040 LET P(0)=P(L9)
2050 GOTO 1990
2060 PRINT
2070 PRINT
2080 PRINT
2090 PRINT " ** WOULD YOU LIKE TO PLAY AGAIN ** "
2100 PRINT
2110 INPUT R$
2120 PRINT
2130 PRINT
2140 PRINT
2150 IF R$="Y" GO TO 170
2160 IF R$="YES" GO TO 170
2170 GO TO 2450
2180 PRINT
2190 PRINT "      1 2 3 4 5 6 7 8 9"
2200 FOR J=10 TO 18
2210 PRINT J-9;
2220 FOR K=10 TO 18
2230 FOR L=K TO 18
2240 IF A(L,J)<>0 THEN 2270
2250 NEXT L
2260 GOTO 2310
2270 IF A(K,J)=0 THEN 2360
2280 IF A(K,J)=1 THEN 2340
2290 PRINT "Y ";
2300 NEXT K
2310 PRINT
2320 NEXT J
2330 RETURN
2340 PRINT "C ";
2350 GOTO 2300
2360 PRINT ". ";
2370 GOTO 2300
2380 PRINT "BAD PLAY; MAKE A NEW ONE"
2390 GOTO 260
2400 LET A(14,14)=1
2410 PRINT "COMP. PLAYS 1ST WITH 5 5"
2420 LET S=0
2430 GOTO 250
2440 STOP
2450 END

```

JACK:

DESCRIPTION

Calling all gamblers or would be gamblers — this is a simulation of the game of Black-Jack. The program adheres to the same house rules as used in the Las Vegas gambling casinos and is an exceptionally realistic version of the game.

USERS

Persons who like to play cards, gamble, or just play games will enjoy JACK, and if you've always dreamed of sitting in a casino in Vegas playing cards, with this program only the setting will be different. The winnings obtained with this program are very close to those obtained in the real casinos, except you can't spend your winnings but then like wise you don't have to make monthly payments to repay your losses.

INSTRUCTIONS

To play Jack simply type RUN after the program is loaded. The program will ask if you would like instructions. A response of "0" will start the game immediately, any other numeric response will cause a detailed listing of the instructions to be printed. The program is instructional and will prompt for all inputs.

LIMITATIONS

The source code requires 7K Bytes of memory for storage and 8K for execution.

```

20 DIM W(10),E(10),T(10),U(10)
25 PRINT " DO YOU NEED "

26 PRINT"INSTRUCTIONS (1=YES,0=NO)"
28 INPUT K
30 IF K=0 THEN 112
32 PRINT

        HERE ARE THE LAS VEGAS RULES FOR PLAYING BLACKJACK:"
34 PRINT
36 PRINT">WAGER: THE HOUSE LIMIT IS $500, SO TYPE IN A NUMBER "
38 PRINT" FROM 0 TO 500. TO TERMINATE GAME, ENTER ZERO."
40 PRINT
42 PRINT">THE DEAL: I DEAL MYSELF 2 CARDS AND SHOW YOU ONE. THEN I"
44 PRINT" DEAL YOU TWO CARDS, AND ASK IF YOU WANT A HIT (ANOTHER"
46 PRINT" CARD). YOU HAVE SEVERAL OPTIONS DEPENDING ON THE CARDS"
48 PRINT" YOU HOLD AND MY UP CARD:"
50 PRINT" * STAND - BY TYPING A ZERO"
52 PRINT" * TAKE A HIT - BY TYPING A ONE"
54 PRINT" * GO DOWN FOR DOUBLES - BY TYPING A TWO"
56 PRINT" * SPLIT A PAIR - BY TYPING A THREE"
58 PRINT
60 PRINT">INSURANCE: IF MY UP CARD IS AN ACE, I WILL ASK IF YOU"
62 PRINT" WANT INSURANCE. IF YOU DO TYPE A ONE, BETTING ONE-HALF"
64 PRINT" OF YOUR WAGER THAT I DO HAVE BLACKJACK. IF I DO, I PAY"
66 PRINT" 2-TO-1 ON YOUR INSURANCE BET. YOU LOSE YOUR ORIGINAL WAGER"
68 PRINT" SINCE I HAVE BLACKJACK, SO WE ARE EVEN FOR THE HAND."
70 PRINT" IF I DON'T HAVE BLACKJACK, YOU LOSE YOUR INSURANCE BET"
72 PRINT" AND THE GAME CONTINUES. "
74 PRINT
76 PRINT" IF YOU REFUSE INSURANCE (BY TYPING A ZERO) THE GAME"
78 PRINT" CONTINUES AS NORMAL."
80 PRINT
82 PRINT">THE PLAY: WHEN YOU FINALLY STAND (BY TYPING A ZERO)"
84 PRINT" I WILL DRAW CARDS UNTIL:"
86 PRINT" *I HAVE AT LEAST A HARD 17 (HARD MEANS THE TOTAL"
88 PRINT" DOES NOT INCLUDE AN ACE BEING COUNTED AS 11)"
90 PRINT" *I HAVE A SOFT 18 (SOFT MEANS THE TOTAL INCLUDES AN"
92 PRINT" ACE COUNTED AS 11)"
94 PRINT" *I REACH A TOTAL OF 21"
96 PRINT" *I EXCEED 21 AND BUST"
98 PRINT
100 PRINT">ITEMS:"
102 PRINT" *I PAY 1.5-TO-1 ON BLACKJACK"
104 PRINT" *I DON'T RECOGNIZE 5-CARDS-AND-UNDER"
106 PRINT" *YOU MAY DOUBLE DOWN ON A SPLIT HAND"
108 PRINT" *YOU DON'T LOSE ON A TIE HAND...WE PUSH"

```

110 PRINT"

<<<GOOD LUCK>>>

"

112 PRINT"THE 600 IS THE DEALER AND GETS A BREAK AT 1945 HOURS. WHAT"

114 PRINT"TIME IS IT NOW";

116 LET K=0

118 LET M1=0

120 INPUT N

122 FOR I = 1 TO N/2

124 LET X=T(10*RND(X))

126 NEXT I

128 DIM D(52)

130 FOR A=1 TO 52

132 LET D(A) = 0

134 NEXT A

136 DIM Q(52)

138 FOR A=0 TO 39 STEP 13

140 FOR C=1 TO 13

142 LET Q(A+C)=C

144 NEXT C

146 NEXT A

148 PRINT

150 LET K=K+1

152 FOR P=1 TO 5

154 LET E(P)=0

156 LET U(P) = 0

158 LET T(P)=0

160 NEXT P

162 LET U(3)=1

164 PRINT

166 PRINT "WAGER";

168 LET P=1

170 INPUT W

172 LET W(2)=W

174 IF W<=0 THEN 582

176 IF W<=500 THEN 182

178 PRINT "THAT'S TOO MUCH - HOUSE LIMIT IS \$500."

180 GOTO 166

182 PRINT

184 PRINT "I SHOW " ,

186 GOSUB 366

188 IF E(1)=0 THEN 192

190 LET U(4)=1

192 LET U(5)=1

194 GOSUB 366

196 LET M=X

198 LET P=2

200 PRINT "FIRST CARD IS";

202 GOSUB 366

```

204 LET G=X
206 PRINT "NEXT CARD IS";
208 GOSUB 366
210 IF U(2)>0 THEN 262
212 LET S=X
214 IF U(3)<>1 THEN 262
216 IF T(P)<>21 THEN 230
218 PRINT "***BLACKJACK***"
220 PRINT "MY HOLE CARD WAS ";
222 LET X=M
224 GOSUB 434
226 LET W1=W1+1.5*W
228 GOTO 538
230 IF U(4)=0 THEN 250
232 PRINT "INSURANCE ANYONE (TYPE 1 OR 0)";
234 INPUT I
236 IF I=0 THEN 250
238 IF T(1)<>21 THEN 246
240 LET W1=W1+W
242 PRINT "YOU WIN $";W1;" ON YOUR INSURANCE BET";
244 GOTO 250
246 LET W1=W1-W/2
248PRINT"YOU LOST $";W/2;" ON INSURANCE BET - I DONT HAVE BLKJAK"
250 IF T(1)<>21 THEN 262
252 PRINT "I HAVE BLACKJACK"
254 PRINT "MY HOLE CARD IS ";
256 LET X=M
258 GOSUB 434
260 GOTO 534
262 IF T(P)<=21 THEN 278
264 IF E(P)>0 THEN 274
266 PRINT "YOU BUSTED, ";
268 PRINT "YOUR TOTAL IS ";T(P)
270 IF U(2)=1 THEN 328
272 GOTO 494
274 LET E(P)=E(P)-1
276 LET T(P)=T(P)-10
278 IF U(1)=2 THEN 268
280 LET U(3)=U(3)+1
282 PRINT "HIT";
284 INPUT U(1)
286 IF U(1)<>3 THEN 350
288 IF U(2)>0 THEN 346
290 IF U(3)<>2 THEN 346
292 IF Q(G)=Q(S) THEN 298
294 PRINT "NOW IS THAT A PAIR "
296 GOTO 282
298 LET U(2)=1
300 IF Q(G)<>1 THEN 304
302 LET U(1)=2
304 LET P=3

```



```

306 PRINT "          PLAY HAND ONE NOW"
308 PRINT "FIRST CARD IS";
310 LET W(3)=4
312 LET X=G
314 GOSUB 318
316 GOTO 206
318 GOSUB 414
320 GOSUB 434
322 LET V(3)=1
324 LET T(P)=C
326 RETURN
328 LET P=2
330 LET V(2)=2
332 PRINT "          PLAY HAND TWO NOW"
334 PRINT "FIRST CARD IS";
336 LET X=S
338 GOSUB 318
340 IF Q(G)=1 THEN 344
342 LET U(1)=0
344 GOTO 206
346 PRINT "NO SPLITS NOW -- TRY AGAIN"
348 GOTO 282
350 IF U(1)<>2 THEN 360
352 IF U(3)=2 THEN 358
354 PRINT "TOO LATE TO DOUBLE, CHARLIE."
356 GOTO 282
358 LET W(P)=2*W(P)
360 IF U(1)>0 THEN 206
362 GOTO 268
364 REM DEAL
366 GOSUB 380
368 LET T(P)=T(P)+C
370 IF U(5)=0 THEN 376
372 LET U(5)=0
374 RETURN
376 GOSUB 434
378 RETURN
380 LET N=10 *(1+ABS(COS(N+W1)))
382 FOR A=1 TO N
384 LET X=INT(52.99999999*RND(Y))
386 IF X=0 THEN 384
388 NEXT A
390 IF D(X)=0 THEN 410
392 LET R=R+1
394 IF R<50 THEN 380
396 FOR A=1 TO 52
398 IF D(A) =K THEN 402
400 LET D(A)=0
402 NEXT A
404 LET R=0
406 PRINT "I RESHUFFLED"

```

```

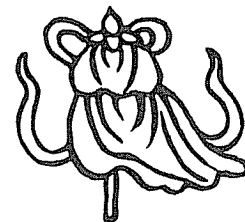
408 GOTO 380
410 LET P=0
412 LET D(X)=K
414 IF Q(X)<>1 THEN 422
416 LET C=11
418 LET E(P)=E(P)+1
420 RETURN
422 IF Q(X)>10 THEN 428
424 LET C=Q(X)
426 RETURN
428 LET C=10
430 RETURN
432 REM PRINT CARD
434 GOSUB 440
436 GOSUB 470
438 RETURN
440 IF Q(X)<>1 THEN 446
442 PRINT " ACE ";
444 RETURN
446 IF Q(X)>10 THEN 452
448 PRINT Q(X);
450 RETURN
452 IF Q(X)>11 THEN 458
454 PRINT " JACK ";
456 RETURN
458 IF Q(X)>12 THEN 464
460 PRINT " QUEEN ";
462 RETURN
464 PRINT " KING ";
466 RETURN
468 REM PRINT SUIT
470 IF X>39 THEN 480
472 IF X>26 THEN 484
474 IF X>13 THEN 488
476 PRINT "OF SPADES"
478 RETURN
480 PRINT "OF CLUBS"
482 RETURN
484 PRINT "OF HEARTS"
486 RETURN
488 PRINT "OF DIAMONDS"
490 RETURN
492 REM DEALER PLAYS
494 LET P=2
496 PRINT "MY HOLE CARD IS  ",
498 LET X=M
500 GOSUB 434
502 IF T(2)<22 THEN 508
504 IF U(2)=0 THEN 534
506 IF T(3)>21 THEN 534
508 LET P=1

```

```

510 IF T(1)<17 THEN 548
512 IF T(1)>17 THEN 516
514 IF E(1)>0 THEN 548
516 IF T(1)>21 THEN 554
518 LET P=2
520 PRINT "MY TOTAL IS ";T(1)
522 IF T(P)>21 THEN 534
524 IF T(1)>21 THEN 530
526 IF T(1)>T(P) THEN 534
528 IF T(1)=T(P) THEN 536
530 LET W1=W1+W(P)
532 GOTO 536
534 LET W1=W1-W(P)
536 IF U(2)>0 THEN 542
538 GOSUB 566
540 GOTO 150
542 LET P=3
544 LET U(2)=0
546 GOTO 522
548 PRINT "I DRAW          ";
550 GOSUB 366
552 GOTO 508
554 IF E(1)= 0 THEN 562
556 LET E(1) = E(1) -1
558 LET T(1)=T(1)-10
560 GOTO 508
562 PRINT "I BUSTED***";
564 GOTO 518
566 IF W1<0 THEN 574
568 IF W1=0 THEN 578
570 PRINT"YOU'RE AHEAD $";W1
572 RETURN
574 PRINT "YOU'RE BEHIND $";-W1
576 RETURN
578 PRINT "YOU'RE EVEN"
580 RETURN
582 END

```



HERE ARE THE LAS VEGAS RULES FOR PLAYING BLACKJACK:

>WAGER: THE HOUSE LIMIT IS \$500, SO TYPE IN A NUMBER FROM 0 TO 500. TO TERMINATE GAME, ENTER ZERO.

>THE DEAL: I DEAL MYSELF 2 CARDS AND SHOW YOU ONE. THEN I DEAL YOU TWO CARDS, AND ASK IF YOU WANT A HIT (ANOTHER CARD). YOU HAVE SEVERAL OPTIONS DEPENDING ON THE CARDS YOU HOLD AND MY UP CARD:

- * STAND - BY TYPING A ZERO
- * TAKE A HIT - BY TYPING A ONE
- * GO DOWN FOR DOUBLES - BY TYPING A TWO
- * SPLIT A PAIR - BY TYPING A THREE

>INSURANCE: IF MY UP CARD IS AN ACE, I WILL ASK IF YOU WANT INSURANCE. IF YOU DO TYPE A ONE, BETTING ONE-HALF OF YOUR WAGER THAT I DO HAVE BLACKJACK. IF I DO, I PAY 2-TO-1 ON YOUR INSURANCE BET. YOU LOSE YOUR ORIGINAL WAGER SINCE I HAVE BLACKJACK, SO WE ARE EVEN FOR THE HAND. IF I DON'T HAVE BLACKJACK, YOU LOSE YOUR INSURANCE BET AND THE GAME CONTINUES.

IF YOU REFUSE INSURANCE (BY TYPING A ZERO) THE GAME CONTINUES AS NORMAL.

>THE PLAY: WHEN YOU FINALLY STAND (BY TYPING A ZERO) I WILL DRAW CARDS UNTIL:

- *I HAVE AT LEAST A HARD 17 (HARD MEANS THE TOTAL DOES NOT INCLUDE AN ACE BEING COUNTED AS 11)
- *I HAVE A SOFT 18 (SOFT MEANS THE TOTAL INCLUDES AN ACE COUNTED AS 11)
- *I REACH A TOTAL OF 21
- *I EXCEED 21 AND BUST

>ITEMS:

- *I PAY 1.5-TO-1 ON BLACKJACK
- *I DON'T RECOGNIZE 5-CARDS-AND-UNDER
- *YOU MAY DOUBLE DOWN ON A SPLIT HAND
- *YOU DON'T LOSE ON A TIE HAND...WE PUSH

<<<GOOD LUCK>>>

LIFE:

DESCRIPTION

This program is a very detailed version of the game of "Life" as written up in the October 1970 issue of Scientific American. It has been modified to allow one to play either against the computer or against another player. The object of the game is to destroy your opponents life units while saving at least some of your life units. This is more than just a game as it actually follows patterns found in real life environments.

USERS

Game enthusiasts of all types will find something of interest in this game. "Life" will be of special interest to individuals involved with population projections, economic growth situations, and the problems related to over and under developed areas.

INSTRUCTIONS

The program is self prompting and should be listed for a full set of instructions before playing your initial game. Program instructions are available when the program is initiated.

LIMITATIONS

The source code for this simulation requires 13K Bytes of memory for storage and 21K for execution, this does not include the amount of memory required for storage of your Basic compiler. This program uses two dimensional arrays; see line 106. Line 108 has a MAT A = ZER and 110 has a MAT B = ZER. Lines 104 and 122 contain Restore statements.

LIFE

```

50 REM *** THE PROGRAM ITSELF IS BASED ON THE GAME "LIFE" WHICH WAS INVENTED
60 REM BY JOHN HORTON CONWAY, AS WRITTEN UP IN SCIENTIFIC AMERICAN, OCT 1970
70 REM
80 REM
90 REM      ***** GLOSSARY OF VARIABLE NAMES *****
100 REM
101 E=E9=0
102 REM ***** SET UP BOARD AND HOME REGIONS *****
104 RESTORE
106 DIM A(13,32),B(13,32)
108 MAT A = ZER
110 MAT B = ZER
112 FOR J = 4 TO 2 STEP -1
114     FOR I = 2 TO 12
116         READ A(I,J)
118     NEXT I
120 NEXT J
122 RESTORE
124 FOR J = 29 TO 31
126     FOR I = 12 TO 2 STEP -1
128         READ A(I,J)
130     NEXT I
132 NEXT J
134 DATA 1,1,0,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,0,0,1,1
136 DATA 1,16,1,20,1,24,2,16,1,19,1,21,1,23,1,25,3,16,3,20,3,24,4,16
138 DATA 5,16,5,20,5,24,6,19,6,21,6,23,6,25,7,20,7,24,9,20,9,24,10,19
140 DATA 10,21,10,23,10,25,11,20,11,24
142 REM THAT WAS THE FIRST MACHINE PATTERN
144 DATA 1,17,2,17,3,17,4,17,5,17,6,17,7,17,8,17,9,17,10,17,11,17,1,20
146 DATA 1,21,2,20,4,20,4,25,5,20,5,21,5,24,5,25,7,20,7,21,7,24,7,25
148 DATA 8,20,8,25,10,20,11,20,11,21
150 REM THAT WAS THE SECOND MACHINE PATTERN
152 DATA 1,16,11,16,5,17,5,18,6,18,6,19,7,18,1,23,1,24,2,22,2,25,3,21
154 DATA 3,23,3,25,4,21,4,24,5,22,5,23,7,22,7,23,8,21,8,24,9,21,9,23
156 DATA 9,25,10,22,10,25,11,23,11,24
158 REM THAT WAS THE THIRD MACHINE PATTERN
160 REM
170 REM
180 REM
190 REM
200 PRINT
210 PRINT
220 PRINT
230 PRINT
240 PRINT
250 PRINT

```

```

260 PRINT
270 PRINT"          BECOME A TWO-DIMENSIONAL TYRANT "
280 PRINT
290 PRINT"  EXPERIENCE THE JOY OF CONFLICT AS YOU DETERMINE THE DESTINY"
300 PRINT"    OF YOUR WORLD OF LIFE-UNITS"
310 PRINT
320 PRINT"  WATCH HELPLESSLY AS YOUR LIFE-UNITS AIMLESSLY BLUNDER ABOUT!"
330 PRINT
340 PRINT"          PLAY THE          LIFE'!"
350 PRINT
360 PRINT"  DO YOU WANT A BASIC EXPLANATION OF THE GAME ";
370 GOSUB 19000
380 IF @ = 0 THEN 1400
390 PRINT
400 PRINT
410 PRINT"  THE GAME          LIFE' IS BASED ON THE GAME LIFE."
420 PRINT
430 PRINT"  LIFE-UNITS ARE COUNTERS ON AN INFINITE GRID. THE PRESENCE OF A LIFE-"
440 PRINT"  UNIT IS INDICATED AS A ZERO ON THE PRINTED GAME BOARD. ON THE BOARD"
450 PRINT"  EACH LIFE-UNIT IS SURROUNDED BY A MAXIMUM OF EIGHT NEIGHBORS. (ALL"
460 PRINT"  ADJACENT UNITS ARE COUNTED, INCLUDING DIAGONAL NEIGHBORS)"
470 PRINT
480 PRINT"  THE GAME PROCEEDS FROM ONE GENERATION TO ANOTHER. THE LIFE-UNITS"
490 PRINT"  THEMSELVES DO NOT MOVE; THEY WILL EITHER LIVE OR DIE. THE CURRENT"
500 PRINT"  POPULATION OF AN AREA WILL DETERMINE THE POPULATION OF THE THE NEXT"
510 PRINT"  GENERATION. EXISTING LIFE-UNITS MAY DIE; OR NEW ONES MAY BE BORN IN"
520 PRINT"  EMPTY SPACES; ACCORDING TO THE FOLLOWING RULES:"
530 PRINT
540 PRINT"  DEATH  - A LIFE-UNIT WILL DIE IF IT HAS LESS THAN TWO NEIGHBORS..."
550 PRINT"          (FROM ISOLATION); OR MORE THAN THREE NEIGHBORS... (FROM "
560 PRINT"          OVERCROWDING). "
570 PRINT
580 PRINT"  LIFE   - A LIFE-UNIT WILL LIVE IF IT HAS TWO OR THREE NEIGHBORS."
590 PRINT
600 PRINT"  BIRTH  - A NEW LIFE-UNIT WILL BE BORN IN AN EMPTY SPACE THERE ARE"
610 PRINT"          EXACTLY THREE NEIGHBORING LIFE-UNITS."
620 PRINT
630 PRINT"  EXAMPLES: "
640 PRINT
650 PRINT"          0 0"
660 PRINT
670 PRINT"  THE ABOVE LIFE-UNITS WILL BOTH DIE BECAUSE EACH HAS ONLY"
680 PRINT"  ONE NEIGHBOR."
690 PRINT
700 PRINT"          0 0"
710 PRINT"          0 0"
720 PRINT"          0 0"
730 PRINT
740 PRINT"  TWO OF THE ABOVE LIFE-UNITS WILL DIE FROM OVERCROWDING. TWO NEW"
750 PRINT"  LIFE-UNITS WILL BE BORN."

```

```

760 PRINT" THE NEXT GENERATION OF THE ABOVE GROUP WILL BE:"
770 PRINT
780 PRINT"   0 0"
790 PRINT"   0   0"
800 PRINT"   0 0"
810 PRINT
820 PRINT" THIS PATTERN IS STABLE; EACH LIFE-UNIT HAS TWO NEIGHBORS--SO IT"
830 PRINT" LIVES --- AND NO SPACE HAS EXACTLY THREE NEIGHBORS--SO NO NEW"
840 PRINT" LIFE-UNITS ARE BORN."
850 PRINT
860 PRINT" SOME PATTERNS OSCILLATE - THIS ONE IS CALLED THE BLINKER:"
870 PRINT
880 PRINT"   0                               0"
890 PRINT"   0   ->   0 0 0   ->   0"
900 PRINT"   0                               0"
910 PRINT
920 PRINT" NOTICE THAT THE TWO UNITS ON THE END OF THE BLINKER DIE; AND THAT"
930 PRINT" THE ONE IN THE CENTER LIVES BECAUSE IT HAD TWO NEIGHBORS. TWO NEW"
940 PRINT" LIFE-UNITS WERE BORN NEXT TO THE CENTER BECAUSE THEY HAD THREE"
950 PRINT" NEIGHBORS."
960 PRINT
970 PRINT" SOME PATTERNS ARE STABLE; LIKE THESE:"
980 PRINT
990 PRINT"           0"
1000 PRINT"   0 0           0 0           0 0 0"
1010 PRINT"   0 0           0 0           0 0 0"
1020 PRINT"           0 0"
1030 PRINT
1040 PRINT" THESE ARE CALLED THE BOX; THE LOOP; AND THE SNAKE."
1050 PRINT
1060 PRINT
1070 PRINT" SOME PATTERNS REPRODUCE THEMSELVES AND 'MOVE' ACROSS THE BOARD."
1080 PRINT" THIS ONE IS CALLED A GLIDER:"
1090 PRINT
1100 PRINT" -----"
1110 PRINT" :- 0 - - : ;- - - - : !- - - - : :- - - - : !- - - - : "
1111 PRINT" :- - 0 - : -> :0 - 0 - : ->:- - 0 - : ->:- 0 - - : ->:- - 0 - : "
1120 PRINT" :0 0 0 - : :- 0 0 - : !0 - 0 - : :- - 0 0 : :- - - 0 : "
1130 PRINT" :- - - - : :- 0 - - : !- 0 0 - : :- 0 0 - : :- 0 0 0 : "
1140 PRINT" -----"
1150 PRINT
1160 PRINT
1170 PRINT" NOTICE THAT THE GLIDER 'MOVES' ONE SQUARE DOWN AND ONE SQUARE"
1180 PRINT" TO THE RIGHT EVERY FOUR GENERATIONS; IT IS SAID TO MOVE AT ONE-"
1190 PRINT" QUARTER OF THE SQUARE-ROOT OF TWO TIMES THE SPEED OF LIGHT."
1200 PRINT
1210 PRINT
1220 PRINT" THE PLAY:"
1230 PRINT

```



```

1240 PRINT" MY GAME: 'LIFE', IS PLAYED ON A BOARD 11 x 30. EACH PLAYER HAS A"
1260 PRINT" UP FOR YOU BY THE COMPUTER AT THE START OF EACH GAME. EACH PLAYER"
1270 PRINT" TRIES TO DESTROY THE STABLE PATTERNS IN HIS OPPONENTS HOME REGION."
1280 PRINT" THE METHOD OF ATTACK IS VARIABLE. SOME TIMES GLIDERS OR SIMILAR"
1290 PRINT" PATTERNS ARE USED. SOMETIMES PATTERNS WHICH ARE KNOWN TO EXPAND ARE"
1300 PRINT" EMPLOYED. IN ANY CASE THE ONLY CHANCE THE PLAYERS WILL HAVE TO PLAN"
1310 PRINT" THEIR STRATEGY WILL BE AT THE OUTSET OF THE GAME. IT IS THE INITIAL"
1320 PRINT" PLAN OF THEIR WORLD WHICH WILL WIN OR LOSE THE BATTLE OF LIFE!"
1330 PRINT
1340 PRINT
1350 PRINT
1400 PRINT
1410 PRINT" DO YOU WANT A LISTING OF THE RULES "
1420 GOSUB 19000
1430 IF 0 = 0 THEN 2000
1440 PRINT
1450 PRINT
1460 PRINT" HERE ARE THE RULES:"
1470 PRINT
1480 PRINT" EACH PLAYER HAS A HOME REGION WHICH HE TRIES TO DEFEND. AT THE"
1490 PRINT" SAME TIME; HOWEVER; HIS PLAN MUST DESTROY HIS OPPONENTS HOME REGION."

1500 PRINT" THUS EACH PLAYER MUST SET UP HIS INITIAL PLAN OF HIS WORLD WITH"
1510 PRINT" A PHILOSOPHY OF DEFENSE; OFFENSE; OR ATTEMPT BOTH."
1520 PRINT
1530 PRINT" EACH PLAYER MAY USE UP TO THIRTY LIFE-UNITS IN HIS INITIAL"
1540 PRINT" PATTERN. THIS IS IN ADDITION TO THE HOME REGIONS WHICH I HAVE"
1550 PRINT" ALREADY SET UP FOR EACH PLAYER. A PLAYER MAY PUT THE LIFE-UNITS"
1560 PRINT" OF HIS INITIAL PATTERN ON HIS SIDE OF THE BOARD ONLY. FOR PLAYER"
1570 PRINT" ONE THIS IS FROM Y = 4 TO Y = 15; (THE REGION FROM Y = 1 TO Y = 3"
1580 PRINT" IS RESERVED FOR THE HOME REGION. DO NOT TRY TO PLACE ANY LIFE-UNITS"

1590 PRINT" IN THIS AREA!)"
1600 PRINT
1610 PRINT" THE BOARD INITIALLY LOOKS LIKE THIS:"
1620 PRINT
1630 GOSUB 15000
1640 PRINT
1650 LET Z = 1
1660 PRINT
1670 PRINT" PLAYER ONE HAS THE BOTTOM OF THE BOARD AS IT WAS PRINTED OUT."
1680 PRINT" PLAYER TWO (WHO MIGHT BE THE MACHINE) HAS THE TOP OF THE BOARD."
1690 PRINT
1700 PRINT" IF YOU WANT TO PLAY AGAINST THE COMPUTER; TYPE 'MACHINE' WHEN I"
1710 PRINT" ASK YOU FOR PLAYER TWO'S NAME. "
1720 PRINT
2000 PRINT
2010 PRINT

```

```

2020 PRINT" WHAT IS PLAYER ONE'S NAME ";
2030 INPUT A$
2040 GOSUB 14000
2050 PRINT
2060 PRINT" WHAT IS PLAYER TWO'S NAME ";
2070 INPUT B$
2080 GOSUB 14000
2090 PRINT
2100 IF Z = 1 THEN 2170
2110 PRINT
2120 PRINT" THE BOARD INITIALLY LOOKS LIKE THIS:"
2130 PRINT
2140 GOSUB 15000
2150 PRINT
2160 PRINT
2170 PRINT" ATTENTION ";A$;
2175 P = 1
2180 IF T=0 THEN 2210
2190 PRINT"; INPUT YOUR COORDINATES AS BEFORE."
2200 GO TO 2340
2210 PRINT"; PLEASE GIVE ME YOUR INITIAL PATTERN OF LIFE-UNITS."
2220 PRINT" IN ORDER TO PLACE A LIFE-UNIT AT POINT A;B (CARTESIAN COORDINATES)"
2230 PRINT" TYPE THE VALUES IN SEPARATED BY A COMMA. IN OTHER WORDS -- TYPE THE"
2240 PRINT" X COORDINATE; A. FOLLOW IT WITH A COMMA. THEN TYPE THE Y COOR-"
2250 PRINT" DINATE; B; FOLLOWED BY A CARRIAGE RETURN."
2260 PRINT" WHEN YOU HAVE FINISHED YOUR INITIAL PATTERN; TYPE IN 0;0."
2270 PRINT" ALSO; AFTER YOUR 30TH LIFE-UNIT IS POSITIONED; I WILL ASSUME"
2280 PRINT" THAT YOU ARE FINISHED; AND I WILL GO ON..."
2290 PRINT
2300 PRINT" I AM NOW READY FOR INPUT OF YOUR LIFE-UNITS. EACH TIME I TYPE A '?'"
2310 PRINT" PLEASE TYPE IN ONE PAIR OF COORDINATES."
0002320 T = 1
0002330 REM
0002340 X = 1
0002350 INPUT M,N
0002360 IF P = 1 GOTO 2400
0002370 C = 16
0002380 D = 30
0002390 GO TO 2420
0002400 C = 4
0002410 D = 15
0002420 IF M = 0 THEN 2500
0002425 IF N < C THEN 2450
0002430 IF N > D THEN 2450
0002435 IF M > 11 THEN 2450
0002440 IF M < 0 THEN 2450
0002445 GO TO 2470
0002450 PRINT"!! YOU CAN'T PUT A LIFE-UNIT THERE !!"
0002455 PRINT" YOU ARE LIMITED TO Y FROM ";C;" TO ";D;" AND"

```

```

0002460 PRINT" X FROM 1 TO 11 !"
0002465 PRINT" TRY AGAIN."
0002467 GO TO 2350
0002470 LET A(N+1,N+1)=1
0002490 X=X+1
0002490 IF X <= 30 THEN 2350
0002500 IF P = 2 THEN 2670
0002510 PRINT" ";A$;" IS SET UP !"
0002520 F=2
0002530 IF B$ = "MACHINE" THEN 2580
0002540 IF B$ = "MACHINE" THEN 2580
0002550 PRINT
0002560 PRINT" ATTENTION ";B$;
0002570 GO TO 2180
0002580 M9=1
0002585 R = RND(-1)
0002590 GOSUB 13000
0002600 IF R < .333 THEN 2640
0002610 GOSUB 13000
0002620 IF R < .666 THEN 2640
0002630 GOSUB 13000
0002640 MAT A = A + B
0002645 PRINT
0002650 PRINT" THE MACHINE IS READY!"
0002660 GO TO 2680
0002670 PRINT" EVERYONE IS SET UP!"
0002680 PRINT" THE BATTLE BEGINS !"
0002685 PRINT
0002690 PRINT" HERE'S THE COMPLETE BOARD:"
0002700 PRINT
0002710 GOSUB 15000
0002720 PRINT
0002730 PRINT" How many GENERATIONS DO YOU WANT "
0002740 PRINT" BEFORE PRINTING THE BOARD AGAIN ";
0002750 REM
0002760 GO TO 2780
0002770 PRINT" How Many GENERATIONS Now ";
0002780 INPUT E
0002785 E9 = E9 + E
0002790 IF E = 0 THEN 3000
0002800 REM GENERATION CHANGE LOGIC
0002810 FOR H = E TO 1 STEP -1
0002815 MAT B = ZER
0002820 FOR J = 2 TO 31
0002830 FOR I = 2 TO 12
0002840 LET S=A(I-1,J)+A(I-1,J+1)+A(I,J+1)+A(I+1,J+1)
0002850 LET S=S+A(I+1,J)+A(I+1,J-1)+A(I,J-1)+A(I-1,J-1)
0002870 IF S = 2 THEN 2920
0002880 IF S = 3 THEN 2910
0002890 LET B(I,J) = -A(I,J)
0002900 GO TO 2920

```

```

0002910         LET B(I,J) = 1 - A(I,J)
0002920         NEXT I
0002930     NEXT J
0002940     MAT A = A+B
0002950 NEXT H
0002960 GOSUB 15000
0002970 GO TO 2770
0002980 REM
0002990 REM *** EXIT OR AGAIN ***
0003000 PRINT
0003010 PRINT
0003020 PRINT
0003030 PRINT" THANK YOU FOR PLAYING 'LIFE' ";A$
0003040 IF M9 = 1 THEN 3060
0003050 PRINT" YOU TOO ";B$
0003060 PRINT"

"
0003070 PRINT" WOULD YOU LIKE TO START ANOTHER GAME ";
0003080 GOSUB 19000
0003090 IF a = 1 THEN 3150
0003100 PRINT"

"
0003104 PRINT
0003120 PRINT
0003130 STOP

0003150 GO TO 100
0013000 REM
0013010 REM *** ROUTINE TO SET UP MACHINE'S LIFE-UNITS ***
0013020 REM
0013030 MAT B = ZER
0013040 FOR I = 1 TO 29
0013050     READ U,U
0013060     B(U+1,U+1) = 1
0013070 NEXT I
0013080 RETURN
0013090 REM
0013100 RETURN
0014000 REM
0014010 REM ROUTINE TO CHECK 'STRAIGHTNESS' OF AN INPUT NAME
0014020 REM
0014030 REM
0014040 REM
0014050 REM
0014060 REM
0014070 REM
0014080 REM

```

```

0014090 RETURN
0015000 REM
0015010 REM*** BOARD PRINTING ROUTINE ***
0015020 REM
0015030 PRINT
0015040 PRINT
0015060 PRINT"      :-----:"
0015070 FOR J = 31 TO 2 STEP -1
0015080     IF J NE 31 THEN 15110
0015090     PRINT" 30: ";
0015100     GO TO 15300
0015110     IF J NE 26 THEN 15140
0015120     PRINT" 25: ";
0015130     GO TO 15300
0015140     IF J NE 21 THEN 15170
0015150     PRINT" 20: ";
0015160     GO TO 15300
0015170     IF J NE 16 THEN 15200
0015180     PRINT" 15: ";
0015190     GO TO 15300
0015200     IF J NE 11 THEN 15230
0015210     PRINT" 10: ";
0015220     GO TO 15300
0015230     IF J NE 6 THEN 15260
0015240     PRINT"  5: ";
0015250     GO TO 15300
0015260     IF J NE 2 THEN 15290
0015270     PRINT"  1: ";
0015280     GO TO 15300
0015290     PRINT"      : ";
0015300     FOR I = 2 TO 12
0015310         IF A(I,J) = 1 THEN 15340
0015320         PRINT"- ";
0015330         GO TO 15350
0015340         PRINT"0 ";
0015350     NEXT I
0015360     PRINT": "
0015370 NEXT J
0015380 PRINT"      :-----:"
0015390 PRINT"      1   3   5   7   9  11"
0015400 PRINT
0015410 PRINT
0015420 RETURN
19000 REM **** YES OR NO FETCH ROUTINE ****
19010 INPUT Q$
19020 IF Q$ = "YES" THEN 19080
19030 IF Q$ = "yes" THEN 19080
19040 IF Q$ = "NO" THEN 19100
19050 IF Q$ = "no" THEN 19100
19060 PRINT" !!! PLEASE, ANSWER YES OR NO !!!"
19065 PRINT"      ANSWER: ";
19070 GO TO 19010
19080 Q = 1
19090 RETURN
19100 Q = 0
19110 RETURN
19120 REM
19130 STOP
19140 END

```

BECOME A TWO-DIMENSIONAL TYRANT

EXPERIENCE THE JOY OF CONFLICT AS YOU DETERMINE THE DESTINY
OF YOUR WORLD OF LIFE-UNITS

WATCH HELPLESSLY AS YOUR LIFE-UNITS AIMLESSLY BLUNDER ABOUT!

PLAY THE 'BATTLE OF LIFE'!

DO YOU WANT A BASIC EXPLANATION OF THE GAME ?Y
!!! PLEASE, ANSWER YES OR NO !!!
ANSWER ?YES

THE GAME 'BATTLE OF LIFE' IS BASED ON THE GAME LIFE.

LIFE-UNITS ARE COUNTERS ON AN INFINITE GRID. THE PRESENCE OF A LIFE-UNIT IS INDICATED AS A ZERO ON THE PRINTED GAME BOARD. ON THE BOARD EACH LIFE-UNIT IS SURROUNDED BY A MAXIMUM OF EIGHT NEIGHBORS. (ALL ADJACENT UNITS ARE COUNTED, INCLUDING DIAGONAL NEIGHBORS)

THE GAME PROCEEDS FROM ONE GENERATION TO ANOTHER. THE LIFE-UNITS THEMSELVES DO NOT MOVE, THEY WILL EITHER LIVE OR DIE. THE CURRENT POPULATION OF AN AREA WILL DETERMINE THE POPULATION OF THE THE NEXT GENERATION. EXISTING LIFE-UNITS MAY DIE, OR NEW ONES MAY BE BORN IN EMPTY SPACES, ACCORDING TO THE FOLLOWING RULES:

DEATH - A LIFE-UNIT WILL DIE IF IT HAS LESS THAN TWO NEIGHBORS... (FROM ISOLATION); OR MORE THAN THREE NEIGHBORS... (FROM OVERCROWDING).

LIFE - A LIFE-UNIT WILL LIVE IF IT HAS TWO OR THREE NEIGHBORS.

BIRTH - A NEW LIFE-UNIT WILL BE BORN IN AN EMPTY SPACE THERE ARE EXACTLY THREE NEIGHBORING LIFE-UNITS.

EXAMPLES:

0 0

THE ABOVE LIFE-UNITS WILL BOTH DIE BECAUSE EACH HAS ONLY ONE NEIGHBOR.

0 0
0 0
0 0

TWO OF THE ABOVE LIFE-UNITS WILL DIE FROM OVERCROWDING. TWO NEW LIFE-UNITS WILL BE BORN. THE NEXT GENERATION OF THE ABOVE GROUP WILL BE:

```

  0 0
0   0
  0 0

```

THIS PATTERN IS STABLE, EACH LIFE-UNIT HAS TWO NEIGHBORS--SO IT LIVES --- AND NO SPACE HAS EXACTLY THREE NEIGHBORS--SO NO NEW LIFE-UNITS ARE BORN.

SOME PATTERNS OSCILLATE - THIS ONE IS CALLED THE BLINKER:

```

  0           0
  0   ->   0 0 0   ->   0
  0           0

```

NOTICE THAT THE TWO UNITS ON THE END OF THE BLINKER DIE, AND THAT THE ONE IN THE CENTER LIVES BECAUSE IT HAD TWO NEIGHBORS. TWO NEW LIFE-UNITS WERE BORN NEXT TO THE CENTER BECAUSE THEY HAD THREE NEIGHBORS.

SOME PATTERNS ARE STABLE, LIKE THESE:

```

           0
    0 0      0 0      0 0 0
    0 0      0 0      0 0 0
           0 0

```

THESE ARE CALLED THE BOX, THE LOAF, AND THE SNAKE.

SOME PATTERNS REPRODUCE THEMSELVES AND 'MOVE' ACROSS THE BOARD. THIS ONE IS CALLED A GLIDER:

```

-----
:- 0 - - :   :- - - - :   :- - - - :   :- - - - :   :- - - - :
:- - 0 - : -> :- 0 - 0 - : -> :- - 0 - : -> :- 0 - - : -> :- - 0 - :
:0 0 0 - :   :- 0 0 - :   :- 0 - 0 - :   :- - 0 0 :   :- - - 0 :
:- - - - :   :- 0 - - :   :- 0 0 - :   :- 0 0 - :   :- 0 0 0 :
-----

```

NOTICE THAT THE GLIDER 'MOVES' ONE SQUARE DOWN AND ONE SQUARE TO THE RIGHT EVERY FOUR GENERATIONS; IT IS SAID TO MOVE AT ONE-QUARTER OF THE SQUARE-ROOT OF TWO TIMES THE SPEED OF LIGHT.

THE PLAY:

MY GAME, 'LIFE', IS PLAYED ON A BOARD 11 X 30. EACH PLAYER HAS A UP FOR YOU BY THE COMPUTER AT THE START OF EACH GAME. EACH PLAYER TRIES TO DESTROY THE STABLE PATTERNS IN HIS OPPONENTS HOME REGION.

THE METHOD OF ATTACK IS VARIABLE. SOME TIMES GLIDERS OR SIMILAR PATTERNS ARE USED. SOMETIMES PATTERNS WHICH ARE KNOWN TO EXPAND ARE EMPLOYED. IN ANY CASE THE ONLY CHANCE THE PLAYERS WILL HAVE TO PLAY THEIR STRATEGY WILL BE AT THE OUTSET OF THE GAME. IT IS THE INITIAL PLAY OF THEIR WORLD WHICH WILL WIN OR LOSE THE BATTLE OF LIFE!

DO YOU WANT A LISTING OF THE RULES ?YES

HERE ARE THE RULES:

EACH PLAYER HAS A HOME REGION WHICH HE TRIES TO DEFEND. AT THE SAME TIME, HOWEVER, HIS PLAN MUST DESTROY HIS OPPONENTS HOME REGION. THUS EACH PLAYER MUST SET UP HIS INITIAL PLAN OF HIS WORLD WITH A PHILOSOPHY OF DEFENSE, OFFENSE, OR ATTEMPT BOTH.

EACH PLAYER MAY USE UP TO THIRTY LIFE-UNITS IN HIS INITIAL PATTERN. THIS IS IN ADDITION TO THE HOME REGIONS WHICH I HAVE ALREADY SET UP FOR EACH PLAYER. A PLAYER MAY PUT THE LIFE-UNITS OF HIS INITIAL PATTERN ON HIS SIDE OF THE BOARD ONLY. FOR PLAYER ONE THIS IS FROM Y = 4 TO Y = 15. (THE REGION FROM Y = 1 TO Y = 3 IS RESERVED FOR THE HOME REGION. DO NOT TRY TO PLACE ANY LIFE-UNITS IN THIS AREA)

THE BOARD INITIALLY LOOKS LIKE THIS:


```

:-----:
30: 0 0 - - - 0 0 - 0 0 - :
: 0 - 0 - 0 - 0 - 0 - 0 :
: - 0 0 - 0 0 - - - 0 0 :
: - - - - - - - - - - :
: - - - - - - - - - - :
25: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
20: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
15: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
10: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
: - - - - - - - - - - :
5: - - - - - - - - - - :
: - - - - - - - - - - :
: 0 0 - - - 0 0 - 0 0 - :
: 0 - 0 - 0 - 0 - 0 - 0 :
1: - 0 0 - 0 0 - - - 0 0 :
:-----:
      1   3   5   7   9  11

```

PLAYER ONE HAS THE BOTTOM OF THE BOARD AS IT WAS PRINTED OUT.
 PLAYER TWO (WHO MIGHT BE THE MACHINE) HAS THE TOP OF THE BOARD.

IF YOU WANT TO PLAY AGAINST THE COMPUTER, TYPE 'MACHINE' WHEN I
 ASK YOU FOR PLAYER TWO'S NAME.

WHAT IS PLAYER ONE'S NAME ?

LOANS:

DESCRIPTION

This program computes annuities for both payments and withdrawals for loans, mortgages, savings, etc.

USERS

Householders, individuals who have checking or savings accounts or have outstanding loans or are contemplating any of the above will be able to put this program to good use.

INSTRUCTIONS

Upon execution the program will ask if you want instructions. LOANS is fully documented and will prompt for all necessary data while the program is running.

LIMITATIONS

Line 100 contains a DEF FNX () statement; this function is used throughout the program. In line 1200 LOG () and 1260 ABS () statements appear. The source code will require 6K Bytes of memory for storage. Execution will require an additional 1000 Bytes of memory for table storage for a total of 7K Bytes for execution.

LOANS

```

30 PRINT "WANT INSTRUCTIONS FIRST (1=YES, 2=NO)";
40 INPUT L
50 IF L<>1 THEN 70
60 GOSUB 600
70 LET S=0
80 PRINT
90 PRINT "WHICH ANNUITY TYPE (1=PAYMENT, 2=WITHDRAWAL)";
100 DEF FNA(U)=.01*INT(100*U+.5)
110 INPUT A
120 GOTO 170
130 IF A=1 THEN 160
140 LET A=1
150 GOTO 170
160 LET A=2
170 IF A=1 THEN 220
180 IF A=2 THEN 1050
190 PRINT
200 PRINT " ANSWER 1 OR 2 PLEASE."
210 GOTO 80
220 IF S=1 THEN 260
230 PRINT
240 PRINT "WHICH VARIABLE IS UNKNOWN (1=N, 2=A, 3=I, 4=R)";
250 INPUT D
260 IF D=1 THEN 330
270 IF D=2 THEN 410
280 IF D=3 THEN 490
290 IF D=4 THEN 520
300 PRINT
310 PRINT " ANSWER 1, 2, 3, OR 4 PLEASE."
320 GOTO 230
330 PRINT
340 PRINT "WHAT ARE A($), I(PCT), R($)";
350 INPUT A1, I1, R1
360 LET I1=I1/100
370 LET N1=LOG((A1*I1/R1)+1)/LOG(1+I1)
380 PRINT
390 PRINT "NUMBER OF PERIODS = N =" ; N1
400 GOTO 920
410 PRINT
420 PRINT "WHAT ARE N(INTEGER), I(PCT), R($)";
430 INPUT N1, I1, R1
440 LET I1=I1/100
450 LET A1=R1*((1+I1)N1-1)/I1
460 PRINT
470 PRINT "AMOUNT AT END OF N PERIODS = A =" ; A1
480 GOTO 920

```

```

490 PRINT
500 PRINT "WHAT ARE N(INTEGER),A($),R($)";
510 GOTO 1700
520 PRINT
530 PRINT "WHAT ARE N(INTEGER),A($),I(PCT)";
540 INPUT N1,A1,I1
550 LET I1=I1/100
560 LET P1=A1*I1/((1+I1)↑N1-1)
570 PRINT
580 PRINT "PAYMENT EACH PERIOD = R ="!R1
590 GOTO 920
600 PRINT
610 PRINT "THIS PROGRAM PERFORMS CALCULATIONS FOR"
620 PRINT "BOTH PAYMENT AND WITHDRAWAL ANNUITIES."
630 PRINT
640 PRINT "VARIABLES ARE DENOTED AS FOLLOWS:"
650 PRINT
660 PRINT "  N = NUMBER OF PERIODS"
670 PRINT "  P = ORIGINAL PRINCIPAL AMOUNT"
680 PRINT "  A = TOTAL AMOUNT AT END OF N PERIODS"
690 PRINT "  I = INTEREST RATE PER PERIOD, IN PERCENT"
700 PRINT "  R = AMOUNT OF PAYMENT/WITHDRAWAL EACH PERIOD"
710 PRINT
720 PRINT "FOR A PAYMENT ANNUITY, YOU MAY GIVE ANY"
730 PRINT "  THREE OF N,A,I,R AND FIND THE FOURTH."
740 PRINT
750 PRINT "FOR A WITHDRAWAL ANNUITY, YOU GIVE ANY"
760 PRINT "  THREE OF N,P,I,R AND FIND THE FOURTH."
770 PRINT
780 PRINT "NOTE: FOR LOAN OR MORTGAGE, USE THE"
790 PRINT "  WITHDRAWAL ANNUITY OPTION."
800 PRINT
810 PRINT "AFTER EACH CASE, YOU MAY CHOOSE ONE"
820 PRINT "OF THE FOLLOWING ALTERNATIVES:"
830 PRINT
840 PRINT "  1 = ANOTHER CASE, SAME UNKNOWN VARIABLE"
850 PRINT "  2 = ANOTHER CASE, DIFFERENT UNKNOWN"
860 PRINT "  3 = ANOTHER CASE, OTHER TYPE OF ANNUITY"
870 PRINT "  4 = TOTAL INTEREST PAID OVER N PERIODS"
880 PRINT "  5 = TABLE OF WITHDRAWALS, PRINCIPAL, AND INTEREST"
890 PRINT "  6 = STOP THE PROGRAM"
900 PRINT
910 RETURN
920 PRINT
930 PRINT "ANOTHER CASE (TYPE CODE NUMBER)";
940 INPUT S
950 IF S=1 THEN 170
960 IF S=2 THEN 170
970 IF S=6 THEN 9999
980 IF S=3 THEN 130
990 IF A<>1 THEN 1030

```

```

1000 PRINT
1010 PRINT "ALTERNATIVE NOT APPLICABLE TO PAYMENT ANNUITIES."
1020 GOTO 920
1030 IF S=4 THEN 1240
1040 IF S=5 THEN 1240
1050 IF S=1 THEN 1090
1060 PRINT
1070 PRINT "WHICH VARIABLE IS UNKNOWN (1=N,2=P,3=I,4=R)";
1080 INPUT A1
1090 IF A1=1 THEN 1160
1100 IF A1=2 THEN 1660
1110 IF A1=3 THEN 1740
1120 IF A1=4 THEN 2090
1130 PRINT
1140 PRINT " ANSWER 1,2,3, OR 4 PLEASE."
1150 GOTO 1030
1160 PRINT
1170 PRINT "WHAT ARE P($),I(PCT),R($)";
1180 INPUT P1,I1,R1
1190 LET I1=I1/100
1200 LET N1=LOG(R1/(R1-P1*I1))/LOG(1+I1)
1210 PRINT
1220 PRINT "NUMBER OF PERIODS = N =";N1
1230 GOTO 920
1240 LET N1=N1+.000001
1250 LET P2=INT(N1)-N1
1260 IF ABS(P2)>.000001 THEN 1290
1270 LET N1=N1-1
1280 GOTO 1300
1290 LET N1=INT(N1)
1300 LET P2=1
1310 IF S=4 THEN 1380
1320 IF S=5 THEN 1330
1330 LET P2=0
1340 PRINT
1350 PRINT "PERIOD","PRINCIPAL","INTEREST","PRINC BAL","INT TO DATE"
1360 PRINT
1370 PRINT " @"," ", " ", " ",P1
1380 LET X=0
1390 LET X9=P1
1400 LET Y=0
1410 FOR S=1 TO N1
1420 LET A3=(R1-P1*I1)*(1+I1)(S-1)
1430 LET A2=FNA(A3)
1440 LET C=R1-A3
1450 LET B=FNA(C)
1460 LET X=X+A3
1470 LET X9=X9-A3
1480 LET Y=FNA(Y+C)
1490 IF P2 <>0 THEN 1510
1500 PRINT S,A2,B,X9,Y

```

```

1510 NEXT S
1520 LET S=N1+1
1530 LET A3=P1-X
1540 LET A2=FNA(A3)
1550 LET B1=I1*A3
1560 LET B=FNA(B1)
1570 LET X=X+A3
1580 LET X9=0
1590 LET Y=Y+B1
1600 IF P2=0 THEN 1640
1610 PRINT
1620 PRINT "TOTAL INTEREST PAID =" ; Y
1630 GOTO 920
1640 PRINT S, A2, B, X9, Y
1650 GOTO 920
1660 PRINT
1670 PRINT "WHAT ARE N(INTEGER), I(PCT), R($) ";
1680 INPUT N1, I1, R1
1690 LET I1=I1/100
1700 LET P1=R1*((1+I1)N1-1)/(I1*(1+I1)N1)
1710 PRINT
1720 PRINT "ORIGINAL PRINCIPAL AMOUNT IS" ; P1
1730 GOTO 920
1740 PRINT
1750 PRINT "WHAT ARE N(INTEGER), P($), R($) ";
1760 INPUT N1, P1, R1
1770 PRINT
1780 IF N1*R1<=P1 THEN 2170
1790 LET I1=.1
1800 LET I2=0
1810 LET I3=1
1820 IF A<>1 THEN 1850
1830 LET P3=R1*((1+I1)N1-1)/I1
1840 GOTO 1860
1850 LET P3=R1*((1+I1)N1-1)/(I1*(1+I1)N1)
1860 LET P4=ABS(P1-P3)
1870 IF P4>0.001 THEN 1900
1880 PRINT "INTEREST PER PERIOD (PCT) = I =" ; 100*I1
1890 GOTO 920
1900 IF P4/P1>.3 THEN 2010
1910 IF A=1 THEN 1940
1920 IF P3<P1 THEN 1930
1930 GOTO 1950
1940 IF P3>P1 THEN 1930
1950 LET I2=I1
1960 LET I1=(I1+I3)/2
1970 GOTO 1820
1980 LET I3=I1
1990 LET I1=(I1+I2)/2
2000 GOTO 1820
2010 IF A=1 THEN 2040

```

```

2020 IF P3<P1 THEN 2060
2030 GOTO 1950
2040 IF P3>P1 THEN 2060
2050 GOTO 1950
2060 LET I3=I1
2070 LET I1=(I1+I2)/4
2080 GOTO 1820
2090 PRINT
2100 PRINT "WHAT ARE N(INTEGER), P($), I(PCT)";
2110 INPUT N1,P1,I1
2120 LET I1=I1/100
2130 LET R1=P1*I1*(1+I1)N1/((1+I1)N1-1)
2140 PRINT
2150 PRINT "WITHDRAWAL EACH PERIOD = R ="(R1)
2160 GOTO 920
2170 PRINT"THE PRINCIPAL [P] IS NOT RECOVERED; [I] MUST BE POSITIVE."
2180 GO TO 1740
9999 END

```



SAMPLE PROBLEM #1

TO DETERMINE THE AMOUNT ON HAND ASSUMING DEPOSITS OF \$35.00 A MONTH;
INTEREST PAID IS 4-7/8% A YEAR, COMPOUNDED QUARTERLY.
THE VARIABLES ARE CONVERTED TO A QUARTERLY BASIS AND ENTERED AS SHOWN

$N = 3 * 4 = 12$ PERIODS
A = VARIABLE OF INTEREST
 $I = 4.875/4 = 1.21875$ /PERIOD
 $R = 35 * 3 = 105$ /PERIOD

SOLUTION

*RUN

WANT INSTRUCTIONS FIRST (1=YES, 2=NO) ?2
WHICH ANNUITY TYPE (1=PAYMENT, 2=WITHDRAWAL) ?1
WHICH VARIABLE IS UNKNOWN (1=N, 2=A, 3=I, 4=R) ?2
WHAT ARE N(INTEGER), I(PCT), R(\$)?16,1.21875,105.00
AMOUNT AT END OF N PERIODS = A = 1842.652

SAMPLE PROBLEM #2

TO DETERMINE THE MONTHLY PRINCIPAL AND INTEREST PAYMENTS ON A
\$20,900 MORTGAGE.
PERIOD OF TIME IS 30 YEARS.
INTEREST RATE IS 5-1/2% PER YEAR.
THE VARIABLES ARE CONVERTED TO A MONTHLY BASIS AND ENTERED AS
SHOWN.

$N = 30 * 12 = 360$ PERIODS
P = 20900
 $I = 5.5/12 = .4583333333$ /PERIOD
R = VARIABLE OF INTEREST

SOLUTION

*RUN

WANT INSTRUCTIONS FIRST (1=YES, 2=NO) ?2
 WHICH ANNUITY TYPE (1=PAYMENT, 2=WITHDRAWAL) ?2
 WHICH VARIABLE IS UNKNOWN (1=N, 2=P, 3=I, 4=R) ?4
 WHAT ARE N(INTEGER), P(\$), I(PCT) ?360,20900,.450333333
 WITHDRAWAL EACH PERIOD = R = 118.6679
 ANOTHER CASE (TYPE CODE NUMBER) ?4
 TOTAL INTEREST PAID = 21820.45
 ANOTHER CASE (TYPE CODE NUMBER) ?5

PERIOD	PRINCIPAL	INTEREST	PRINC BAL	INT TO DATE
0			20900	
1	22.89	95.79	20877.12	95.79
2	22.98	95.69	20854.14	191.48
3	23.09	95.58	20831.06	287.06
4	23.19	95.48	20807.86	382.54
5	23.30	95.37	20784.56	477.91
6	23.41	95.26	20761.16	573.17
7	23.51	95.16	20737.65	668.33
8	23.62	95.05	20714.03	763.38
9	23.73	94.94	20690.30	858.32
10	23.84	94.83	20666.46	953.15
11	23.95	94.72	20642.51	1047.87
12	24.06	94.61	20618.46	1142.48
13	24.17	94.50	20594.29	1236.98
14	24.28	94.39	20570.01	1331.37
15	24.39	94.28	20545.62	1425.65
16	24.50	94.17	20521.12	1519.82
17	24.61	94.06	20496.51	1613.88

MAZES:

DESCRIPTION

This program generates uniquely different mazes in two dimensions. The maximum size of a maze is 23 x 25. Each maze is distinctly different from the preceding one and has only ONE solution.

USERS

Puzzle and riddle fans will enjoy this game. The complexity of the mazes generated is varied according to the dimensions but all should provide fair competition for the initiated as well as the uninitiated player.

INSTRUCTIONS

Type RUN and the program will print out a full list of instructions and then ask you for the size of the maze you would like to construct. The greater the dimensions of the maze the more complex is its' solution.

LIMITATIONS

Mazes uses two dimensional arrays to generate the patterns. The source code requires 6K Bytes for storage. With the DIM statement in line 240 set at (23, 25) the program will require 18K Bytes of memory for execution. If this DIM statement is reduced the program will execute in less memory space but the mazes generated will be correspondingly less complex.

MAZES

```

120REM
130REM DESCRIPTION -- MAZES CONSTRUCTS A MAZE OF ANY DIMENSIONS
140REM THE USER WISHES UP TO 23 BY 23. EACH
150REM MAZE IS GUARANTEED TO HAVE ONE AND ONLY
160REM ONE SOLUTION.
170REM          ****          ****          ****
180REM
190REM INSTRUCTIONS--TYPE 'RUN' AND FOLLOW INSTRUCTIONS
200REM
210REM*****
220REM
240 DIM W(23,23), U(23,23)
250 PRINT "THIS PROGRAM WILL PRINT OUT A DIFFERENT MAZE EVERY TIME IT"
260 PRINT "IS RUN AND GUARANTEES ONLY ONE PATH THROUGH. YOU CAN CHOOSE"
270 PRINT "THE DIMENSIONS OF THE MAZE. I.E. THE NUMBER OF SQUARES WIDE"
280 PRINT "AND THE NUMBER OF SQUARES LONG. A 23 BY 23 MAZE IS THE "
290 PRINT "MAXIMUM, BUT ANY DIMENSIONS UP TO THESE LIMITS ARE OK."
300 PRINT
310 PRINT "WHAT ARE YOUR WIDTH AND LENGTH";
320 INPUT H,U
330 PRINT
340 IF H<>1 THEN 390
350 IF U<>1 THEN 390
360 PRINT "MEANINGLESS DIMENSIONS, TRY AGAIN."
370 PRINT
380 GO TO 310
390 PRINT
400 PRINT
410 LET Q = 0
420 LET Z = 0
430REM
440REM
450 REM          STEPS 1900 THRU 3200 PICK THE
460 REM          SQUARE IN WHICH TO START.
470REM
480REM
490 LET X = INT(RND(X)*H+1)
500 FOR I = 1 TO H
510 IF I = X THEN 540
520 PRINT ":--";
530 GOTO 550
540 PRINT ": ";
550 NEXT I
560 PRINT ":"
570 LET C = 1

```

```

580 LET W(X,1) = 0
590 LET C = C+1
600 LET R = X
610 LET S = 1
620 GOTO 900
630REM
640REM
650 REM
660 REM
670 REM
680 REM
690 REM
700 REM
710REM
720REM
730 IF R <> H THEN 810
740 IF S <> U THEN 780
750 LET R = 1
760 LET S = 1
770 GOTO 890
780 LET R = 1
790 LET S = S+1
800 GOTO 890
810 LET R = R+1
820REM
830REM
840 REM
850 REM
860 REM
870REM
880REM
890 IF W(R,S) = 0 THEN 730
900 IF R-1 = 0 THEN 1330
910 IF W(R-1,S) <> 0 THEN 1330
920 IF S-1 = 0 THEN 1100
930 IF W(R,S-1) <> 0 THEN 1100
940 IF R = H THEN 990
950 IF W(R+1,S) <> 0 THEN 990
960 LET X = INT(RND(X)*3+1)
970 REM
980 ON X GOTO 1850,1920,1990
990 IF S <> U THEN 1030
1000 IF Z = 1 THEN 1070
1010 LET Q = 1
1020 GOTO 1040
1030 IF W(R,S+1) <> 0 THEN 1070
1040 LET X = INT(RND(X)*3+1)
1050 REM
1060 ON X GOTO 1850,1920,2090
1070 LET X = INT(RND(X)*2+1)
1080 REM

```

STEPS 4900 THRU 5700 LOOK FOR THE SQUARE IN WHICH TO START A NEW PATH WHEN THE OLD ONE GETS BLOCKED. THE LINES ARE SCANNED FOR A SQUARE WHICH HAS ALREADY BEEN USED BUT IS ADJACENT TO AN OPEN SQUARE.

STEPS 6500 THRU 13000 CHECK TO SEE WHICH ADJACENT SQUARES ARE OPEN FOR THE PATH TO FOLLOW.

LEFT, UP, RIGHT

LEFT, UP, DOWN

LEFT, UP

```

1090 ON X GOTO 1850,1920
1100 IF R = H THEN 1230
1110 IF W(R+1,S) <> 0 THEN 1230
1120 IF S<> U THEN 1160
1130 IF Z = 1 THEN 1200
1140 LET Q = 1
1150 GOTO 1170
1160 IF W(R,S+1) <> 0 THEN 1200
1170 LET X = INT(RND(X)*3+1)
1180 REM LEFT,RIGHT,DOWN
1190 ON X GOTO 1850,1990,2090
1200 LET X = INT(RND(X)*2+1)
1210 REM LEFT,RIGHT
1220 ON X GOTO 1850,1990
1230 IF S <> U THEN 1270
1240 IF Z = 1 THEN 1320
1250 LET Q = 1
1260 GOTO 1280
1270 IF W(R,S+1) <> 0 THEN 1320
1280 LET X = INT(RND(X)*2+1)
1290 REM LEFT,DOWN
1300 ON X GOTO 1850,2090
1310 REM LEFT
1320 GOTO 1850
1330 IF S-1 = 0 THEN 1580
1340 IF W(R,S-1) <> 0 THEN 1580
1350 IF R=H THEN 1480
1360 IF W(R+1,S) <> 0 THEN 1480
1370 IF S<> U THEN 1410
1380 IF Z = 1 THEN 1450
1390 LET Q = 1
1400 GOTO 1420
1410 IF W(R,S+1) <> 0 THEN 1450
1420 LET X = INT(RND(X)*3+1)
1430 REM UP,RIGHT,DOWN
1440 ON X GOTO 1920,1990,2090
1450 LET X = INT(RND(X)*2+1)
1460 REM UP,RIGHT
1470 ON X GOTO 1920,1990
1480 IF S <> U THEN 1520
1490 IF Z = 1 THEN 1570
1500 LET Q = 1
1510 GOTO 1530
1520 IF W(R,S+1) <> 0 THEN 1570
1530 LET X = INT(RND(X)*2+1)
1540 REM UP,DOWN
1550 ON X GOTO 1920,2090
1560 REM UP
1570 GOTO 1920
1580 IF R = H THEN 1700
1590 IF W(R+1,S) <> 0 THEN 1700

```

```

1600 IF S <> U THEN 1640
1610 IF Z = 1 THEN 1690
1620 LET Q = 1
1630 GOTO 1650
1640 IF W(R,S+1) <> 0 THEN 1690
1650 LET X = INT(RND(X)*2+1)
1660 REM RIGHT,DOWN
1670 ON X GOTO 1990,2090
1680 REM RIGHT
1690 GOTO 1990
1700 IF S <> U THEN 1740
1710 IF Z = 1 THEN 1780
1720 LET Q = 1
1730 GOTO 1760
1740 IF W(R,S+1) <> 0 THEN 1780
1750 REM DOWN
1760 GOTO 2090
1770 REM BLOCKED
1780 GOTO 2290
1790REM
1800REM
1810 REM STEPS 13900 THRU 18100 MAKE A RANDOM
1820 REM CHOICE OF AVAILABLE PATHS.
1830REM
1840REM
1850 LET W(R-1,S) = C
1860 LET C = C+1
1870 LET V(R-1,S) = 2
1880 LET R = R-1
1890 IF C = H*U+1 THEN 2310
1900 LET Q = 0
1910 GOTO 900
1920 LET W(R,S-1) = C
1930 LET C = C+1
1940 LET V(R,S-1) = 1
1950 LET S = S-1
1960 IF C = H*U+1 THEN 2310
1970 LET Q = 0
1980 GOTO 900
1990 LET W(R+1,S) = C
2000 LET C = C+1
2010 IF V(R,S) = 0 THEN 2040
2020 LET V(R,S) = 3
2030 GOTO 2050
2040 LET V(R,S) = 2
2050 LET R = R+1
2060 IF C = H*U+1 THEN 2310
2070 LET Q = 0
2080 GOTO 1330
2090 IF Q = 1 THEN 2190
2100 LET W (R,S+1) = C

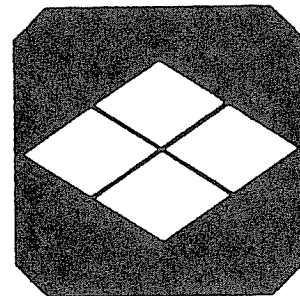
```

```

2110 LET C = C+1
2120 IF U(R,S) = 0 THEN 2150
2130 LET U(R,S) = 3
2140 GOTO 2160
2150 LET U(R,S) = 1
2160 LET S = S+1
2170 IF C = H*U+1 THEN 2310
2180 GOTO 900
2190 LET Z = 1
2200 IF U(R,S) = 0 THEN 2240
2210 LET U(R,S) = 3
2220 LET Q = 0
2230 GOTO 2290
2240 LET U(R,S) = 1
2250 LET Q = 0
2260 LET R = 1
2270 LET S = 1
2280 GOTO 890
2290 GOTO 730
2300 REM
2310 FOR J = 1 TO U
2320   PRINT "I";
2330   FOR I = 1 TO H
2340     IF U(I,J) < 2 THEN 2370
2350     PRINT "  ";
2360     GOTO 2380
2370     PRINT "  I";
2380   NEXT I
2390   PRINT
2400   FOR I = 1 TO H
2410     IF U(I,J) = 0 THEN 2450
2420     IF U(I,J) = 2 THEN 2450
2430     PRINT ": ";
2440     GOTO 2460
2450     PRINT ":-";
2460   NEXT I
2470   PRINT ":"
2480 NEXT J
2490 END

```

PRINTS OUT THE MAZE FROM HERE ON.



POKER:

DESCRIPTION

If you like card games and would like to play for high stakes or low but just haven't got the extra money, this is the way to go! This program simulates a five card draw poker game. Up to five players are allowed and the computer is the dealer and keeps everybody honest.

USERS

Anyone who enjoys a good poker game will enjoy this program. If you've always wanted to try your hand at poker this is one sure way of doing it safely and for the old pro its' going to be a little bit harder to bluff, but go ahead and try.

INSTRUCTIONS

To use this program simply type the word RUN. The game will then commence. The game is self instructing and will ask for all necessary inputs. For additional information list Poker.

LIMITATIONS

With the exception of the ABS () statement in line 2230 the program should immediately execute in any basic speaking computer with at least 8K of available memory. The source code for Poker will require 7K Bytes of memory for storage.

POKER

```

0050 PRINT "THIS IS THE POKER GAME OF FIVE CARD DRAW FOR 1 TO 5 PLAYERS"
0060 PRINT
0070 PRINT "    THE DEALER IS A GI AND GETS A COFFEE BREAK EVERY HALF";
0080 PRINT "HOUR. WHAT TIME IS IT NOW?";
0090 INPUT N
0100 REM
0110 REM THE FOLLOWING LOOPS ARE USED TO GET RANDOM NUMBERS GOING FOR THE DEAL.
0120 REM
0130 FOR I= 1 TO N/2
0140     K=T(10*RND(X))
0150 NEXT I
0160 DIM D(52)
0170 FOR A=1 TO 52
0180     D(A) = 0
0190 NEXT A
0200 DIM Q(52)
0210 FOR A=0 TO 39 STEP 13
0220     FOR C = 1 TO 13
0230         Q(A+C) = C
0240     NEXT C
0250 NEXT A
0260 K = K+1
0270 FOR P=1 TO 5
0280     E(P) = 0
0290     U(P) = 0
0300     T(P) = 0
0310 NEXT P
0320 U(3) = 1
0330 PRINT
0340 REM
0350 REM START OF THE GAME.
0360 REM
0370 PRINT"ENTER THE NUMBER OF HANDS YOU WANT DEALT";
0380 INPUT H
0390 IF H =0 GOTO 3130
0400 G =0
0410 REM
0420 REM THE FOLLOWING IS USED TO GET THE PLAYERS NAMES.
0430 REM
0440 PRINT"NOW ENTER THE FIRST NAME OF ";
0450 IF G=1 GOTO 540
0460 IF G=2 GOTO 590
0470 IF G=3 GOTO 640
0480 IF G=4 GOTO 690
0490 PRINT"PLAYER 1";
0500 INPUT A$

```

```

0510 G = G+1
0520 IF G=H GOTO 750
0530 IF G<H GOTO 440
0540 PRINT"PLAYER 2";
0550 INPUT B$
0560 G = G+1
0570 IF G=H GOTO 750
0580 IF G<H GOTO 440
0590 PRINT"PLAYER 3";
0600 INPUT C$
0610 G = G+1
0620 IF G=H GOTO 750
0630 IF G<H GOTO 440
0640 PRINT"PLAYER 4";
0650 INPUT D$
0660 G = G+1
0670 IF G=H GOTO 750
0680 IF G<H GOTO 440
0690 PRINT"PLAYER 5";
0700 INPUT E$
0710 PRINT
0720 REM
0730 REM USED TO MAKE THE GAME INTERESTING.
0740 REM
0750 PRINT"ANTE UP! PUT SOMETHING IN THE POT. YOU MAY BET ANY AMOUNT";
0760 PRINT "FROM $1.00 TO $100.00 DO NOT USE THE $ SIGN";
0770 P = 1
0780 W = 0
0790 INPUT W
0800 PRINT
0810 W(2) = W
0820 IF W<=0 THEN 3130
0830 REM
0840 REM DEALING EVERYONE THEIR HAND ONE AT A TIME.
0850 REM
0860 IF H=1 GOTO 900
0870 PRINT"NOW IF EVERYONE BUT "A$" WILL STEP BACK I WILL DEAL HIS HAND."
0880 PRINT A$;
0890 GOSUB 2860
0900 PRINT A$
0910 FOR C1=1 TO 5
0920     GOSUB 2160
0930 NEXT C1
0940 GOSUB 2770
0950 IF H=1 GOTO 1370
0960 PRINT"STEP BACK AND HAVE "B$" STEP UP."
0970 PRINT
0980 PRINT B$;
0990 GOSUB 2860
1000 PRINT B$
1010 FOR C2=1 TO 5

```

```

1020     GOSUB 2160
1030 NEXT C2
1040 GOSUB 2770
1050 IF H=2 GOTO 1370
1060 PRINT"STEP BACK AND HAVE "C$" STEP UP."
1070 PRINT
1080 PRINT C$;
1090 GOSUB 2860
1100 PRINT C$
1110 FOR C3=1 TO 5
1120     GOSUB 2160
1130 NEXT C3
1140 GOSUB 2770
1150 IF H=3 GOTO 1370
1160 PRINT"STEP BACK AND HAVE "D$" STEP UP."
1170 PRINT
1180 PRINT D$;
1190 GOSUB 2860
1200 PRINT
1210 PRINT D$
1220 FOR C4=1 TO 5
1230     GOSUB 2160
1240 NEXT C4
1250 GOSUB 2770
1260 IF H=4 GOTO 1370
1270 PRINT"STEP BACK AND HAVE "E$" STEP UP."
1280 PRINT
1290 PRINT E$;
1300 GOSUB 2860
1310 PRINT
1320 PRINT E$
1330 FOR C5=1 TO 5
1340     GOSUB 2160
1350 NEXT C5
1360 GOSUB 2770
1370 PRINT"NOW LOOK AT YOUR CARDS AND BET'EM LIKE YOU SEE EM";
1380 W1 = 0
1390 INPUT W1
1400 PRINT
1410 PRINT"DID ANYONE DROP OUT? ENTER THE NUMBER OF PLAYERS STILL IN";
1420 INPUT H1
1430 PRINT
1440 PRINT"IF YOU DON'T WANT ANY OR HAVE DROPEO OUT ENTER A ZERO."
1450 PRINT
1460 PRINT A$;
1470 INPUT P1
1480 IF P1=0 GOTO 1540
1490 IF P1=>6 GOTO 3060
1500 FOR C6=1 TO P1
1510     GOSUB 2160
1520 NEXT C6

```

```

1530 GOSUB 2770
1540 IF H=1 GOTO 2920
1550 PRINT"STEP BACK AND HAVE "B$" STEP UP."
1560 PRINT
1570 PRINT B$;
1580 GOSUB 2860
1590 PRINT
1600 PRINT"HOW MANY CARDS DO YOU WANT "B$;
1610 P2 = 0
1620 INPUT P2
1630 IF P2=0 GOTO 1690
1640 IF P2=>6 GOTO 3060
1650 FOR C7=1 TO P2
1660     GOSUB 2160
1670 NEXT C7
1680 GOSUB 2770
1690 IF H=2 GOTO 2920
1700 PRINT"STEP BACK AND HAVE "C$" STEP UP."
1710 PRINT
1720 PRINT C$;F1730 GOSUB 2860
1740 PRINT
1750 PRINT" HOW MANY CARDS DO YOU WANT "C$;
1760 P3 = 0
1770 INPUT P3
1780 IF P3=0 GOTO 1840
1790 IF P3=>6 GOTO 3060
1800 FOR C8=1 TO P3
1810     GOSUB 2160
1820 NEXT C8
1830 GOSUB 2770
1840 IF H=3 GOTO 2920
1850 PRINT"STEP BACK AND HAVE "D$" STEP UP."
1860 PRINT
1870 PRINT D$;
1880 GOSUB 2860
1890 PRINT
1900 PRINT "HOW MANY CARDS DO YOU WANT "D$;
1910 P4 = 0
1920 INPUT P4
1930 IF P4=0 GOTO 1990
1940 IF P4=>6 GOTO 3060
1950 FOR C9=1 TO P4
1960     GOSUB 2160
1970 NEXT C9
1980 GOSUB 2770
1990 IF H=4 GOTO 2920
2000 PRINT"STEP BACK AND HAVE "E$" STEP UP."
2010 PRINT
2020 PRINT E$;
2030 GOSUB 2860
2040 PRINT

```

```

2050 PRINT " HOW MANY CARDS DO YOU WANT ";
2060 P5 = 0
2070 INPUT P5
2080 IF P5 = 0 GOTO 2140
2090 IF P5 => 6 GOTO 3060
2100 FOR F1 = 1 TO P5
2110     GOSUB 2160
2120 NEXT F1
2130 GOSUB 2770
2140 IF H=5 GOTO 2920
2150 REM THIS IS USED TO DEAL THE CARDS.
2160 GOSUB 2230
2170 T(P) = T(P)+C
2180 IF U(5) = 0 GOTO 2210
2190 U(5) = 0
2200 RETURN
2210 GOSUB 2470
2220 RETURN
2230 N = 10*(1+ABS(COS(N+W1)))
2240 FOR A=1 TO N
2250     X =INT(52.9999999*RND(Y))
2260     IF X = 0 GOTO 2230
2270 NEXT A
2280 IF D(X)=0 THEN 2350
2290 R = R+1
2300 IF R<50 THEN 2230
2310 FOR A=1 TO 52
2320     IF D(A) =K GOTO 2340
2330     D(A)=0
2340 NEXT A
2350 R = 0
2360 D(X) = K
2370 IF Q(X)<>1 THEN 2410
2380 C = 11
2390 E(P) = E(P)+1
2400 RETURN
2410 IF Q(X)>10 THEN 2440
2420 C = Q(X)
2430 RETURN
2440 C = 10
2450 RETURN
2460 REM THIS PART IS USED TO PRINT THE CARDS.
2470 GOSUB 2500
2480 GOSUB 2650
2490 RETURN
2500 IF Q(X)<>1 GOTO 2530
2510 PRINT " ACE ";
2520 RETURN
2530 IF Q(X)>10 GOTO 2560
2540 PRINT Q(X);
2550 RETURN

```

```

2560 IF Q(X)>11 GOTO 2590
2570 PRINT " JACK ";
2580 RETURN
2590 IF Q(X)>12 GOTO 2620
2600 PRINT " QUEEN ";
2610 RETURN
2620 PRINT " KING ";
2630 RETURN
2640 REM PRINT SUIT
2650 IF X>39 GOTO 2700
2660 IF X>26 GOTO 2720
2670 IF X>13 GOTO 2740
2680 PRINT "OF SPADES"
2690 RETURN
2700 PRINT "OF CLUBS"
2710 RETURN
2720 PRINT "OF HEARTS"
2730 RETURN
2740 PRINT "OF DIAMONDS"
2750 RETURN
2760 REM THIS IS USED TO GIVE THE PLAYERS ENOUGH PAPER TO REMOVE HAND
2770 PRINT""
2780 PRINT"NOW"
2790 PRINT"    IF"
2800 PRINT"        YOU"
2810 PRINT"            WILL"
2820 PRINT"                REMOVE"
2830 PRINT"                    YOUR"
2840 PRINT"                        HAND"
2850 RETURN
2860 PRINT" IS EVERYONE BACK (TYPE YES OR NO)";
2870 INPUT O$
2880 IF O$ = "YES" GOTO 2910
2890 PRINT"COME ON NOW EVERYONE BACK!! NOW IS EVERYONE BACK";
2900 GOTO 2870
2910 RETURN
2920 PRINT"NOW YOU GET ONE MORE CHANCE TO BET YOUR HANDS SO BET THEM HIGH";
2930 INPUT M2
2940 PRINT"DID ANYONE DROP OUT? ENTER THE NUMBER OF PLAYERS STILL IN";
2950 INPUT H2
2960 Z = (M*H)+(M1*H1)+(M2*H2)
2970 PRINT "THE WINNER GETS THE WHOLE THING WHICH IS $";Z
2980 PRINT
2990 PRINT "WANT ANOTHER HAND (TYPE YES OR NO)";
3000 INPUT O$
3010 IF O$ = "NO" GOTO 3130
3020 PRINT"ARE THE SAME PLAYERS IN THE GAME (TYPE YES OR NO)";
3030 INPUT O$
3040 IF O$ = "YES" GOTO 750
3050 IF O$ = "NO" GOTO 370
3060 PRINT "SO SORRY CHARLIE ALL YOU CAN DRAW IS FIVE!!! FIVE THAT'S ALL"
3070 PRINT "WANT TO TRY AGAIN"
3080 IF P1=>6 GOTO 1460
3090 IF P2=>6 GOTO 1600
3100 IF P3=>6 GOTO 1750
3110 IF P4=>6 GOTO 1900
3120 IF P5=>6 GOTO 2050
3130 END

```

POPUL:

DESCRIPTION

This program calculates and prints yearly population projections for an area. The projection may be printed for a number of years, up to 99. The program uses compound interest formulas to generate the projections.

USERS

Individuals interested in the economics, social problems and growth factors of an area will find this program very helpful. This could include civic groups, home owners, engineers, environmentalists, and local government consultants. The area of interest may be a fish tank, back yard, or entire city.

INSTRUCTIONS

This program requires no special initialization prior to execution, just load in the program and type RUN. Popul is self instructional and will request all input necessary for operation. List the program for additional program information.

LIMITATIONS

Popul will require 3K Bytes of memory for storage and 5K for program execution. This program should execute without problems in most 4K Basic compilers.

POPUL

```
100REM POPUL (BASIC PROGRAM BEGINS AT LINE 210)
110REM
130REM
140REM DESCRIPTION--AMONG THE MANY APPLICATIONS OF THE COMPOUND
150REM INTEREST FORMULA IS THAT OF POPULATION
160REM PROJECTION. IT IS NOT ENTIRELY ACCURATE IN
170REM THAT IT ASSUMES A STEADY INCREASE EACH YEAR.
175REM BUT IT IS USEFUL FOR SHOWING ROUGHLY HOW AN
180REM AREA WILL INCREASE... THIS PROGRAM WILL
185REM GENERATE DATA FOR ANY NUMBER OF YEARS AT ANY
190REM REQUESTED INTERVALS>
195REM
200REM INSTRUCTIONS--TYPE 'RUN' AND FOLLOW INSTRUCTIONS.
205REM
210REM*****
215 DIM F(100)
217 DIM Z(100)
220 PRINT "THIS PROGRAM WILL PROJECT POPULATION GROWTH FOR ANY NUMBER OF"
230 PRINT "YEARS USING THE COMPOUND INTEREST FORMULA."
240 PRINT
250 PRINT "WHAT IS THE NAME OF THE AREA WE ARE STUDYING"
260 INPUT A$
270 PRINT
280 PRINT
290 PRINT "PRINT THE ANNUAL PERCENT OF GROWTH FOR YOUR POPULATION";
300 INPUT R
310 PRINT
320 PRINT "FOR HOW MANY YEARS DO YOU WISH TO HAVE DATA COMPUTED";
330 INPUT N
340 LET I=0
350 LET A = N
360 LET A2 = A
370 IF A < 9 THEN 570
380 IF A = 9 THEN 590
390 PRINT "LIST THE FIRST 9 YEARS TO BE COMPUTED; SEPARATE YOUR NUMBERS"
391 PRINT "WITH COMMAS."
400 GO TO 500
410 PRINT
420 PRINT
430 PRINT "WHAT IS THE YEAR FOR YOUR BASIC DATA";
440 INPUT B
450 PRINT
460 PRINT "WHAT IS THE POPULATION FOR THE BASE YEAR(ND COMMAS PLEASE)"
470 INPUT P
480 PRINT
```



```

490 GO TO 630
500 INPUT F(I+1),F(I+2),F(I+3),F(I+4),F(I+5),F(I+6),F(I+7),F(I+8),F(I+9)
510 LET A = A-9
520 LET I = I+9
530 IF A < 1 THEN 410
540 IF A < 9 THEN 605
545 PRINT
550 PRINT "LIST THE NEXT 9 YEARS YOU WISH COMPUTED"
560 GO TO 500
570 PRINT "LIST THE ";N;"YEARS YOU WISH COMPUTED AND";9-N;"ZEROS"
571 PRINT "SEPARATE NUMBERS WITH COMMAS. "
580 GO TO 500
590 PRINT "LIST THE FIRST 9 YEARS FOR WHICH YOU WISH PROJECTIONS"
591 PRINT "SEPARATE THE NUMBERS WITH COMMAS."
600 GO TO 500
605 PRINT
610 PRINT "LIST THE LAST ";A;"YEARS AND" ;9-A;"ZEROS"
620 GO TO 500
630 PRINT
640 PRINT "POPULATION PROJECTION IS AS FOLLOWS"
650 PRINT
660 PRINT   A$
670 PRINT
680 PRINT "      DATE      POPULATION "
690 PRINT
700 FOR M = 1 TO N
710 LET X = 1 + (R/100)
720 LET Y = X*(F(M)-B)
730 LET Z (M) = INT (P*Y)
740 PRINT F(M),Z(M)
750 NEXT M
760 PRINT
770 PRINT
780 END

```



PROFITS:

DESCRIPTION

PROFITS is used to analyze the profit from various departments of a company. With slight modification this program can analyze various household expenses such as: phone, heat, light, groceries, etc. versus income. For business use the amount of inventory and its' rate of change is taken into consideration when determining the profit or profit margin.

USERS

Individuals who are cost conscious put this program to good use. It could be used to analyze house-hold spending versus income. These results could be used to streamline or trim the budget. Businessmen as well as efficiency experts could benefit from using this program by increasing their departments' profit margins.

INSTRUCTIONS

List the program for detailed instructions. Profits is self prompting and will request all needed information while executing.

LIMITATIONS

The TAB () statement is used throughout this program starting in line 320. This program will store and execute in 4K Bytes of memory in most systems, excluding the amount of memory required to store the basic compiler.

PROFITS

```

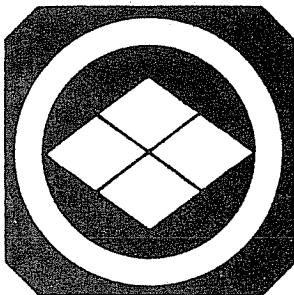
11REM
12REM*****
14REM
15REM DESCRIPTION: THIS BASIC PROGRAM COMPUTES PROFITABILITY OF
16REM                DEPARTMENTS OF A FIRM.
17REM
18REM*****
20REM
22REM INSTRUCTIONS-- TYPE "RUN" TO USE
24REM                PROGRAM WILL ASK FOR NUMBER OF DEPART-
26REM                MENTS, SALES, BEGINNING INVENTORY, ENDING
28REM                INVENTORY, AND PURCHASES
34REM
35REM*****
36REM
100 PRINT "HOW MANY DEPARTMENTS DO YOU HAVE";
110 INPUT D
120 PRINT "PERIOD LENGTH(MOS)";
130 INPUT P;
140 LET D9 = 100 * 30 * P;
150 PRINT "COMPANY NAME";
160 INPUT C;
170 PRINT "
"
180 PRINT "FOR EACH DEPARTMENT ENTER:"
190 PRINT "    SALES, BEGINNING INVENTORY, ENDING INVENTORY, PURCHASES"
200 PRINT
210 FOR N = 1 TO D
220 PRINT N;
230 INPUT S(N), B(N), E(N), P(N)
240 LET C(N) = P(N) + B(N) - E(N)
250 LET G(N) = S(N) - C(N)
260 NEXT N
270 PRINT"
"
280 PRINT "-----"
290 PRINT
300 PRINT C;
310 PRINT
320 PRINT TAB(46); "GROSS"; TAB(58); "% GROSS"
330 PRINT TAB(10); "SALES"; TAB(22); "C/G/S"; TAB(34); "%C/G/S";

```

```

340 PRINT TAB(46); "PROFIT"; TAB(58); "PROFIT"
350 PRINT
360 FOR N = 1 TO D
370 PRINT "DEPT"; N;
380 PRINT TAB(10); S(N); TAB(22); C(N);
390 PRINT TAB(34); INT(10000*C(N)/S(N) + .5)/100;
400 PRINT TAB(46); G(N); TAB(58); INT(10000*G(N)/S(N) + .5)/100
410 LET X = X + S(N)
420 LET Y = Y + C(N)
430 LET Z = Z + G(N)
440 LET W = W + E(N)
450 NEXT N
460 PRINT
470 PRINT "TOTALS"; TAB(10); X; TAB(22); Y; TAB(34); INT(10000*Y/X + .5)/100;
480 PRINT TAB(46); Z; TAB(58); INT(10000*Z/X + .5)/100
490 PRINT
500 PRINT
510 PRINT
520 PRINT TAB(10); "PERCENT"; TAB(22); "PERCENT"; TAB(34); "PERCENT";
530 PRINT TAB(46); "PERCENT"; TAB(58); "DAYS"
540 PRINT TAB(10); "OF SALES"; TAB(22); "OF C/G/S"; TAB(34); "GROSS PFT";
550 PRINT TAB(46); "INVENTORY"; TAB(58); "INVENTORY"
560 PRINT
570 PRINT
580 FOR N = 1 TO D
590 PRINT "DEPT"; N;
600 PRINT TAB(10); INT(10000*S(N)/X + .5)/100;
610 PRINT TAB(22); INT(10000*C(N)/Y + .5)/100;
620 PRINT TAB(34); INT(10000*G(N)/Z + .5)/100;
630 PRINT TAB(46); INT(10000*E(N)/W + .5)/100;
640 PRINT TAB(58); INT(D9*( (B(N)+E(N)) / (2*C(N)) ) + .5)/100;
650 PRINT "DAYS"
660 NEXT N
670 PRINT
680 PRINT "OVERALL INVENTORY TURNOVER";
690 PRINT INT(D9*(W/Y) + .5)/100; "DAYS"
700 PRINT
710 PRINT "-----"
720 END

```



EXAMPLE PROBLEM

RUN

HOW MANY DEPARTMENTS DO YOU HAVE ? 3
 PERIOD LENGTH (MONTHS) ? 12
 COMPANY NAME ? HARRY GREEN CO.

FOR EACH DEPARTMENT ENTER: SALES, BEGINNING INVENTORY, ENDING INVENTORY, PURCHASES

1 7300000, 400000, 330000, 300000
 2 7200000, 300000, 200000, 200000
 3 7500000, 600000, 500000, 500000

HARRY GREEN CO.

	SALES	C/G/S	%C/G/S	GROSS PROFIT	% GROSS PROFIT
DEPT 1	300000	400000	133.33	-100000	-33.33
DEPT 2	200000	300000	150.00	-100000	-50.00
DEPT 3	500000	600000	120.00	-100000	-20.00
TOTALS	1000000	1300000	130.00	-300000	-30.00

	PERCENT OF SALES	PERCENT OF C/G/S	PERCENT GROSS PFT	PERCENT INVENTORY	DAYS INVENTORY
DEPT 1	30	30.77	33.33	30	315 DAYS
DEPT 2	20	23.08	33.33	20	300 DAYS
DEPT 3	50	46.15	33.33	50	330 DAYS

OVERALL INVENTORY TURNOVER 276.92 DAYS

QUBIC:

DESCRIPTION

This is the game of TIC-TAC-TOE in three dimensions. The game is played on a 4x4x4 board. The playing board may be printed before each move, if desired. This is a very complex game and it will be extremely difficult for you to beat the computer, although I have seen it done where the player solves the algorithms with a calculator before each move.

USERS

Game enthusiasts and problem solvers will find Qubic both challenging and interesting.

INSTRUCTIONS

Type RUN and Qubic will ask you if you want instructions. Each move codes a 3 digit number specifying the level and location of the move. All necessary prompts and instructions will be generated by the program during execution.

LIMITATION

This game uses two dimensional arrays; see line 315, throughout. The source code requires 7K Bytes for storage and will execute in 12K.

QUBIC

```

120REM
130REM DESCRIPTION--THE PROGRAM PLAYS 3-DIMENSIONAL TIC-TAC-TOE ON A
140REM 4X4X4 BOARD WITH THE USER. THE USER CAN WIN IF HE PLAYS CORRECTLY.
150REM
160REM INSTRUCTIONS--TYPE "RUN" AND FOLLOW INSTRUCTIONS.
170REM
180REM
190REM * * * * *
200REM
210 PRINT "DO YOU WANT INSTRUCTIONS";
220 INPUT C$
230 IF C$="NO" THEN 320
240 IF C$="YES" THEN 265
250 PRINT "INCORRECT ANSWER. PLEASE TYPE REMYESREM OR REMNOREM";
260 GOTO 220
265 PRINT
270 PRINT "THE GAME IS TIC-TAC-TOE IN A 4 X 4 X 4 CUBE."
280 PRINT "EACH MOVE IS INDICATED BY A 3 DIGIT NUMBER, WITH EACH"
290 PRINT "DIGIT BETWEEN 1 AND 4 INCLUSIVE. THE DIGITS INDICATE THE"
300 PRINT "LEVEL, ROW, AND COLUMN, RESPECTIVELY, OF THE OCCUPIED"
305 PRINT "PLACE. "
306 PRINT
307 PRINT "TO PRINT THE PLAYING BOARD, TYPE 0 (ZERO) AS YOUR MOVE."
308 PRINT "THE PROGRAM WILL PRINT THE BOARD WITH YOUR MOVES INDI-"
309 PRINT "CATED WITH A (Y), THE MACHINE'S MOVES WITH AN (M), AND"
310 PRINT "UNUSED SQUARES WITH A ( )."
311 PRINT
312 PRINT "TO STOP THE PROGRAM RUN, TYPE 1 AS YOUR MOVE."
315 DIM X(64),L(76),M(76,4),Y(16)
320 FOR I = 1 TO 16
330 READ Y(I)
340 NEXT I
350FOR I=1 TO 76
360FOR J = 1 TO 4
370 READM(I,J)
380 NEXT J
390 NEXT I
400 FOR I = 1 TO 64
410 LET X (I) =0
420 NEXT I
430 LET Z=1
440 PRINT "DO YOU WANT TO MOVE FIRST";
450 INPUT S$
460 IF S$="NO" THEN 630

```

```

470 IF S#="YES" THEN 500
480 PRINT "INCORRECT ANSWER. PLEASE TYPE REMYESREM OR REMNOREN. ";
490 GOTO 450
500 PRINT " "
510 PRINT "YOUR MOVE";
520 INPUT J1
521 IF J1=1 THEN 2770
522 IF J1<>0 THEN 525
523 GOSUB 2550
524 GOTO 500
525 IF J1<111 THEN 2750
526 IF J1>444 THEN 2750
530 GOSUB 2500
540 LET K1=INT(J1/100)
550 LET J2=(J1-K1*100)
560 LET K2=INT(J2/10)
570 LET K3= J1 - K1*100 -K2*10
580 LET N=16*K1+4*K2+K3-20
590 IF X(M)=0 THEN 620
600 PRINT "THAT SQUARE IS USED, TRY AGAIN"
610 GOTO 500
620 LET X(M)=1
630 GOSUB 1640
640 FOR J=1 TO 3
650 FOR I=1 TO 76
660 IF J=1 THEN 720
670 IF J=2 THEN 790
680 IF J=3 THEN 930
690 NEXT I
700 NEXT J
710 GOTO 1300
720 IF L(I)<>4 THEN 690
730 PRINT "YOU WIN AS FOLLOWS";
740 FOR J=1 TO 4
750 LET M=M(I, J)
760 GOSUB 1570
770 NEXT J
780 GOTO 1490
790 IF L(I)<>15 THEN 690
800 FOR J=1 TO 4
810 LET M=M(I, J)
820 IF X(M)<>0 THEN 860
830 LET X(M)=5
840 PRINT "MACHINE MOVES TO";
850 GOSUB 1570
860 NEXT J
870 PRINT ", AND WINS AS FOLLOWS"
880 FOR J=1 TO 4
890 LET M=M(I, J)
900 GOSUB 1570
910 NEXT J

```



```

920 GOTO 1490
930 IF L(I)<>3 THEN 690
940 PRINT "NICE TRY MACHINE MOVES TO";
950 FOR J=1 TO 4
960 M=M(I,J)
970 IF X(M)<>0 THEN 1010
980 X(M)=5
990 GOSUB 1570
1000 GOTO 500
1010 NEXT J
1020 GOTO 1300
1030 FOR I = 1 TO 76
1040 LET L(I)=X(M(I,1))+X(M(I,2))+X(M(I,3))+X(M(I,4))
1050 LET L = L(I)
1060 IF L <2 THEN 1130
1070 IF L >=3 THEN 1130
1080 IF L >2 THEN 2230
1090 FOR J = 1 TO 4
1100 IF X(M(I,J))<>0 THEN 1120
1110 LET X(M(I,J))=1/8
1120 NEXT J
1130 NEXT I
1140 GOSUB 1640
1150 FOR I = 1 TO 76
1160 IF L(I)=1/2 THEN 2360
1170 IF L(I)=1+3/8 THEN 2360
1180 NEXT I
1190 GOTO 1830
1200 LET Z = 1
1210 IF X(Y(Z))=0 THEN 1250
1220 LET Z=Z+1
1230 IF Z<>17 THEN 1210
1240 GOTO 1720
1250 LET M=Y(Z)
1260 LET X(M)=5
1270 PRINT "MACHINE MOVES TO";
1280 GOSUB 1570
1290 GOTO 500
1300 LET X=X
1310 FOR I=1 TO 76
1320 LET L(I)=X(M(I,1))+X(M(I,2))+X(M(I,3))+X(M(I,4))
1330 LET L=L(I)
1340 IF L<10 THEN 1410
1350 IF L>=11 THEN 1410
1360 IF L>10 THEN 2230
1370 FOR J=1 TO 4
1380 IF X(M(I,J))<>0 THEN 1400
1390 LET X(M(I,J))=1/8
1400 NEXT J
1410 NEXT I
1420 GOSUB 1640

```

```

1430 FOR I=1 TO 76
1440 IF L(I)=.5 THEN 2360
1450 IF L(I)=5+3/8 THEN 2360
1460 NEXT I
1470 GOSUB 2500
1480 GOTO 1030
1490 PRINT " "
1500 PRINT "DO YOU WANT TO TRY ANOTHER GAME?";
1510 INPUT X$
1520 IF X$="YES" THEN 400
1530 IF X$="NO" THEN 1560
1540 PRINT "INCORRECT ANSWER. PLEASE TYPE REMYESREM OR REMNOREM";
1550 GOTO 1510
1560 STOP
1570 LET K1=INT((M-1)/16)+1
1580 LET J2=M-16*(K1-1)
1590 LET K2=INT((J2-1)/4)+1
1600 LET K3=M-(K1-1)*16-(K2-1)*4
1610 LET M=K1*100+K2*10+K3
1620 PRINT M;
1630 RETURN
1640 FOR S=1 TO 76
1650 LET J1 = M(S,1)
1660 LET J2=M(S,2)
1670 LET J3=M(S,3)
1680 LET J4=M(S,4)
1690 LET L(S)=X(J1)+X(J2)+X(J3)+X(J4)
1700 NEXT S
1710 RETURN
1720 FOR I=1 TO 64
1730 IF X(I)<>0 THEN 1800
1740 LET X(I)=5
1750 LET M=I
1760 PRINT "MACHINE LIKES";
1770 GOSUB 1570
1780 PRINT " "
1790 GOTO 500
1800 NEXT I
1810 PRINT "THE GAME IS A DRAW"
1820 GOTO 1490
1830 FOR K=1 TO 18
1840 LET P=0
1850 FOR I=4*K-3 TO 4*K
1860 FOR J=1 TO 4
1870 LET P=P+X(M(I,J))
1880 NEXT J
1890 NEXT I
1900 IF P<4 THEN 1940
1910 IF P<5 THEN 1970
1920 IF P<9 THEN 1940
1930 IF P<10 THEN 1970

```

```

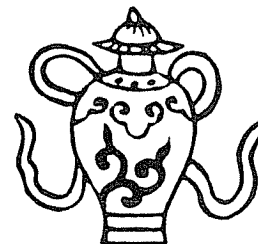
1940 NEXT K
1950 GOSUB 2500
1960 GOTO 1200
1970 LET S=1/8
1980 FOR I=4*K-3 TO 4*K
1990 GOTO 2370
2000 NEXT I
2010 LET S=0
2020 GOTO 1980
2030 DATA 1,49,52,4,13,61,64,16,22,39,23,38,26,42,27,43
2040 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
2050 DATA 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38
2060 DATA 39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56
2070 DATA 57,58,59,60,61,62,63,64
2080 DATA 1,17,33,49,5,21,37,53,9,25,41,57,13,29,45,61
2090 DATA 2,18,34,50,6,22,38,54,10,26,42,58,14,30,46,62
2100 DATA 3,19,35,51,7,23,39,55,11,27,43,59,15,31,47,63
2110 DATA 4,20,36,52,8,24,40,56,12,28,44,60,16,32,48,64
2120 DATA 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61
2130 DATA 2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62
2140 DATA 3,7,11,15,19,23,27,31,35,39,43,47,51,55,59,63
2150 DATA 4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64
2160 DATA 1,6,11,16,17,22,27,32,33,38,43,48,49,54,59,64
2170 DATA 13,10,7,4,29,26,23,20,45,42,39,36,61,58,55,52
2180 DATA 1,21,41,61,2,22,42,62,3,23,43,63,4,24,44,64
2190 DATA 49,37,25,13,50,38,26,14,51,39,27,15,52,40,28,16
2200 DATA 1,18,35,52,5,22,39,56,9,26,43,60,13,30,47,64
2210 DATA 49,34,19,4,53,38,23,8,57,42,27,12,61,46,31,16
2220 DATA 1,22,43,64,16,27,38,49,4,23,42,61,13,26,39,52
2230 FOR J=1 TO 4
2240 IF X(M(I,J))<>1/8 THEN 2330
2250 LET X(M(I,J))=5
2260 IF L(I)<5 THEN 2290
2270 PRINT "LETREMS SEE YOU GET OUT OF THIS: MACHINE MOVES TO";
2280 GOTO 2300
2290 PRINT "YOU FOX. JUST IN THE NICK OF TIME, MACHINE MOVES TO";
2300 LET M=M(I,J)
2310 GOSUB 1570
2320 GOTO 500
2330 NEXT J
2340 PRINT "MACHINE CONCEDES THIS GAME."
2350 GOTO 1490
2360 LET S=1/8
2370 IF I-INT(I/4)*4>1 THEN 2400
2380 LET A=1
2390 GOTO 2410
2400 LET A=2
2410 FOR J=A TO 5-A STEP 5-2*A
2420 IF X(M(I,J))=S THEN 2450
2430 NEXT J
2440 GOTO 2000

```

```

2450 LET X(M(I,J))=5
2460 LET M=M(I,J)
2470 PRINT "MACHINE TAKES";
2480 GOSUB 1570
2490 GOTO 500
2500 FOR I=1 TO 64
2510 IF X(I)<>1/8 THEN 2530
2520 LET X(I)=0
2530 NEXT I
2540 RETURN
2550FOR I=1TO4
2560FOR J=1TO4
2562FOR I1=1TOJ
2564PRINT"  ";
2566NEXT I1
2570FOR K=1TO4
2600LET Q=16*I+4*J+K-20
2610IF X(Q)<>0THEN2630
2620PRINT"( ) ";
2630IF X(Q)<>5THEN2650
2640PRINT"(M) ";
2650IF X(Q)<>1THEN2660
2655PRINT"(Y) ";
2660IF X(Q)<>1/8THEN2670
2665PRINT"( ) ";
2670NEXT K
2680PRINT
2690PRINT
2700NEXT J
2710PRINT
2720PRINT
2730NEXT I
2740RETURN
2750PRINT"INCORRECT MOVE, RETYPE IT--";
2760GOTO520
2770END

```



DO YOU WANT INSTRUCTIONS ?YES

THE GAME IS TIC-TAC-TOE IN A 4 X 4 X 4 CUBE.
EACH MOVE IS INDICATED BY A 3 DIGIT NUMBER, WITH EACH
DIGIT BETWEEN 1 AND 4 INCLUSIVE. THE DIGITS INDICATE THE
LEVEL, ROW, AND COLUMN, RESPECTIVELY, OF THE OCCUPIED
PLACE.

TO PRINT THE PLAYING BOARD, TYPE 2 (ZERO) AS YOUR MOVE.
THE PROGRAM WILL PRINT THE BOARD WITH YOUR MOVES INDI-
CATED WITH A (Y), THE MACHINE'S MOVES WITH AN (M), AND
UNUSED SQUARES WITH A ().

TO STOP THE PROGRAM RUN, TYPE 1 AS YOUR MOVE.

DO YOU WANT TO MOVE FIRST ?0

INCORRECT ANSWER. PLEASE TYPE REMYESREM OR REMNOREM. ?NO

MACHINE MOVES TO 111

YOUR MOVE ?3

```

(M)      ( )      ( )      ( )
      ( )      ( )      ( )      ( )
          ( )      ( )      ( )      ( )
              ( )      ( )      ( )      ( )
                  ( )      ( )      ( )      ( )
                      ( )      ( )      ( )      ( )
                          ( )      ( )      ( )      ( )
                              ( )      ( )      ( )      ( )
                                  ( )      ( )      ( )      ( )
                                      ( )      ( )      ( )      ( )
                                          ( )      ( )      ( )      ( )
                                              ( )      ( )      ( )      ( )
                                                  ( )      ( )      ( )      ( )
                                                      ( )      ( )      ( )      ( )

```

RATES:

DESCRIPTION

This program calculates effective annual interest rates for various interests with different compounding times.

USERS

Persons planning on spending money who may have a choice of financing plans can use this program to determine the true or effective interest they will be repaying, i.e. Businessmen, Housewives, Homeowners, etc.

INSTRUCTIONS

After the program has been loaded into memory type RUN and the program will prompt for all necessary information.

LIMITATIONS

Rates uses two dimensional arrays; see line 530. Line 550 has a MAT INPUT statement as does line 670. Line 770 contains a TAB () statement and lines 490 and 1020 contain print using statements. The program will require 2K Bytes to store the source code and 7K Bytes for execution.

RATES

```

110REM
120REM
140REM  DESCRIPTION--THIS BASIC PROGRAM COMPUTES EFFECTIVE ANNUAL
150REM          RATES OF INTEREST.....
160REM
170REM  INSTRUCTIONS--TYPE 'RUN' TO USE
180REM          INPUT IS REQUESTED
190REM
200REM
210REM*****
220REM
530 DIM R(20),N(20,20),Q(20),Z(20)
540 PRINT "NOMINAL ANNUAL INTEREST RATES--UP TO 20 DIFFERENT RATES"
550 MAT INPUT R
560 LET Q1=NUM(X)
570 FOR J=1 TO Q1
580 IF R(J)<.5 THEN 600
590 LET R(J)=R(J)/100
600 NEXT J
610 PRINT
620 PRINT "EACH RATE COMPOUNDED HOW MANY TIMES PER YEAR?"
630 PRINT "FOR EACH RATE YOU CAN TYPE IN UP TO 20 DIFFERENT COMPOUNDING"
640 PRINT "TIMES (TYPE IN A ZERO FOR CONTINUOUS COMPOUNDING)"
650 FOR J=1 TO Q1
660 PRINT "FOR ";100*R(J);
670 MAT INPUT Z
680 LET Q(J)=NUM(X)
690 FOR K=1 TO Q(J)
700 LET N(J,K)=Z(K)
710 NEXT K
720 NEXT J
730 FOR J=1 TO 5
740 PRINT
750 NEXT J
760 PRINT "NOMINAL ANNUAL  %";TAB(26);"TIMES COMP./ANNUM";
770 PRINT TAB(51);"EFFECTIVE ANNUAL  %"
780 PRINT "-----";TAB(26);"-----";
790 PRINT TAB(51);"-----"
800 PRINT
810 PRINT
820 FOR J=1 TO Q1
830 FOR K=1 TO Q(J)
840 IF N(J,K)=0 THEN 900
850 LET N=N(J,K)
860 LET R=R(J)
870 LET E=100*((1+R/N)^N-1)

```

```

880 LET A=N(J,K)
890 GO TO 1010
900 LET E=(EXP(R(J))-1)*100
930 LET R(J)=R(J)*100
940 PRINT USING 1100, R(J);E
950 GO TO 1030
1010 LET R(J)=R(J)*100
1020 PRINT USING 1110, R(J);A;E
1030 LET R(J)=R(J)/100
1040 NEXT K
1050 PRINT
1060 NEXT J
1100: ###.###
1110: ###.###
1200 END

```

```

CONT. COMP.
###

```

```

###.###
###.###

```



SAMPLE PROBLEM

RUN

NOMINAL ANNUAL INTEREST RATES - UP TO 20 DIFFERENT RATES.
?12.61;12.98;6;6.5;12;12.5

EACH RATE COMPOUNDED HOW MANY TIMES PER YEAR?
FOR EACH RATE YOU CAN TYPE IN UP TO 20 DIFFERENT COMPOUNDING TIMES
(TYPE IN A ZERO FOR CONTINUOUS COMPOUNDING).

FOR 12.61 ?21
FOR 12.98 ?30
FOR 6.00 ?0
FOR 6.50 ?2;4;6
FOR 12.00 ?12
FOR 12.50 ?0

NOMINAL ANNUAL %	TIMES COMP/ANNUM	EFFECTIVE ANNUAL %
12.610	21	13.3968
12.980	30	13.8282
6.000	CONT. COMP.	6.1837
6.500	2	6.6056
6.500	4	6.6602
6.500	6	6.6786
12.000	12	12.6825
12.500	CONT. COMP.	13.3148

RETIRE:

DESCRIPTION

This program computes the rate of retirement pay for a civil service worker. It can be modified to generate retirement schedules for other plans as desired.

USERS

Persons desiring to plan for their retirement will find this program very beneficial. In addition to providing you with a retirement pay schedule it can be very useful Now in planning or deciding which job to take or what position or level of current income to work towards. While this program has been set up for government workers it can easily be modified to encompass any retirement plan beigh offered by an employer.

INSTRUCTIONS

Upon execution the program will ask if you want instructions. Type 1 for Yes or 0 for No. The program will request all necessary data as needed to determine your retirement benefits.

LIMITATIONS

Starting in line 1020 the DEF FNX () statement is used. In addition the PRINT USING statement is used extensively through out this program. RETIRE will require 6K Bytes of memory for program storage and will execute in most 8K Basic compilers with 7K Bytes of available memory.

RETIRE

```

1000 S1 = N1 = H1 = T9 = 0
1010 DIM N(12)
1020 DEF FND (M1,D1,Y1) = M(M1) + D1 + 365*Y1
1030 FOR I = 1 TO 12
1040 READ N(I)
1050 NEXT I
1060 PRINT "DO YOU WISH INSTRUCTIONS ? YES TYPE 1  NO TYPE 0"
1070 INPUT T1
1080 PRINT
1090 IF T1 = 0 GOTO 1260
1100 PRINT "----ALL DATA ENTRIES ARE NUMERICAL AND SEPARATED"
1110 PRINT "      BY COMMAS WHEN THERE IS MORE THAN ONE ENTRY"
1120 PRINT "----ALL DATES ARE      MONTH, DAY, YEAR"
1130 PRINT
1140 PRINT "----THE ENTRY FOR  MONTH, DAY, YEAR, SALARY: "
1150 PRINT "  -ARE PAY CHANGE DATES WITH APPROPRIATE SALARY;"
1160 PRINT "  -ENTRIES ARE TO BE IN REVERSE CHRONOLOGICAL ORDER;"
1170 PRINT "  -WHEN THREE YEAR PERIOD IS OBTAINED, ENTRY WILL"
1180 PRINT "    AUTOMATICALLY' OUT OUT"
1190 PRINT
1200 PRINT "----SURVIUOR ANNUITY TO CHILDREN:"
1210 PRINT "  -TO QUALIFY YOUR CHILD MUST:"
1220 PRINT "  -A. BE UNDER 18 AND BE UNMARRIED; OR"
1230 PRINT "  -B. BE UNDER 22, UNMARRIED AND A FULL TIME STUDENT; OR"
1240 PRINT "  -C. BE DISABLED BEFORE 18, INCAPABLE OF SELF-SUPPORT."
1250 PRINT
1260 PRINT " ENTER---- BIRTHDAY"
1270 INPUT M1,D1,Y1
1280 PRINT
1290 B1 = FND (M1,D1,Y1)
1300 PRINT " ENTER---- SERVICE COMPUTATION DATE"
1310 INPUT M1,D1,Y1
1320 PRINT
1330 B2 = FND (M1,D1,Y1)
1340 PRINT " ENTER---- RETIREMENT DATE"
1350 INPUT M1,D1,Y1
1360 PRINT
1370 B3 = FND (M1,D1,Y1)
1430 PRINT " ENTER----TODAYS DATE"
1440 INPUT M1,D1,Y1
1450 PRINT
1460 B5 = B4 = FND (M1,D1,Y1)
1461 PRINT "HOW MANY CHILDREN ELIGIBLE FOR SURVIUOR ANNUITY"
1462 INPUT E1
1463 PRINT

```

```

1470 PRINT
1480 PRINT "ENTER---MONTH, DAY, YEAR, SALARY"
1490 INPUT M1, D1, Y1, S
1500 IF S < H1 GOTO 1520
1510 H1 = S
1520 B6 = FMD (M1, D1, Y1)
1530 W = INT((B5 - B6)/7)
1540 W1 = W1 + W
1550 IF W1 >= 156 GOTO 1590
1560 S1 = S1 + (W/52) * S
1570 B5 = B6
1580 GOTO 1490
1590 W = W - W1 + 156
1600 S1 = S1 + (W/52) * S
1610 H = H2 = S1/3
1620 A = A2 = (B4 - B2)/365
1630 A3 = (B3 - B2)/365
1640 G = (B4 - B1)/365
1650 W1 = 20075 + B1 - B3
1660 W2 = INT(W1/365)
1670 W3 = W1 - W2*365
1680 F = W2*12 + INT(W3/30)
1690 S4 = INT((B3 - B4)/7)
1700 S5 = (H1 - H) * ((S4 + 78)/78) + 2*H - H1
1710 REM--- H1 = HIGH SALARY
1720 REM--- H = H2 = HIGH THREE YEAR AVERAGE SALARY
1730 REM--- A = A2 = YEARS OF SERVICE AS OF TODAY'S DATE
1740 REM--- A3 = YEARS OF SERVICE TO RETIREMENT DATE
1750 REM--- G = YEARS OF AGE
1760 REM--- F = NUMBER OF MONTHS UNDER 55 AT RETIREMENT
1770 REM--- S5 = ESTIMATED HIGH THREE AVERAGE SALARY
1780 REM          TO RETIREMENT DATE
1790 REM--- S4 = WEEKS DIFFERENCE BETWEEN TODAY AND RETIREMENT
1800 PRINT
1801 IF T9 > 0 GOTO 1870
1840 PRINT USING 1850; H
1850: HIGH THREE AVERAGE SALARY IS #####.##
1860 PRINT
1870 PRINT USING 1880; S5
1880: ESTIMATED HIGH THREE AVERAGE SALARY AT RETIREMENT #####.##
1890 PRINT
1900 GOTO 2060
1910 REM BEGIN - SUBROUTINE FOR BASIC ANNUITY FORMULA
1920 A1 = A - 5
1930 B = .015 * H * 5
1940 IF A1 > 5 GOTO 1980
1950 C = .0175 * H * A1
1960 D = 0
1970 GOTO 2000
1980 C = .0175 * H * 5
1990 D = .02 * H * (A-10)

```

```

2000 E = B + C + D
2010 RETURN
2020 REM--- INPUT IS A = YEARS OF SERVICE
2030 REM--- INPUT IS H = HIGH THREE SALARY
2040 REM--- OUTPUT IS E = ANNUITY
2050 REM END --- SUBROUTINE FOR BASIC ANNUITY FORMULA
2060 IF F > 0 GOTO 2063
2061 F = 0
2063 M2 = F/6
2070 PRINT USING 2080,M2
2080: RETIREMENT ANNUITIES REDUCED ###.## PERCENT
2090 PRINT USING 2100,F
2100: FOR ### MONTHS UNDER 55 YEARS OLD
2110 PRINT
2120 IF S4 = 0 GOTO 2170
2130 A = A3
2140 H = S5
2150 PRINT " RETIREMENT INFORMATION IS ESTIMATED"
2160 PRINT
2170 GOSUB 1920
2200 T = E - E * (F/6) * .01
2210 T1 = T/12
2220 T2 = T/52
2230 PRINT USING 2240,T
2240: #####.## ANNUITY TO RETIRED EMPLOYEE (NO SURVIUOR BENEFIT)
2250 PRINT USING 2260,T1
2260: #####.## PER MONTH
2270 PRINT USING 2280,T2
2280: #####.## PER WEEK
2290 PRINT
2300 IF T > 3600 GOTO 2340
2310 K = .975 * T
2320 L = 0
2330 GOTO 2360
2340 K = .975 * 3600
2350 L = .9 * (T - 3600)
2360 T3 = K + L
2370 T1 = T3/12
2380 T2 = T3/52
2390 PRINT USING 2400,T3
2400: #####.## ANNUITY TO RETIRED EMPLOYEE (SURVIUOR BENEFIT)
2410 PRINT USING 2260,T1
2420 PRINT USING 2280,T2
2430 PRINT
2440 T3 = T * .55
2450 T1 = T3/12
2460 T2 = T3/52
2470 PRINT USING 2480,T3
2480: #####.## ANNUITY TO SURVIVING SPOUSE OF RETIRED EMPLOYEE
2490 PRINT USING 2260,T1
2500 PRINT USING 2280,T2

```

```

2510 PRINT
2520 PRINT
2521 IF T9 > 0 GOTO 2900
2530 H = H2
2531 A = A2
2532 GOSUB 1920
2540 A5 = .55 * E
2550 A6 = .222 * H
2560 A = A + 60 - G
2570 GOSUB 1920
2590 A7 = .55 * E
2600 T3 = A5
2610 IF A6 < A7 GOTO 2650
2620 IF A5 > A7 GOTO 2670
2630 T3 = A7
2640 GOTO 2670
2650 IF A5 > A6 GOTO 2670
2660 T3 = A6
2670 W1 = H * .6
2680 PRINT USING 2690,T3
2690: #####.## ANNUITY TO SURVIVING SPOUSE OF EMPLOYEE
2700 T1 = T3/12
2710 T2 = T3/52
2720 PRINT USING 2260,T1
2730 PRINT USING 2280,T2
2740 PRINT
2750 IF E1 = 0 GOTO 2900
2751 A5 = W1/E1
2760 A6 = 2700/E1
2770 IF A5 > A6 GOTO 2830
2780 IF A5 > 900 GOTO 2810
2790 T1 = A5
2800 GOTO 2850
2810 T1 = 900
2820 GOTO 2850
2830 IF A6 > 900 GOTO 2810
2840 T1 = A6
2850 T2 = T1 * E1
2860 PRINT "SURVIOR ANNUITY TO CHILDREN OF EMPLOYEE"
2870 PRINT USING 2880,T1,T2
2880 : EACH CHILD #####.##, TOTAL CHILDREN ANNUITY #####.##
2900 PRINT
2910 PRINT "DO YOU WISH TO PROJECT ANOTHER RETIREMENT DATE?"
2911 PRINT "YES TYPE 1 NO TYPE 0"
2912 INPUT T9
2913 PRINT
2914 IF T9 = 0 GOTO 2960
2915 PRINT "ENTER--- RETIREMENT DATE"
2920 INPUT M1,D1,Y1
2930 PRINT
2940 B3 = FND (M1,D1,Y1)
2950 GOTO 1610
2960 STOP
2970 DATA 0,31,59,90,120,151
2980 DATA 181,212,243,273,304,334
2990 END

```

SAVINGS:

DESCRIPTION

This program computes the amount of money an investment will accumulate over a period of years, given the interest rate, compounding period and the type of investment.

USERS

Anyone who invests or saves money will find this program can help in determining the most advantageous place to invest your money. This would include homeowners, businessmen and companies, as well as anyone who has a savings account.

INSTRUCTIONS

After the program has been loaded into memory type RUN. The program will then print out a full set of user instructions.

LIMITATIONS

This program should execute in any Basic speaking computer with 3K available memory. Two kilo Bytes are required for program storage and Savings should execute without incident in most 4K Basic compilers.

SAVINGS

```
100 PRINT
110 PRINT "    THIS PROGRAM CALCULATES THE AMOUNT OF MONEY THAT"
120 PRINT "WOULD ACCUMULATE AFTER N YEARS AT AN ANNUAL INTEREST"
130 PRINT "RATE R COMPOUNDED T TIMES PER YEAR, WHEN THE INITIAL"
140 PRINT "AMOUNT IS P AND AN AMOUNT D IS ADDED AT THE BEGINNING"
150 PRINT "OF EACH SUBSEQUENT YEAR.  NOTE THAT P AND D ARE GIVEN"
160 PRINT "IN DOLLARS, N AND T MUST BE INTEGERS, AND R IS GIVEN"
170 PRINT "AS A PERCENTAGE."
180 PRINT
190 PRINT "WHAT ARE P,D,N,T,R";
200 INPUT L,N,Y,X,A
210 IF X<>0 THEN 250
220 PRINT
230 PRINT "% MUST NOT EQUAL ZERO..."
240 GOTO 180
250 LET D=0
260 LET B=.01*A
270 LET S=L
280 LET D=0
290 LET S=L
300 FOR Z=1 TO Y
310 LET S=S+D
320 FOR W=1 TO X
330 LET S=S*(1+B/X)
340 NEXT W
350 LET D=N
360 NEXT Z
370 PRINT
380 PRINT "    AFTER ";Y;"YEARS, ";L;"DOLLARS INVESTED AT";A
390 PRINT "    PERCENT COMPOUNDED ";X;"TIMES PER YEAR, WITH THE"
400 PRINT "    ADDITION OF ";N;"DOLLARS PER YEAR, YIELDS A TOTAL"
410 PRINT "    OF ";S;"DOLLARS."
420 PRINT
430 PRINT "MORE DATA (1=YES,0=NO)";
440 INPUT I
450 IF I=1 THEN 190
9999 END
```


SAMPLE PROBLEM

TO CALCULATE THE TOTAL AMOUNT OF MONEY THAT WILL HAVE ACCUMULATED AFTER A FOUR YEAR PERIOD WITH AN INITIAL INVESTMENT OF \$6000. THE INTEREST RATE IS 6%, COMPOUNDED FOUR TIMES PER YEAR. AN ADDITIONAL \$200 HAS BEEN DEPOSITED AT THE BEGINNING OF THE SECOND, THIRD AND FOURTH YEARS.

SOLUTION

RUN

THIS PROGRAM CALCULATES THE AMOUNT OF MONEY THAT WOULD ACCUMULATE AFTER N YEARS AT AN ANNUAL INTEREST RATE R COMPOUNDED T TIMES PER YEAR, WHEN THE INITIAL AMOUNT IS P AND AN AMOUNT D IS ADDED AT THE BEGINNING OF EACH SUBSEQUENT YEAR. NOTE THAT P AND D ARE GIVEN IN DOLLARS, N AND T MUST BE INTEGERS, AND R IS GIVEN AS A PERCENTAGE.

WHAT ARE P,D,N,T,R ?6000,200,4,4,6

AFTER 4 YEARS, 6000 DOLLARS INVESTED AT 6 PERCENT COMPOUNDED 4 TIMES PER YEAR, WITH THE ADDITION OF 200 DOLLARS PER YEAR, YIELDS A TOTAL OF 8290.608 DOLLARS

MORE DATA (1=YES, 0=NO) ?1

WHAT ARE P,D,N,T,R ?6000,0,4,4,6

AFTER 4 YEARS, 6000 DOLLARS INVESTED AT 6 PERCENT COMPOUNDED 4 TIMES PER YEAR, WITH THE ADDITION OF 0 DOLLARS PER YEAR, YIELDS A TOTAL OF 7613.913 DOLLARS.

MORE DATA (1=YES, 0=NO) ?1

WHAT ARE P,D,N,T,R ?0,200,4,4,6

AFTER 4 YEARS, 0 DOLLARS INVESTED AT 6 PERCENT COMPOUNDED 4 TIMES PER YEAR, WITH THE ADDITION OF 200 DOLLARS PER YEAR, YIELDS A TOTAL OF 676.6948 DOLLARS.

MORE DATA (1=YES, 0=NO) ?1

WHAT ARE P,D,N,T,R ?6000,200,4,1,5.25

AFTER 4 YEARS, 6000 DOLLARS INVESTED AT 5.25 PERCENT COMPOUNDED 4 TIMES PER YEAR, WITH THE ADDITION OF 200 DOLLARS PER YEAR, YIELDS A TOTAL OF 8027.977 DOLLARS

MORE DATA (1=YES, 0=NO) ?0

SBA:

DESCRIPTION

SBA calculates a loan repayment schedule for "Small Business Administration" loans. The printout covers the entire period of the loan utilizing a tabular format for the various categories and provides a comprehensive yearly summary statement.

USERS

Businessmen who are interested in obtaining SBA funds for use in their business will find this program very informative and far less costly to use than an accountant. As small businesses are measured in terms of dollar income per operating unit, this means that even large companies may be able to obtain SBA loans for their various offices around the country.

INSTRUCTIONS

The program is self prompting but should be listed for detailed input instructions. After it is loaded into memory and listed, if desired, type RUN.

LIMITATIONS

Line 340 contains a DEF FNX () statement, line 380 an ABS () statement, line 630 a Change statement, line 650 a FNEND statement and line 1180 a TAB () statement. Some of these statements may not be in your Basic compiler but if you have an 8K version of Basic these statements may easily be converted to your compiler's statements. Statement definitions are shown in appendix A at the end of Volume II. The source code for SBA requires 5K Bytes for storage and will execute in 6K Bytes of memory.

SBA

```
113REM
120REM DESCRIPTION-- COMPUTES AND PRINTS A MONTHLY PAYMENTS
130REM SCHEDULE FOR A SMALL BUSINESS ADMINISTRATION LOAN.
140REM
150REM AT THE END OF EACH CALENDAR YEAR AND AT THE FINAL RETIREMENT
160REM OF THE LOAN IT PRINTS A COMPREHENSIVE SUMMARY STATEMENT
170REM GIVING THE BEGINNING PRINCIPAL, ENDING PRINCIPAL, PRINCIPAL
180REM REPAYMENTS MADE, AND TOTAL INTEREST PAID DURING THE YEAR.
190REM
200REM THE PROGRAM ASSUMES THAT THE BANK SERVICE FEE WILL BE
210REM REIMBURSED BY THE SBA.
230REM
240REM INSTRUCTIONS-- TO USE THE PROGRAM TYPE 'RUN'.
250REM INPUT DATA WILL BE REQUESTED
260REM
270REM*****
280REM
290 PRINT "PLEASE LIST THIS PROGRAM FOR INSTRUCTIONS."
300 PRINT
310 DIM M$(12)
320 DIM P(12)
330REM FUNCTION FOR PRINTING IN A DOLLARS AND CENTS FORMAT
340 DEF FNP(P1,P5)
350 LET P0=0
360 LET FNP=P1
370 IF P1>=0 THEN 400
380 LET P1 =ABS(P1)
390 LET P0=1
395REM NUMBER OF DIGITS LEFT OF DECIMAL POINT
400 LET P9=7
405REM NUMBER OF DECIMALS
410 LET P8=2
420 LET P(0) = P9+P8+1
430 LET P2 = INT(P1*(10↑P8)+ .5)
440 FOR P4 = 1 TO P8
450 LET P3=P2-INT (P2/10)*10
460 LET P(P9+P8+2-P4) =P3+48
470 LET P2=INT(P2/10)
480 NEXT P4
490 LET P(P9+1) = 46
500 FOR P4=1 TO P9
510 LET P3=P2-INT (P2/10)*10
520 LET P(P9+1-P4)=P3+48
530 IF P2 = 0 THEN 560
540 LET P2 =INT (P2/10)
550 GO TO 610
```

```

560 IF P0=0 THEN 600
570 LET P(P9+1-P4) =45
580 LET P0=0
590 GO TO 610
600 LET P(P9+1-P4) = 32
610 NEXT P4
620 PRINT TAB(P5-2);
630 CHANGE P TOP$
640 PRINT P$;
650 FNEND
660 DATA JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
670 FOR Z9= 1TO12
680 READ M$(Z9)
690 NEXT Z9
700 DEF FNI(A,R)=INT(100*(R/12)*A+.5)/100
710 PRINT "BORROWER'S NAME";
720 INPUT C$
730 PRINT "ORIGINAL PRINCIPAL";
740 INPUT A1
750 LET A9=A1
760 PRINT "SBA SHARE";
770 INPUT S1
780 IF S1<1 THEN 800
790 LET S1 = S1/100
800 PRINT "BANK RATE";
810 INPUT R3
820 IF R3<1 THEN 840
830 LET R3=R3/100
840 PRINT "SBA RATE";
850 INPUT R2
860 IF R2<1 THEN 880
870 LET R2=R2/100
880 PRINT "LOAN DATED (MO,YR)";
890 INPUT M,Y
900 LET M=M+1
910 IF M <= 12 THEN 940
920 LET M=1
930 LET Y=Y+1
940 PRINT "MONTHLY PAYMENT";
950 INPUT E0
960 LET A2 = INT(100*A1*S1+.5)/100
970 PRINT "SERVICE FEE(%)";
980 INPUT S9
990 LET S9=S9/1200
1000 PRINT "FOR HOW MANY CALENDAR YEARS DO YOU WISH THE SCHEDULE";
1010 INPUT Y9
1020 LET Y8 = 1
1030 PRINT""
1040 PRINT "          ****SBA LOAN SCHEDULE****"
1050 PRINT
1060 PRINT "BORROWER:  " ;C$

```

```

1070 PRINT "AMOUNT OF LOAN";
1080 LET A1 = FNP(A1,17)
1090 PRINT
1100 PRINT "SBA PARTICIPATION:";100*S1;"%"
1110 PRINT "SBA INTEREST RATE:";100*R2;"%"
1120 PRINT "BANK INTEREST RATE:";100*R3;"%"
1130 PRINT "MONTHLY PAYMENT:";
1140 LET E0=FNP(E0,17)
1150 PRINT
1160 PRINT
1170 PRINT
1180 PRINT TAB(15);"BEGINNING";TAB(27);"INTEREST";TAB(39);"PRINCIPAL";
1190 PRINT TAB(51);"SERVICE";TAB(63);"TOTAL"
1200 PRINT TAB(15);"PRINCIPAL";TAB(27);"PAYMENT";TAB(39);"REPAYMENT";
1210 PRINT TAB(51);" FEE";TAB(63);"PAYMENT"
1220 PRINT
1230 LET N=1
1240 LET A3=A1-A2
1250 LET I2 = FNI(A2,R2)
1260 LET I3=FNI(A3,R3)
1270 LET I1=I2+I3
1280 LET E1=E0-I1
1290 IF E1<=1.1*A1 THEN 1340
1300 LET E3=A3
1310 LET E2=A2
1320 LET E1=E2+E3
1330 GO TO1360
1340 LET E2=A2-INT(100*S1*(A1-E1)+.5)/100
1350 LET E3=E1-E2
1360 PRINT"=====
1370 PRINT
1380 PRINT "PMT #";N;
1390 PRINT N$(N);";";Y
1400 PRINT
1410 LET N=N+1
1420 LET M=M+1
1430 LET C1=C1+I1
1440 LET C2=C2+P1
1450 PRINT "TOTAL LOAN";
1460 LET A1=FNP(A1,15)
1470 LET I1=FNP(I1,27)
1480 LET E1=FNP(E1,39)
1490 IF ABS(E0-E1-I1)<.0001 THEN 1520
1500 LET E9=FNP(E1+I1,63)
1510 GO TO 1530
1520 LET E0=FNP(E0,63)
1530 LET C2=C2+E1
1540 PRINT "SBA SHARE";
1550 LET A2=FNP(A2,15)
1560 LET I2=FNP(I2,27)
1570 LET E2=FNP(E2,39)

```

```

1580 LET S2=INT(S9*A2*100+.5)/100
1590 LET S2=FNP(-S2,51)
1600 LET E8=FNP(E2+I2+S2,63)
1610 PRINT
1620 PRINT"BANK SHARE";
1630 LET A3=FNP(A3,15)
1640 LET I3=FNP(I3,27)
1650 LET E3=FNP(E3,39)
1660 LET S2=FNP(-S2,51)
1670 LET E7=FNP(E3+I3+S2,63)
1680 PRINT
1690 IF MK=12 THEN 1980
1700 PRINT
1710 PRINT"===== "
1720 LET M=1
1730 LET Y=Y+1
1740 PRINT
1750 PRINT "FOR";Y-1;": "
1760 LET T1=23
1770 PRINT " BEGINNING PRINCIPAL";
1780 LET A9=FNP(A9,T1)
1790 PRINT
1800 PRINT " ENDING PRINCIPAL";
1810 LET A8=A1-E1
1820 IF A8>=0 THEN 1840
1830 LET A8=0
1840 LET A8=FNP(A8,T1)
1850 PRINT
1860 PRINT" PRINCIPAL REPAYMENT";
1870 LET C2=FNP(C2,T1)
1880 PRINT
1890 PRINT" TOTAL INTEREST PAID";
1900 LET C1=FNP(C1,T1)
1910 PRINT
1920 IF Y8=Y9 THEN 2030
1930 LET Y8 = Y8 + 1
1940 IF A1=0 THEN 2030
1950 LET C1=C2=0
1960 LET A9=A1-E1
1970 PRINT
1980 LET A1=A1-E1
1990 IF A1=0 THEN 1700
2000 LET A2=A2-E2
2010 PRINT
2020 GO TO 1240
2030 PRINT
2040 PRINT
2050 PRINT"===== "
2060 END

```

PLEASE LIST THIS PROGRAM FOR INSTRUCTIONS.
 ENTER DATA

BORROWER'S NAME ?H.L.GREEN
 ORIGINAL PRINCIPAL ?30000
 SBA SHARE ?40
 BANK RATE ?8.5
 SBA RATE ?6
 LOAN DATED (MO, YR) # ?9, 1972
 MONTHLY PAYMENT ?500
 SERVICE FEE (%) ?2.5
 FOR HOW MANY CALENDAR YEARS DO YOU WISH THE SCHEDULE ?1

*****SBA LOAN SCHEDULE*****

BORROWER: H.L.GREEN
 AMOUNT OF LOAN 30000.00
 SBA PARTICIPATION: 40%
 BANK INTEREST RATE: 8.5%
 SBA INTEREST RATE: 6%
 MONTHLY PAYMENT: 500.00

	BEGINNING PRINCIPAL	INTEREST PAYMENT	PRINCIPAL REPAYMENT	SERVICE FEE	TOTAL PAYMENT
PMT # 1 OCT, 1972					
TOTAL LOAN	30000.00	187.50	312.50		500.00
SBA SHARE	12000.00	60.00	125.00	-2.50	182.50
BANK SHARE	18000.00	127.50	187.50	2.50	317.50

PMT # 2 NOV, 1972					
TOTAL LOAN	29687.50	185.54	314.46		500.00
SBA SHARE	11975.00	59.37	125.78	-2.47	182.68
BANK SHARE	17912.50	126.17	188.68	2.47	317.32

PMT # 3 DEC, 1972					
TOTAL LOAN	29373.04	183.59	316.41		500.00
SBA SHARE	11749.22	58.75	126.57	-2.45	182.87
BANK SHARE	17623.82	124.84	189.84	2.45	317.13

FOR 1972:
 BEGINNING PRINCIPAL 30000.00
 ENDING PRINCIPAL 29056.63
 PRINCIPAL REPAYMENT 943.37
 TOTAL INTEREST PAID 556.63

TIC TAC:

DESCRIPTION

This is a computer simulation of the game TIC TAC TOE. The game is played on a standard 3 x 3 board. The computer randomly selects who will make the first play and is your opponent. This is a very simple version of a very old, old game and can provide hours of entertainment for both the novice player and the expert.

USERS

This is a very good starter game for the novice computer game player and its a nice change of pace for the expert who has been frustrated by a few of the more complex games.

INSTRUCTIONS

Load the program into memory and type RUN. The game board will be printed at the beginning of the first game only. It may be necessary to keep a copy of the game board until its arrangement becomes familiar. To play you type in the number of the square where you would like to place your X. The computer will tell you which square it will occupy based on prior moves.

LIMITATIONS

Line 1030 contains a TAB () statement, 1070 MAT Read, 1080 MAT C = ZER and 1090 MAT B = ZER. This program uses two dimensional arrays for operation; see line 1060. The source code stores in 2 Bytes of memory and the program will execute in 3K Bytes of memory, excluding the space required for your Basic compiler. TIC TAC should execute in most 8K Basics with little or no problems.

TICTAC

```

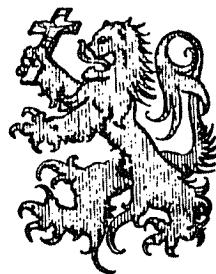
1010 PRINT "----- TIC TAC TOE -----"
1020 PRINT
1030 PRINT TAB(16);"1  2  3"
1040 PRINT TAB(16);"8  9  4"
1050 PRINT TAB(16);"7  6  5"
1060 DIM T(8,3),U(9,4),C(9),B(9)
1070 MAT READ T,U
1080 MAT C=ZER
1090 MAT B=ZER
1100 N=0
1110 IF RND(-1)<.5 THEN 1150
1120 PRINT "I WILL MOVE FIRST"
1130 PRINT
1140 GOTO 1380
1150 PRINT "YOU WILL MOVE FIRST"
1160 PRINT
1170 PRINT "YOUR MOVE"
1180 INPUT M
1190 F=-1
1200 IF M<>INT(M) THEN 1360
1210 IF M<1 THEN 1360
1220 IF M>9 THEN 1360
1230 IF B(M)<>0 THEN 1360
1240 B(M)=F
1250 FOR J=1 TO 4
1260 P=U(M,J)
1270 IF P=0 THEN 1310
1280 C(P)=C(P)+F
1290 IF C(P)=-3 THEN 1490
1300 IF C(P)=3 THEN 1420
1310 NEXT J
1320 N=N+1
1330 IF N=9 THEN 1510
1340 IF F=1 THEN 1170
1350 GOTO 1380
1360 PRINT "ILLEGAL MOVE--TRY AGAIN"
1370 GOTO 1170
1380 GOSUB 1530
1390 PRINT "      I      MOVE TO ";M
1400 F=1
1410 GOTO 1240
1420 PRINT"AND      I      WIN"
1430 PRINT
1440 PRINT
1450 PRINT "ANOTHER GAME";
1460 INPUT A1$
1470 IF LEN(A1$)=3 THEN 1080

```

```

1480 STOP
1490 PRINT "CONGRATULATIONS, YOU BEAT ME"
1500 GOTO 1430
1510 PRINT "THIS GAME IS A DRAW"
1520 GOTO 1430
1530 FOR P=1 TO 8
1540 IF C(P)=2 THEN 1600
1550 NEXT P
1560 FOR P=1 TO 8
1570 IF C(P)=-2 THEN 1600
1580 NEXT P
1590 GOTO 1640
1600 FOR I=1 TO 3
1610 M=T(P,I)
1620 IF B(M)=0 THEN 1790
1630 NEXT I
1640 FOR S=1 TO 9
1650 U(S)=0
1660 IF B(S)<>0 THEN 1720
1670 FOR J=1 TO 4
1680 P=U(S,J)
1690 IF P=0 THEN 1710
1700 U(S)=U(S)+1+ABS(C(P))
1710 NEXT J
1720 NEXT S
1730 U=0
1740 FOR S=1 TO 9
1750 IF U(S)<=U THEN 1780
1760 U=U(S)
1770 M=S
1780 NEXT S
1790 RETURN
1800 DATA 1,2,3,8,9,4,7,6,5,1,8,7,2,9,6,3,4,5,1,9,5,7,9,3
1810 DATA 1,4,7,0,1,5,0,0,1,6,8,0,2,6,0,0,3,6,7,0,3,5,0,0
1820 DATA 3,4,8,0,2,4,0,0,2,5,7,8
1830 END

```



TICTAC

-----TIC TAC TOE-----

1	2	3
9	9	4
7	6	5

YOU WILL MOVE FIRST

YOUR MOVE?3

I MOVE TO 9

YOUR MOVE?7

I MOVE TO 1

YOUR MOVE?5

I MOVE TO 6

YOUR MOVE?2

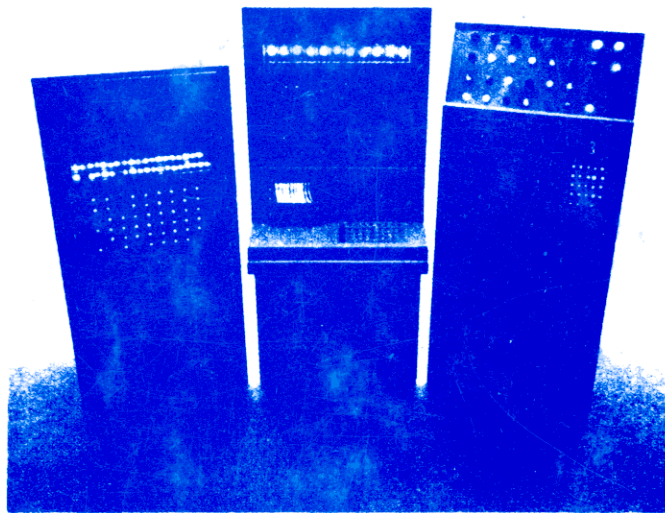
I MOVE TO 4

YOUR MOVE?8

THIS GAME IS A DRAW

ANOTHER GAME?NO

RELIABLE COMPUTER SOFTWARE



FOR YOUR DOWN TO EARTH TASKS