DOCUMENT RESUME

ED 063 783 EM 009 988

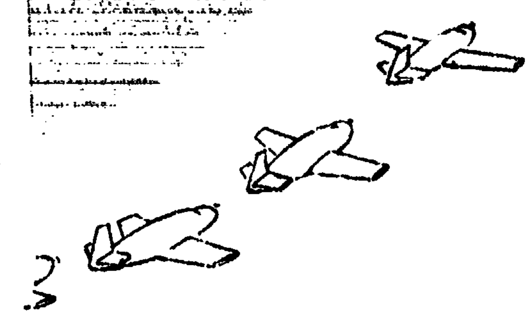
TITLE	Project Solo; Newsletter Number 22.
INSTITUTION	Pittsburgh Univ., Pa. Project Solo.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	1 Jun 72
NOTE	93p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction; *Computer Graphics; Computer Programs; *Educational Games; *Peer Teaching; *Simulation; Student Developed Materials
IDENTIFIERS	*Project Solo

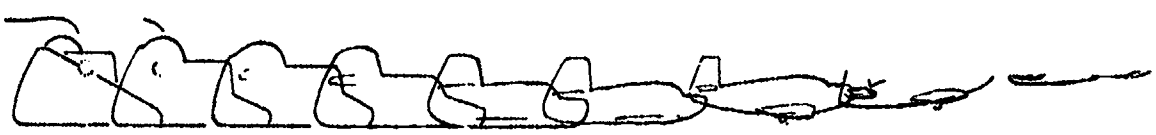
ABSTRACT

Four new Project Solo modules are presented. A module on computer graphing with an x-y plotter explains the format for input to the computer, demonstrates the effect of scaling, and gives several on-line exercises to produce graphs. Several student-written programs in elementary and advanced mathematics, French, and U.S. history show how peer teaching can be accomplished with a computer. Four computer games are presented: hide-and-seek, NIM, MODULO, and space war. Four simulations--lunar landing module, crazy eights, rectangular billiards, and eliptical billiards--are also offered. (JY)

PROJECT SOLO

AN EXPERIMENT IN REGIONAL COMPUTING*
FOR SECONDARY SCHOOL SYSTEMS





University of Pittsburgh • Department of Computer Science • Pittsburgh, Pennsylvania 15213

Newsletter No. 22 June 1, 1972

Computing Builds Strong Minds Ten Ways

Or is it twelve? Some recent research by Emilie Zielinski (author of the Project Solo modules on matrices) revealed that the distinctions made in educational computing (and its benefits) have grown sharper with the years. This is especially true in places where the full power of computing has been made available to students. Some impressions on where this is taking us, and where the real action will be in a few years, are given in the paper "Guidelines for Computer-Related Educational Innovation" (available upon request). Some examples of how these distinctions reveal themselves in the real school world today are given in the enclosed modules. A new 16 mm. film ("My Computer Understands Me") which explores this subject will be ready this fall.

Computing and Peer Teaching

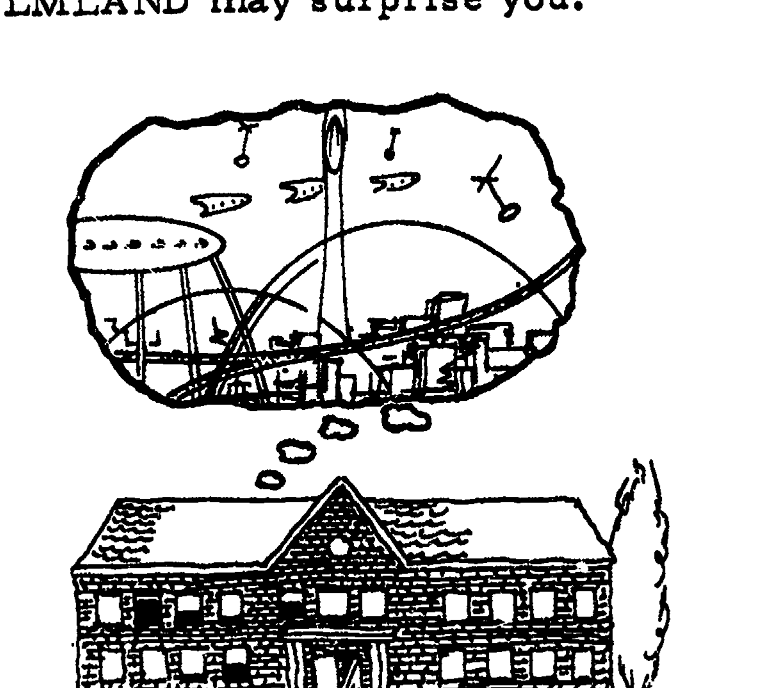
The material in this booklet comes from the students of Rosemarie Kavanagh, Mathematics teacher at Fox Chapel High School. The significance of this work is very deep. It has both educational and economic implications that deserve further study. All 14 programs in this booklet are student authored.

Computer Games and Computer Simulations

Distinctions can be made between games and simulations, although the programs students write in these areas often combine both concepts. The real fun begins when students modify, or better yet create the programs behind simulations and games. For this reason, we believe that listings of such programs should always be furnished to students. The authorship of LMLAND may surprise you.

Son of Project Solo

Project Solo funding terminates at the end of June. We were tempted to include a questionnaire with this newsletter to find out what our readers would like to see from future projects of this sort. Instead, we'd like to (a) seriously invite free-form feedback in the form of a short (or long) note from you or your students, and (b) promise to keep you informed on some quite new directions we will be exploring. To say we are excited about these would be the understatement of the year.



*Supported in part by NSF Grant GJ-1077.

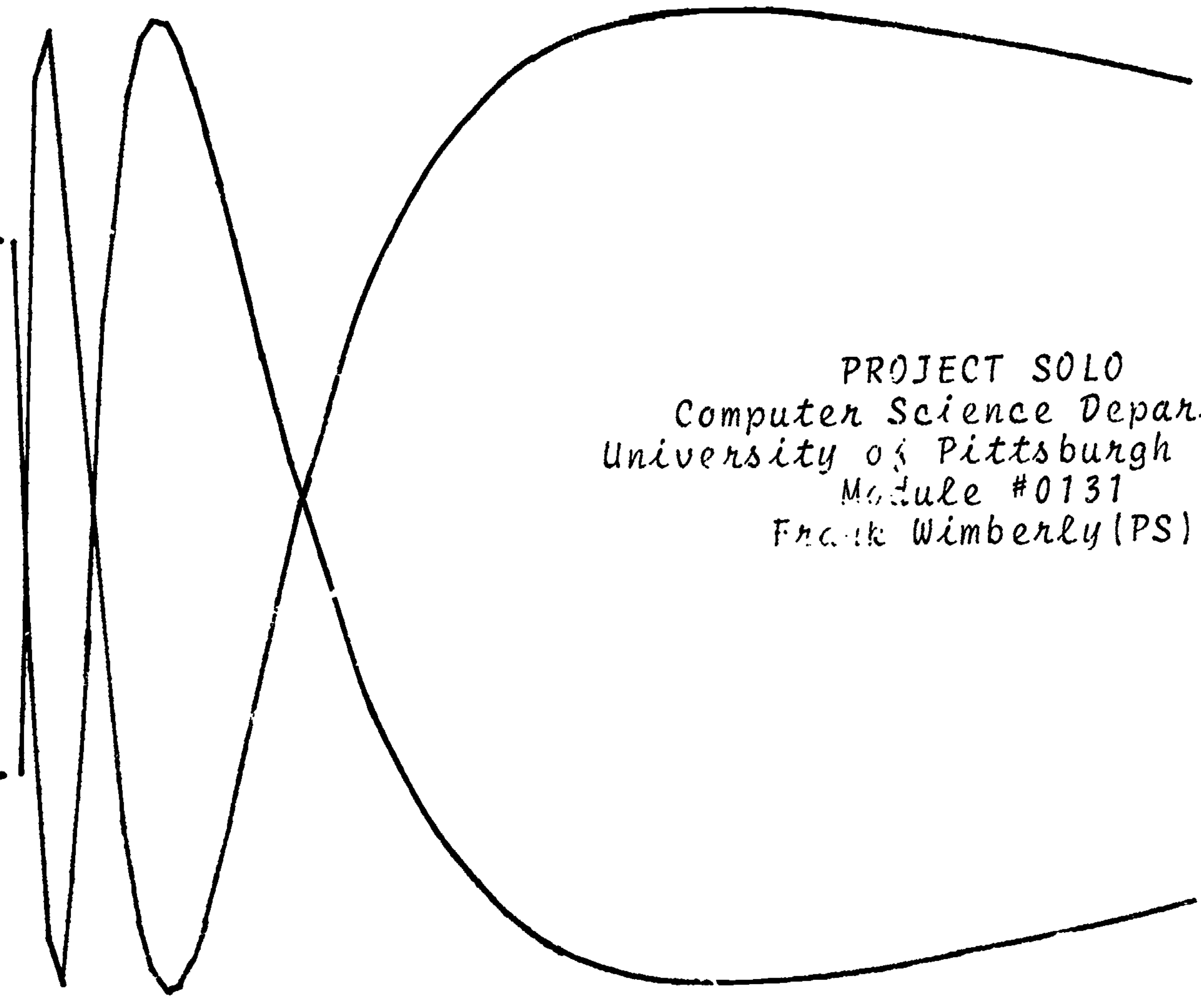
Computer Graphing with an X-Y Plotter

In this module, you will learn to use the plotter in several stages.

- You will study the format of input to the plotter and then do some off-line exercises which illustrate the various forms that input can take.
- You will learn about scaling, which has to do with getting the graph to have the right shape and size. A scaling example will be given which will also introduce the use of a BASIC program to do the typing.
- The most interesting activity in the module involves several on-line exercises which produce graphs like the ones on the cover.

Y = -SIN(1/X)

Y = SIN(1/X)



PROJECT SOLO
Computer Science Department
University of Pittsburgh (15213)
Module #0131
Frank Wimberly (PS)

COMPUTER GRAPHING WITH AN X-Y PLOTTER

Drawing the graph of a function can often give much more insight into its properties than looking at a formula which defines it or at a table of its values. It is hard to overstate the usefulness of graphs in estimating zeros of functions, maximum and minimum points, points of inflection, etc. Graphs make it possible for you to use your geometric intuition in considering how one variable is related to another. For a discussion of graphs and coordinate systems, See Dolciani et al. Modern Introductory Analysis Chapter 5 (Houghton Mifflin Co., Boston, Massachusetts, 02107).

Because graphs are one of the most valuable forms that computer output can take, it would be most helpful to have a device capable of converting the standard form of output (printed numbers) to lines or points on a sheet of graph paper. The Hewlett-Packard Graphic Plotter is such a device. It transforms lists of number pairs which are typed on a terminal (either by hand or by a computer program) into graphs.

In this module, you will learn to use the plotter in several stages.

You will study the format of input to the plotter and then do some off-line exercises which illustrate the various forms that input can take.

You will learn about scaling, which has to do with getting the graph to have the right shape and size. A scaling example will be given which will also introduce the use of a BASIC program to do the typing.

The most interesting activity in the module involves several on-line exercises which produce graphs like the ones on the cover.

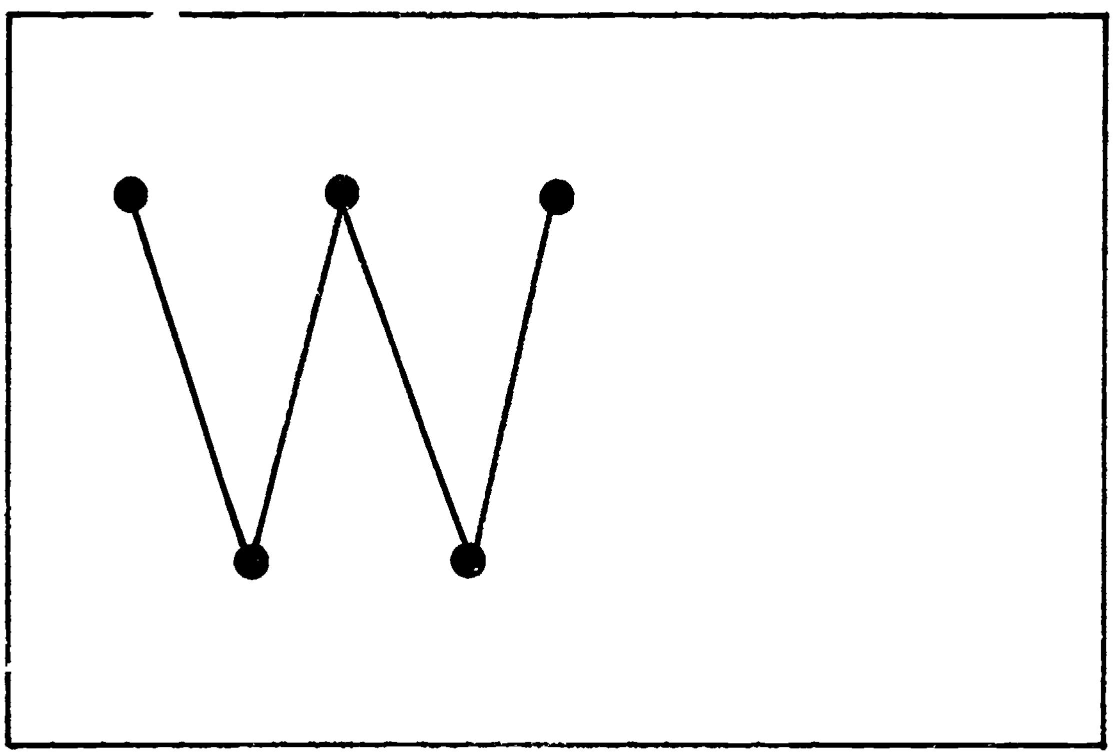
At this point there are several directions that you can take: Module #0132 deals with the problem of graphing the paths of Pirates on the high seas. Module #0134 explores the use of computer graphing in solving interpolation/extrapolation problems. Module #0133 deals with the interesting topic of drawing three-dimensional graphs. Once you have gotten started by doing some of the problems suggested in this module, it should be easy to come up with interesting ideas of your own.

PLOTTER INPUT FORMAT

The plotting area of the Hewlett-Packard Plotter should be thought of as an xy plane where $0 \leq x \leq 9999$ and $0 \leq y \leq 9999$. The point (0,0) is the lower left corner of the plotting area and the point (9999,9999) is the upper right corner. The pen can move to any point in that area whose coordinates can be specified by a pair of integers in those ranges.

NOTE: There are 10,000 x 10,000 = 100,000,000 such points. If the plotter area is 10" x 15" (150 square inches) that means that there are about 667,000 plottable points per square inch. Do you think you could see the difference between an unplottable point (say (4628.9079342, 9175.210365)) and the nearest plottable point ((4629, 9175))?

Basically what happens is that whenever a statement of the form $\boxed{\text{PRINT X,Y}}$ is executed on a terminal (which is connected to the plotter), the pen moves to the point (X,Y) in the plotting area. There are two basic modes of plotting as illustrated in Figure 1. In the first case (Figure 1a), the terminal output contains a line consisting of the letters PLTL (plot lines) followed by a list of number pairs. The plotted output consists of lines drawn between the points represented by the number pairs. The code PLTT shown on the last line means plot terminate.



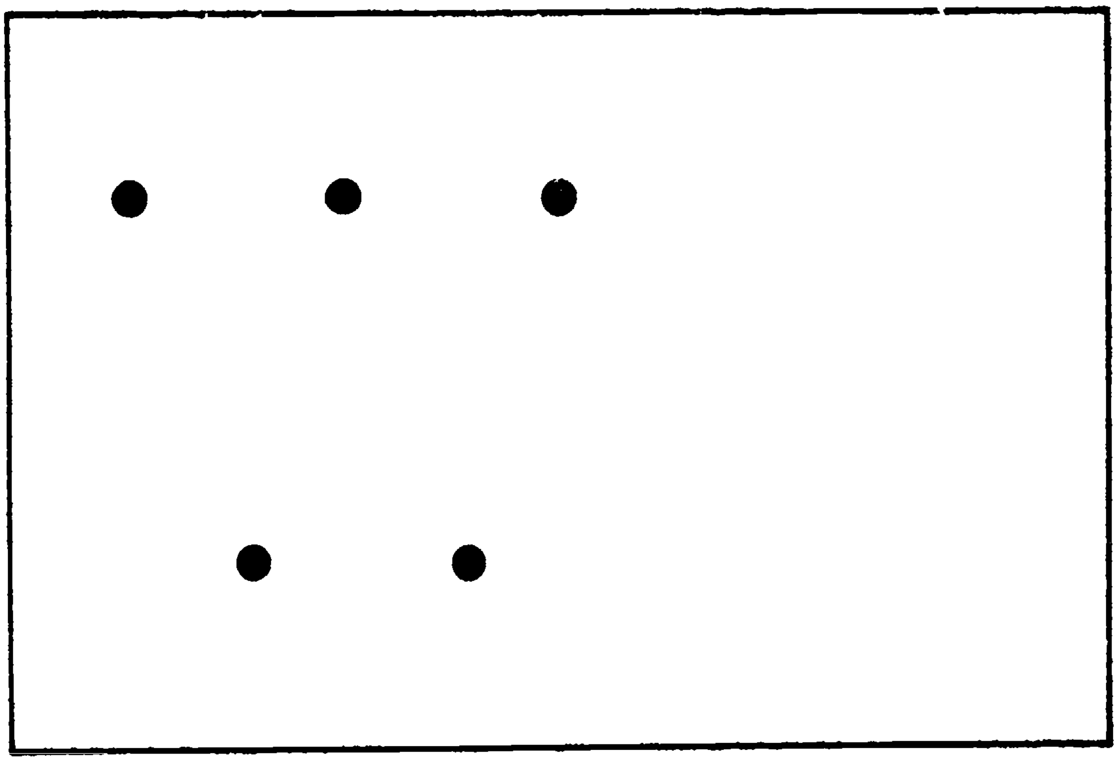
PLOTTER OUTPUT

```
PLTL
1000   7000
2000   3000
3000   7000
4000   3000
5000   7000
PLTT
```

TERMINAL OUTPUT

Figure 1a.

The other mode (Figure 1b) plots points (PLTP) without drawing the lines between them. That is, it puts a dot on the graph paper at the location corresponding to each of the number pairs.



PLOTTER OUTPUT

```
PLTP
1000    7000
2000    3000
3000    7000
4000    3000
5000    7000
PLTT
```

TERMINAL OUTPUT

Figure 1b.

The following list explains the form of the data that is fed to the plotter. In off-line mode the data is fed to the plotter directly by typing on the teletype. In on-line mode it is fed to the plotter by PRINT statements. After either PLTL or PLTP is typed on the terminal, numerical data should be typed in the following format:

A. The coordinates of each point should be placed on a different line.

B. Each value of x (first coordinate) or y (second coordinate) should be a non-negative integer less than or equal to 9999 and may not have a decimal point.

C. Any number of spaces may be typed before the x coordinate.

D. The first space after the x coordinate acts as the separator between the x and y coordinates.

E. Any number of spaces may be typed between the x coordinate and the y coordinate.

F. The y coordinate may be followed by either:

1. <u>A space</u>. In this case the rest of the line up to the carriage return is ignored. The carriage return causes the pen to move to the point whose coordinates were typed.

2. <u>An up-arrow (↑)</u>. If an up-arrow immediately follows the y-coordinate, the pen moves to the point while lifted from the paper.

NOTE: *In PLTP mode the pen always moves while lifted so in this case the up-arrow can serve no purpose. The purpose it can serve in PLTL mode will be illustrated in the exercise below.*

<u>Off-Line Exercise</u>

For this exercise you need a plotter connected to a terminal but no computer. That is, you will be using the terminal in local (off-line) mode.

Prepare the plotter for plotting (See Appendix A) and type the following on the terminal:

```
PLTL

1333      3000
2666      3000
2666      5000
1333      5000
1333      7000
2666      7000
3333      7000↑
4666      7000
4666      7000
3333      3000
```

```
3333      7000
5333      7000↑
5333      3000
6666      3000
7333      7000↑
7333      3000
8666      3000
8666      7000
7333      7000
PLTT
```

<u>Scaling</u>

Suppose you wanted to graph the function $y = x^3$ for $-1 \le x \le 1$ on the Hewlett-Packard Plotter. The corresponding values of y also have the range $-1 \le y \le 1$. Recall that no points in the plotting area ($0 \le x \le 9999$ and $0 \le y \le 9999$) have negative coordinates. Furthermore the graph of a function over such a small interval (length 2) would appear to be just a dot anyway. This suggests a couple of questions:

a. How do you draw graphs which contain points with negative coordinates?

b. How do you graph a function so that the graph is a reasonable size?

The answer lies in the fact that multiplying every value of x by a constant and adding another constant and doing the same for y doesn't change the basic shape of the graph.

The following points are all on the graph of the function $y = x^3$:

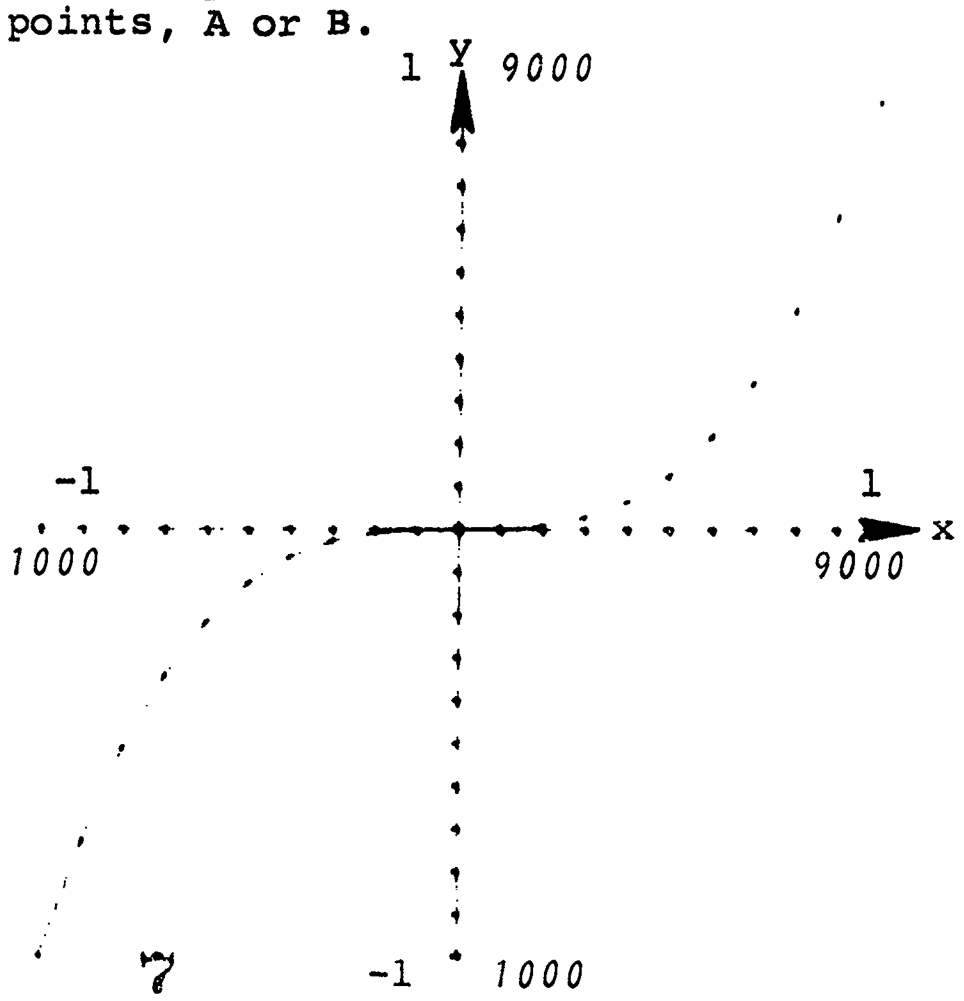
(-1, -1)	(.1, .001)	
(-.9, -.729)	(.2, .008)	
(-.8, -.512)	(.3, .027)	
(-.7, -.343)	(.4, .064)	
(-.6, -.216)	(.5, .125)	SET A
(-.5, -.125)	(.6, .216)	
(-.4, -.064)	(.7, .343)	
(-.3, -.027)	(.8, .512)	
(-.2, -.008)	(.9, .729)	
(-.1, -.001)	(1, 1)	
(0, 0)		

If we multiply every value of x by 4000 and add 5000, and then ignore all digits after the decimal point, and do the same for every value of y we get the following set of points:

(1000, 1000)	(5400, 5003)	
(1400, 2083)	(5800, 5031)	
(1800, 2951)	(6200, 5107)	
(2200, 3627)	(6600, 5255)	
(2600, 4135)	(7000, 5500)	SET B
(3000, 4500)	(7400, 5863)	
(3400, 4743)	(7800, 6371)	
(3800, 4891)	(8200, 7047)	
(4200, 4967)	(8600, 7915)	
(4600, 4995)	(9000, 9000)	
(5000, 5000)		

If you look at Figure 2 you can see that depending on which axis labels you look at you can interpret the graph as belonging to either of the two sets of points, A or B.

Figure 2.



The important point to notice is that the coordinates of the second set of points are in the right range for producing a graph on the Hewlett-Packard Plotter.

The following is a program in BASIC which draws the graph of the function $y = x^3$ for $-1 \leq x \leq 1$ using the above scaling method. Run the program with the plotter attached to your terminal then compare the listing of the program, the output on the terminal, and the graph.

```
10 PRINT "PLTL"
20 FOR Z=-1 TO 1 STEP .1
25 LET X=Z
30 LET Y = X↑3
40 LET X=X*4000+5000
50 LET Y = Y*4000 + 5000
60 PRINT INT(X), INT(Y)
70 NEXT Z
80 PRINT "PLTT"
```

This draws the graph of $y = x^3$.

```
90 PRINT "PLTL"
100 LET Y = 5000
110 FOR X = 1000 TO 9000 STEP 400
115 PRINT X,Y
120 PRINT X, Y - 50
130 PRINT X, Y
140 NEXT X
150 PRINT "PLTT"
```

This puts calibration marks on the X axis and draws the X axis.

```
160 PRINT "PLTL"
170 LET X = 5000
180 FOR Y = 1000 TO 9000 STEP 400
185 PRINT X,Y
190 PRINT X - 50 , Y
200 PRINT X,Y
210 NEXT Y
220 PRINT "PLTT"
```

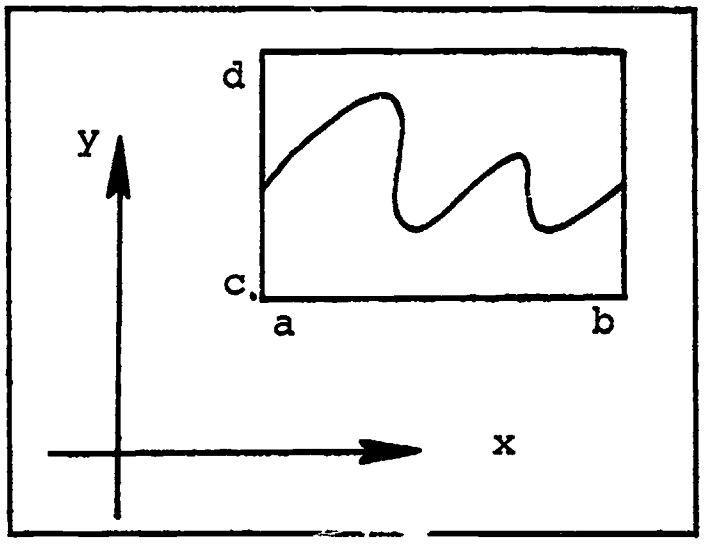
This puts calibration marks on the Y axis and draws the Y axis.

```
230 PRINT "PLTL"
240 PRINT 950, 4850
250 PRINT 975, 4850
260 PRINT 1000,4800:"↑"
270 PRINT 1000,4900
280 PRINT 9000,4800:"↑"
290 PRINT 9000,4900
300 PRINT 4775,1000:"↑"
310 PRINT 4800,1000
320 PRINT 4850,950:"↑"
330 PRINT 4850,1050
340 PRINT 4850,8950:"↑"
350 PRINT 4850,9050
360 PRINT "PLTT"
370 END
```

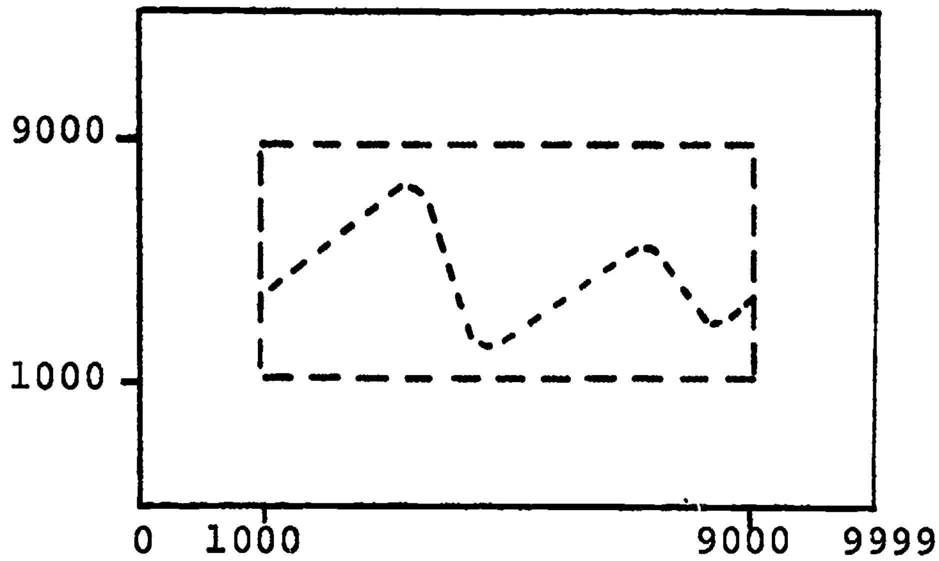
What does this do?

Some Generalizations:

To see how to determine the scaling constants, consider the situation represented below:



PLOTTER PAPER



You want to graph a function where x and y have ranges $a \le x \le b$ and $c \le y \le d$ so that the graph fits in that portion of the plotted area where $1000 \le x \le 9000$ and $1000 \le y \le 9000$. To do this you want to find constants K_x, K_y, J_x, and J_y such that:

$a \le x \le b$ exactly when $1000 \le K_x * x + J_x \le 9000$

$c \le y \le d$ exactly when $1000 \le K_y * y + J_y \le 9000$

It turns out that the following formulas accomplish this:

$$K_x = (9000 - 1000)/(b - a)$$

$$K_y = (9000 - 1000)/(d - c)$$

$$J_x = 5000 - K_x(a + b)/2$$

$$J_y = 5000 - K_y(c + d)/2$$

You should apply them to the problem for which the above program was written to verify that $K_x = K_y = 4000$ and $J_x = J_y = 5000$.

Naturally there is nothing special about the numbers 1000 and 9000; if you wanted a slightly larger graph you could use 500 and 9500.

ON-LINE EXERCISES

1. Write a BASIC program to graph the function $y=x^3-x^2+x-1$ on the interval $-3\le x\le 3$.

   HINT: $-34\le y\le 20$.

2. Write a program which uses the random number function to draw random size squares at random points on the graph paper.

3. Long Term Project: Write a program which reads in a string of characters and then prints the string on the plotter.

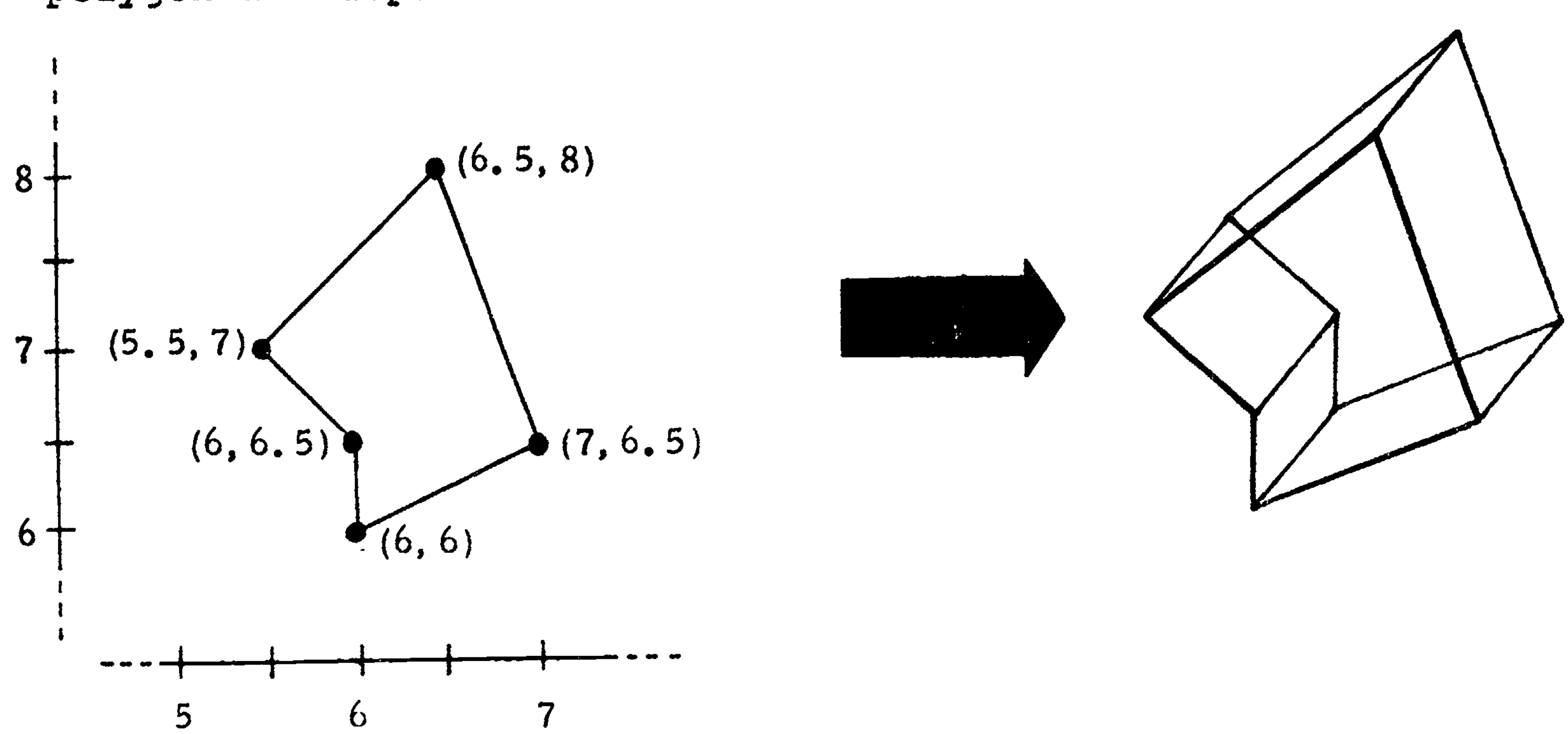
4. For those who are familiar with circular functions:

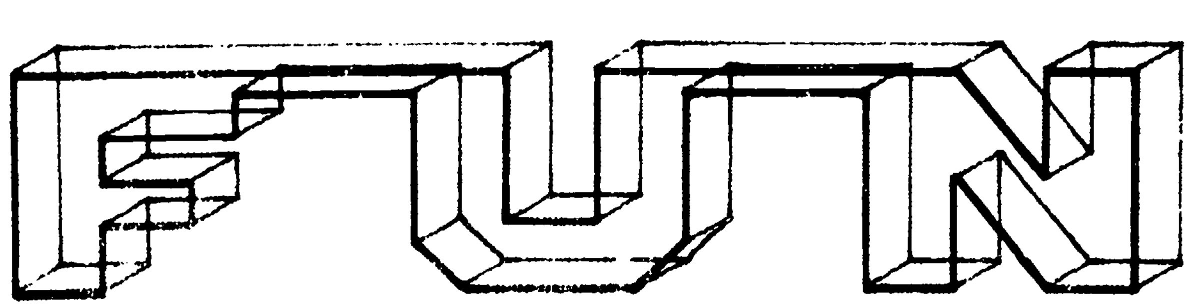
   Find all values of x in the interval $.1\le x\le 1$ which satisfy the equation $\sin(1/x)=-\sin(1/x)$.

   HINT: Write a program which will graph both functions on the same sheet and estimate the values of x where the graphs intersect. See the cover of this module.

5. For the artistically inclined--Use different colored pens, the random generator, and your knowledge of analytic geometry to produce some examples of computer-generated art. (See Project Solo Newsletters No. 12, 13, 15, and 20 for some ideas that you could combine.)

6. Can you write a program that accepts the vertices of polygons as input, and then produces an iso-metric (3-D) drawing of the polygon as output?





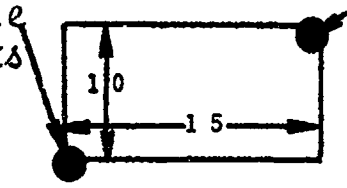
APPENDIX A

Assuming that you have a Hewlett-Packard Plotter connected to a teletype the following steps will prepare the plotter for plotting:

1. Turn teletype on (local for offline work; line for on-line work).
2. Push power button on the plotter ON.
3. Place paper on plotting area.
4. Push chart hold button and wait until paper sticks to plotting area.
5. Push lower left button and adjust lower left knobs to determine the (0,0) point. Be careful not to turn knobs past the point where the pen stops moving.
6. Push upper right button and adjust upper right knobs to determine the (9999,9999) point.

NOTE: Unless you make the plotting area square by putting the upper right point exactly as far to the right as it is above the lower left point, your graph may be slightly elongated in the direction of one or the other axis. This is no problem unless you want your circles to be perfectly round rather than somewhat elliptical.

If you wish to use the full 10 x 15 paper and still get "circular" circles, set the pen here and here, but change all your PRINT statements so that X is multiplied by 0.666667.



PEER TEACHING with COMPUTERS

The following programs were all written by students at Fox Chapel Area High School. These students wrote their programs with their fellow students' needs in mind. Each program is a tutorial aimed at helping younger learners master a new skill. This booklet is divided into three areas.

The first area covers elementary arithmetic skills which include fractions, measurement conversion, volumetric conversion, subtraction, multiplication, and addition. The second area, advanced mathematics, contains tutorials on scientific notation, algebra and logarithms. The third area covers non-mathematical subjects. This includes a U.S. history questionnaire, French grammar & vocabulary, and French history & literature.

Each program has a short introduction written by the student programmers followed by a sample run. Those interested in listings can obtain them by contacting Project Solo.

Table of Contents

PROJECT SOLO / Computer Science Dept. / Univ. of Pittsburgh (15213)
Modules #0205-0218 / Student authors from Fox Chapel High School.

ARITHMETIC

Tables (#0205)

This is a program that was used with a special education class to ease the boredom of both teacher and student being drilled on multiplication facts. The students responded enthusiastically to it.

Woody Lauer, 12th Grade

```
>RUN

WHAT IS YOUR NAME
?WOODY LAUER
WOODY LAUER, WHAT IS THE NUMBER OF THE TABLE YOU
WISH TO CHECK
?6
WHAT DOES 5 X 6=?
?26
YOU'RE WRONG, WOODY LAUER;
TRY AGAIN.
WHAT DOES 5 X 6=?
?30
THAT'S RIGHT
WHAT DOES 0 X 6=?
?12
YOU'RE WRONG, WOODY LAUER;
TRY AGAIN.
WHAT DOES 0 X 6=?
?34
YOU'RE WRONG, WOODY LAUER;
THE CORRECT ANSWER IS    0.
WHAT DOES 1 X 6=?
?0
YOU'RE WRONG, WOODY LAUER;
TRY AGAIN.
WHAT DOES 1 X 6=?
?6
GOOD
WHAT DOES 6 X 6=?
?36
GOOD
WHAT DOES 3 X 6=?
?18
GOOD
WHAT DOES 4 X 6=?
?24
RIGHT
WHAT DOES 2 X 6=?
?12
THAT'S RIGHT
```

Pairs (#0206)

This program can best be utilized by a third or fourth grader. It tests their ability to add and multiply. The computer randomly selects a number between 1 and 90. The child must find either four different pairs of addends whose sum equals the number or a pair of integers whose product yields the number. The student has the option of finding additional pairs of multipliers.

Ann Conner, 12th Grade

```
>RUN
WHAT IS YOUR FULL NAME?
?ANN CONNER
I AM THINKING OF TWO NUMBERS WHICH WHEN ADDED
EQUALS  17. PLEASE TYPE TWO NUMBERS
THAT YOU CAN THINK OF THAT EQUAL  17
?1
?16
THAT IS CORRECT.
CAN YOU THINK OF TWO MORE NUMBERS?
PLEASE TYPE THEM.
?2
?14
THAT IS NOT CORRECT THE ANSWER YOU GET
WHEN ADDING 2 TO  14 IS 16
CAN YOU THINK OF TWO MORE NUMBERS?
PLEASE TYPE THEM.
?2
?15
THAT IS CORRECT.
CAN YOU THINK OF TWO MORE NUMBERS?
PLEASE TYPE THEM.
?3
?16
THAT IS NOT CORRECT THE ANSWER YOU GET
WHEN ADDING 3 TO  16 IS 19
NOW I AM THINKING OF TWO NUMBERS WHICH
WHEN MULTIPLIED TOGETHER EQUAL 61
?61
?1
THAT WAS VERY GOOD YOU ARE CORRECT.
CAN YOU THINK OF TWO MORE NUMBERS WHICH WHEN
MULTIPLIED TOGETHER EQUAL  61 PLEASE TYPE THEM.
?YES
?2
?61
THAT WAS NOT RIGHT. WHEN YOU MULTIPLY 2 BY
 61 YOU GET 122
CAN YOU THINK OF TWO MORE NUMBERS WHICH WHEN
MULTIPLIED TOGETHER EQUAL  61 PLEASE TYPE THEM.
?NO
```

<u>Subtraction</u> (#0207)

This program is designed to enable students to practice subtraction. However, it is more of a challenge to students than just knowing how to subtract. The student is given the difference between two numbers and is asked to find both numbers. He is only given five chances to reason both numbers correctly, so proficiency in substraction is necessary to get both numbers correct.

Terry Wise, 12th Grade

```
>RUN
TYPE YOUR FULL NAME
?TERRY WISE
I AM THINKING OF TWO NUMBERS BETWEEN 40 AND 70
THE DIFFERENCE BETWEEN THEM IS 20
TYPE A NUMBER THAT YOU THINK IS THE LARGER NUMBER
?65
NOW TYPE THE SMALLER NUMBER
?45
 65- 45= 20 BUT THOSE ARE NOT THE CORRECT
NUMBERS
TYPE A NUMBER THAT YOU THINK IS THE LARGER NUMBER
?66
NOW TYPE THE SMALLER NUMBER
?46
 66- 46= 20 BUT THOSE ARE NOT THE CORRECT
NUMBERS
TYPE A NUMBER THAT YOU THINK IS THE LARGER NUMBER
?67
NOW TYPE THE SMALLER NUMBER
?47
 67- 47= 20 BUT THOSE ARE NOT THE CORRECT
NUMBERS
TYPE A NUMBER THAT YOU THINK IS THE LARGER NUMBER
?64
NOW TYPE THE SMALLER NUMBER
?44
 64- 44= 20 BUT THOSE ARE NOT THE CORRECT
NUMBERS
TYPE A NUMBER THAT YOU THINK IS THE LARGER NUMBER
?63
NOW TYPE THE SMALLER NUMBER
?43
GREAT, YOU GOT BOTH NUMBERS RIGHT
IF YOU WOULD LIKE TO TRY ANOTHER PROBLEM, TYPE THE WORD
YES
?NO
GOODBYE TERRY WISE
```

Addition of Fractions (#0208)

This program was used in a general mathematics and a refresher arithmetic course to introduce the addition of fractions before teaching a technique for finding the least common denominator.

Mark Pilant, 10th Grade

```
>RUN
HELLO, MY NAME IS CHARLIE.
TYPE YOUR NAME
?KELLY
TYPE THE NUMERATOR OF THE FIRST FRACTION
?1
TYPE THE DENOMINATOR OF THE FIRST FRACTION
?2
TYPE THE NUMERATOR OF THE SECOND FRACTION
?3
TYPE THE DENOMINATOR OF THE SECOND FRACTION
?4
TYPE THE DENOMINATOR OF YOUR ANSWER
?6
INCORRECT. FIRST FIND A COMMON DENOMINATOR BY
MULTIPLYING BOTH DENOMINATORS TOGETHER.
TYPE YOUR ANSWER TO THIS MUCH OF THE PROBLEM.
?6
INCORRECT. FIRST FIND A COMMON DENOMINATOR BY
MULTIPLYING BOTH DENOMINATORS TOGETHER.
TYPE YOUR ANSWER TO THIS MUCH OF THE PROBLEM.
?8

MULTIPLY THE NUMERATOR AND DENOMINATOR OF THE FIRST FRACTION
BY THE DENOMINATOR OF THE SECOND FRACTION
TYPE THE NUMERATOR OF YOUR NEW FRACTION
?2
INCORRECT
MULTIPLY THE NUMERATOR AND DENOMINATOR OF THE FIRST FRACTION
BY THE DENOMINATOR OF THE SECOND FRACTION
TYPE THE NUMERATOR OF YOUR NEW FRACTION
?4
         TYPE THE DENOMINATOR OF YOUR NEW FRACTION.
?8
    MULTIPLY THE NUMERATOR AND THE DENOMINATOR OF SECOND
FRACTION BY THE DENOMINATOR OF THE FIRST FRACTION
TYPE THE NUMERATOR OF YOUR NEW FRACTION
?5
INCORRECT
    MULTIPLY THE NUMERATOR AND THE DENOMINATOR OF SECOND
FRACTION BY THE DENOMINATOR OF THE FIRST FRACTION
TYPE THE NUMERATOR OF YOUR NEW FRACTION
?6
TYPE THE DENOMINATOR OF YOUR NEW FRACTION
?8
NOW ADD THE TWO NUMERATORS TOGETHER
TYPE YOUR ANSWER
?5
```

```
NOW ADD THE TWO NUMERATORS TOGETHER
TYPE YOUR ANSWER
? 10
NOW YOU HAVE FIGURED THE NUMERATOR OF YOUR FINAL ANSWER
WHICH IS  10
THE DENOMINATOR OF YOUR FINAL ANSWER IS  8
IF YOU WANT TO TRY ANOTHER PROBLEM TYPE THE WORD YES
```

Measurements (#0209)

This is a program used in a refresher arithmetic class. It forces the student to reduce his answer to the simplest form. There are no tutorial features in it because these students have too much difficulty reading and interpreting instructions.

Don Salvin, 12th Grade

```
>RUN
THIS IS A PROGRAM TO ADD SEVERAL SETS OF DENOMINATE
NUMBERS.
HOW MANY SETS OF NUMBERS DO YOU WISH TO ADD?
?2
TYPE THE NUMBER OF YARDS IN THE FIRST SET OF MEASUREMENTS.
?8
TYPE THE NUMBER OF FEET IN THE FIRST SET OF MEASUREMENTS.
?0
TYPE THE NUMBER OF INCHES IN THE FIRST SET OF MEASUREMENTS.
?21
TYPE THE NUMBER OF YARDS IN THE SECOND SET OF MEASUREMENTS.
?6
TYPE THE NUMBER OF FEET IN THE SECOND SET OF MEASUREMENTS.
?0
TYPE THE NUMBER OF INCHES IN THE SECOND SET OF MEASUREMENTS.
?17
HOW MANY INCHES DID YOU GET IN YOUR FINAL ANSWER?
?2
CORRECT
HOW MANY FEET DID YOU GET IN YOUR FINAL ANSWER?
?0
O.K.
HOW MANY YARDS DO YOU HAVE IN YOUR ANSWER?
?15
CONGRATULATION!! YOU DIDN'T EVEN MAKE ONE MISTAKE!
```

Ghost (#0210)

This program can be useful for classes which are learning their conversion factors of measurements because it challenges the student not to spellout the word GHOST. To get through the game without losing, he must get better than ten out of fifteen questions correct. A file is utilized as well as pass and repass statements. To compare the student's answer with the correct answer in the file, an ICO statement is used. Thus, by playing a simple game, the student is both having fun learning and reviewing his conversion factors of various measurements.

Chris Mellott, 11th Grade

```
>RUN

THIS PROGRAM IS DESIGNED TO TEST THE STUDENT ON THE
CONVERSION FACTORS OF VARIOUS MEASUREMENTS SUCH AS
LENGTH, WEIGHT, AND VOLUME. THIS TEST WILL BE IN
THE FORM OF A GAME, WHICH IS CALLED GHOST.
THE STUDENT WILL GET 15 CHANCES AND FOR EACH MIS-
TAKE HE WILL GET ONE OF THE LETTERS IN THE WORD
GHOST.
IF THE STUDENT GETS THROUGH THE TEST WITHOUT SPELLING
OUT THE WORD GHOST HE WINS! HOWEVER, AS SOON AS THE
STUDENT SPELLS OUT THE WORD GHOST HE LOSES AND THE
TEST IS OVER.
HELLO, WHAT IS YOUR NAME.
?CHRIS MELLOTT
NICE TO MEET YOU,CHRIS MELLOTT.
HOW MANY FEET IN A MILE?
?5280

HOW MANY SQUARE INCHES IN A SQUARE FOOT?
?134
G

HOW MANY CUBIC FEET IN A CUBIC YARD?
?27

HOW MANY QUARTS IN TWO PECKS?
?15
GH

HOW MANY FLUID OUNCES IN TWO PINTS?
?30
GHO

HOW MANY INCHES IN HALF A FOOT?
?6

HOW MANY OUNCES IN A POUND?
?12
GHOS

HOW MANY FEET IN A ROD?
?17
GHOST! YOU JUST LOST, SO GOOD-BYE.
```

ADVANCED MATH

Exponents (#0211)

This program is designed to check a student's knowledge of exponents. It employs no logs but instead simulates the same type of thinking that a student, not knowing logs, would employ. It is designed to only compute problems which do not involve decimals and irrational numbers. These problems can easily be done in one's head, since the program is based on whole number powers of the numerator and denominator of each base, exponent, and number. There are two parts to the program. The first part asks the student to supply a number and an exponent, it then computes the base. The second part computes the number given the base and the exponent.

Robert Hanna, 12th Grade

part 1

```
>RUN
TYPE YOUR NAME
?BOB
TYPE THE NUMBER OF PROBLEMS YOU WISH TO DO
?2
TYPE THE NUMERATOR OF THE NUMBER
?64
TYPE THE DENOMINATOR OF THE NUMBER
?1
TYPE THE NUMERATOR OF THE EXPONENT
?2
TYPE THE DENOMINATOR OF THE EXPONENT
?1
THE BASE IS  8/ 1
NOW START TO DO YOUR NEXT PROBLEM
TYPE THE NUMERATOR OF THE NUMBER
?64
TYPE THE DENOMINATOR OF THE NUMBER
?1
TYPE THE NUMERATOR OF THE EXPONENT
?4
TYPE THE DENOMINATOR OF THE EXPONENT
?1
THE PROBLEM CANNOT BE SOLVED IN THIS MANNER, IT MUST BE
SOLVED BY THE USE OF THE LOGARITHMIC TABLES IN THE BACK OF
YOUR TRIGONOMETRY BOOK
FOLLOW LOGOUT PROCEDURES AND GOODBYE BOB
```

part 2

```
>RUN
TYPE YOUR NAME
?BOB
TYPE THE NUMBER OF PROBLEMS YOU WISH TO DO
?2
TYPE THE NUMERATOR OF THE BASE
?0
TYPE THE DENOMINATOR OF THE BASE
```

```
?2
TYPE THE NUMERATOR OF THE EXPONENT
?1
TYPE THE DENOMINATOR OF THE EXPONENT
?1
THE NUMBER IS ZERO BECAUSE THE NUMERATOR OF
THE BASE IS ZERO
NOW START TO DO YOUR NEXT PROBLEM
TYPE THE NUMERATOR OF THE BASE
?-1
TYPE THE DENOMINATOR OF THE BASE
?1
TYPE THE NUMERATOR OF THE EXPONENT
?2
TYPE THE DENOMINATOR OF THE EXPONENT
?1
THE NUMBER IS ONE BECAUSE NEGATIVE ONE TO AN EVEN POWER IS
POSITIVE ONE
FOLLOW LOGOUT PROCEDURES AND GOODBYE BOB
```

Sum of Integers (#0212)

This program is designed to help algebra students who have trouble solving sum of integers problems. It has the flexiblity to check the sum of any number of integers and even tells the student if the problem is impossible. The program is a step by step tutorial that guides the student through to completion. A file of comments, that are randomly called, helps encourage the student. The program utilizes string functions to give the computer a stutter.

David Rorison, 12th Grade

```
>RUN
WHAT IS YOUR FULL NAME?
?DAVID RORISON
HOW MANY CONSECUTIVE INTEGERS ARE THERE IN THIS PROBLEM?
?1
HA,HA. YOU CAN'T FOOL ME. THERE'S GOT TO BE
MORE THAN ONE INTEGER INPUT YOUR NUMBER AGAIN.
?5
WELL,DAVID, HOW ABOUT INPUTTING THE SUM OF THE 5
CONSECUTIVE INTEGERS FOR THIS PROBLEM.
?66
BETTER INPUT YOUR SUM FOR YOUR 5 CONSECUTIVE
INTEGERS AGAIN. THERE ARE NOT 5 CONSECUTIVE
INTEGERS THAT ADD UP TO 66.
?65
```

```
OK IT'S POSSIBLE THIS TIME. THERE ARE 5
CONSECUTIVE INTEGERS THAT ADD UP TO 65.
WHAT IS THE VALUE YOU HAVE FOR THE FIRST CONSECUTIVE INTEGER?
?11
GOOD, YOU GOT IT RIGHT.
WHAT DO YOU THINK IS THE VALUE FOR THE SECOND CONSECUTIVE
INTEGER?
?12
GOOD SHOW DDDDAVID RORISON KEEP IT UP.
DAVID, YOU'LL HAVE TO EXCUSE MY TERRIBLE STUTTERING.
IT'S UNFORTUNATE, BUT I HAVE A SPEECH IMPEDIMENT.
I OCCASIONALLY HAVE TROUBLE PRONOUNCING D'S.
WHAT IS THE VALUE YOU'VE GOTTEN FOR THE THIRD
CONSECUTIVE INTEGER?
?13
WAY TO GO. YOUR INTELLECT IS ASTOUNDING.
WHAT IS YOUR VALUE FOR YOUR 4TH CONSECUTIVE INTEGER.
?14
YOU'RE A REAL GENIUS. PROCEED AT A RAPID PACE.
WHAT IS YOUR VALUE FOR THE FINAL INTEGER.
?15
YOU DID IT. HOW ABOUT DOING ANOTHER PROBLEM?
TYPE YES IF YOU WANT TO TRY AGAIN AND NO IF YOU DON'T.
?YES
HOW MANY CONSECUTIVE INTEGERS ARE THERE IN THIS PROBLEM?
?5
WELL,DAVID, HOW ABOUT INPUTTING THE SUM OF THE 5
CONSECUTIVE INTEGERS FOR THIS PROBLEM.
?65
WHAT IS THE VALUE YOU HAVE FOR THE FIRST CONSECUTIVE INTEGER?
?11
GOOD, YOU GOT IT RIGHT.
WHAT DO YOU THINK IS THE VALUE FOR THE SECOND CONSECUTIVE
INTEGER?
?12
GOOD SHOW DDDDAVID RORISON KEEP IT UP.
DAVID, YOU'LL HAVE TO EXCUSE MY TERRIBLE STUTTERING.
IT'S UNFORTUNATE, BUT I HAVE A SPEECH IMPEDIMENT.
I OCCASIONALLY HAVE TROUBLE PRONOUNCING D'S.
WHAT IS THE VALUE YOU'VE GOTTEN FOR THE THIRD
CONSECUTIVE INTEGER?
?13
WAY TO GO. YOUR INTELLECT IS ASTOUNDING.
WHAT IS YOUR VALUE FOR YOUR 4TH CONSECUTIVE INTEGER.
?14
JUST WHAT I WAS LOOKING FOR. KEEP GOING.
WHAT IS YOUR VALUE FOR THE FINAL INTEGER.
?14
THAT'S NOT CORRECT.
HOW COULD THE 5TH INTEGER= 14 IF THE 4TH
INTEGER= 14.  BETTER TRY AGAIN.
WHAT IS YOUR FINAL INTEGER?
?15
YEA,DAVID YOU'RE MY HERO. YOU GOT ALL 5
INTEGERS CORRECT.
NOW, THAT YOU'VE ACCOMPLISHED THIS FEAT, HOW ABOUT
TRYING ANOTHER PROBLEM?
TYPE YES IF YOU WANT TO, NO IF YOU DON'T.
?NO
WELL, I'M SORRY YOU DON'T WANT TO DO ANOTHER PROBLEM.
MAYBE YOU WILL IN THE FUTURE.
BYE-BYE
```

Work (#0213)

This program was designed for a second year algebra class to individually practice solving work problems. The program creates a file to keep track of the students who have been assigned problems, the pertinent information about each student's problem, and each student's progress. The program allows each student to return to the computer after he has solved the problem. If the student's answer is incorrect, it helps the student find his error.

Don Salvin, 12th Grade

```
>RUN
THIS IS GEORGE HINKLEDINK,MISS KAVANAGH'S FAVORITE COMPUTER.
WHAT IS YOUR FULL NAME?
?DASH GORDON
        CHARLIE IS THE NAME OF MY FAVORITE ANT.THE NAME
OF MY FAVORITE BEE IS JIMMY.JIMMY AND CHARLIE BOTH CARRY
SUGAR FOR A LIVING.
        WHEN JIMMY IS WORKING HIS HARDEST,(AS HE
A L W A Y S DOES),HE CAN CARRY ONE CUP OF SUGAR FOR A DIS-
TANCE OF TEN FEET IN 16 DAYS. BUT CHARLIE IS LAZIER THAN
JIMMY SO THAT IT TAKES CHARLIE 19 DAYS TO CARRY THE
SAME AMOUNT OF SUGAR THE SAME DISTANCE.IF JIMMY AND CHARLIE
WORK TOGETHER,HOW MANY DAYS DO YOU THINK IT WILL TAKE THEM
TO DO THE JOB?

GOOD-BYE FOR NOW.WHEN YOU THINK THAT YOU HAVE THE CORRECT
ANSWER,COME BACK AND RERUN THIS PROGRAM.
FINISHED
DONE!! DONE!!
>RUN
THIS IS GEORGE HINKLEDINK,MISS KAVANAGH'S FAVORITE COMPUTER.
WHAT IS YOUR FULL NAME?
?K
THE FOLLOWING STUDENT(S) HAVE BEEN ASSIGNED A PROBLEM.
STUDENT                   TIM FOR CH.       TIM FOR JIM    COR. ANS.

GEEORGE POPY                   5                 2             1.43
DASH GORDON                   19                16             8.69
NOBODY HAS COMPLETED THIS PROGRAM YET.
FINISHED
DONE!! DONE!!
```

```
>RUN
THIS IS GEORGE HINKLEDINK,MISS KAVANAGH'S FAVORITE COMPUTER.
WHAT IS YOUR FULL NAME?
?DASH GORDON
HOW MANY DAYS DO YOU THINK IT WILL TAKE FOR CHARLIE AND JIMMY
WORKING TOGETHER TO CARRY THE SUGAR?
?1.9
AH HA! DASH GORDON
YOU GOOFED.LET'S FIND OUT WHERE.
IF JIMMY CAN CARRY ALL OF THE SUGAR IN  16 DAYS,THEN
WHAT IS THE NUMERATOR OF THE FRACTION OF THE WORK THAT HE
CAN DO IN ONE DAY?
?1
RIGHT
WHAT IS THE DENOMINATOR OF THE FRACTION OF THE WORK THAT JIMMY
CAN DO IN ONE DAY?
?16
GOOD
IF CHARLIE CAN DO ALL OF THE WORK ALONE IN 19 DAYS THEN
WHAT IS THE NUMERATOR OF THE FRACTION OF THE SUGAR THAT HE
CAN CARRY IN ONE DAY?
?1
CORRECT
WHAT IS THE DENOMINATOR OF THE FRACTION OF THE WORK THAT
CHARLIE CAN DO IN ONE DAY?
?20
THE DENOMINATOR OF THE FRACTION OF THE WORK THAT CHARLIE CAN
DO IN ONE DAY WILL ALWAYS EQUAL THE NUMBER OF DAYS THAT
CHARLIE TAKES TO DO THE JOB ALONE OR 19
WELL, DASH, THERE IS YOUR FIRST MISTAKE!
NOW DIVIDE OUT THE FIRST FRACTION THAT YOU TOLD ME,(OR THAT I
TOLD YOU!).WHAT DO YOU GET?
?0.0625
FINE
DIVIDE OUT THE SECOND FRACTION.WHAT DO YOU GET?
?0.9
ACH MEIN GOTT DASH! YOUR SECOND MISTAKE!!
WHAT IS THE MATTER? IS YOUR MENTAL SLIDE RULE STUCK?
1/ 19= 0.052631579,BUT I WOULD HAVE ACCEPTED ANYTHING
BETWEEN  2.631578947E-03 AND  0.102631579
NOW ADD UP THE TWO DECIMALS THAT HAVE BEEN DIVIDED OUT.WHAT
DO YOU GET?
?0.09
EXACTLY RIGHT!
THIS DECIMAL REPRESENTS THE AMOUNT OF THE JOB THAT JIMMY
AND CHARLIE CAN DO IN ONE DAY.THE RECIPROCAL OF THIS NUM-
BER IS THE NUMBER OF DAYS THAT CHARLIE AND JIMMY WILL TAKE TO
DO THE JOB TOGETHER.WHAT IS THE RECIPROCAL OF ( 0.115131579) ?
?9.5
WELL, DASH, THERE IS YOUR THIRD MISTAKE!!!
THE RECIPROCAL OF A NUMBER IS SIMPLY THAT NUMBER DIVIDED
INTO 1.WHAT IS  0.115131579 INTO 1 ?
?8.69
```

```
EXACTLY RIGHT!
YOU'RE FINISHEDDASH GORDON,YOU HAVE MADE  3 MISTAKES IN ALL.
FINISHED
DONE!! DONE!!

>RUN
THIS IS GEORGE HINKLEDINK,MISS KAVANAGH'S FAVORITE COMPUTER.
WHAT IS YOUR FULL NAME?
?K
THE FOLLOWING STUDENT(S) HAVE BEEN ASSIGNED A PROBLEM.

STUDENT                      TIM FOR CH.        TIM FOR JIM    COR. ANS.

GEEORGE POPY                      5                 2            1.43
DASH GORDON                      19                16            8.69

THE FOLLOWING STUDENT(S) HAVE COMPLETED THIS PROGRAM.

STUDENT                 NUMBER OF           ERROR
                      ERRORS MADE         LOCATIONS


DASH GORDON                 3              (4),(6),(8)
FINISHED
DONE!! DONE!!
```

<u>Scientific Notation</u> (#0214)

This program allows students to either convert numbers from their regular form to scientific notation or into regular form from scientific notation. If the student makes a mistake, it helps the student discover his error.

Steve Swanson, 12th Grade

```
>RUN
WHAT IS YOUR FULL NAME?
?STEVE SWANSON
HOW MANY PROBLEMS DO YOU WISH TO DO, STEVE?
?4
WHICH WILL YOU BE INPUTTING FIRST: A NUMBER IN
'SCIENTIFIC NOTATION' OR A NUMBER IN 'REGULAR
FORM'?
?SCIENTIFIC NOTATION
INPUT THE GIVEN NUMBER IN SCIENTIFIC NOTATION.
?4.35X10↑4
TYPE IN YOUR ANSWER IN REGULAR FORM.
?43500
GOOD WORK, STEVE, THAT'S EXACTLY RIGHT.
TRY ANOTHER PROBLEM.
WHICH WILL YOU BE INPUTTING FIRST: A NUMBER IN
'SCIENTIFIC NOTATION' OR A NUMBER IN 'REGULAR
FORM'?
?SCIENTIFIC NOT
INPUT THE GIVEN NUMBER IN SCIENTIFIC NOTATION.
?4.35X10↑4
TYPE IN YOUR ANSWER IN REGULAR FORM.
?4350
SORRY, STEVE, BUT YOU MADE A MISTAKE.
WHEN CONVERTING FROM SCIENTIFIC NOTATION
TO REGULAR FORM, YOU MOVE THE DECIMAL POINT
OF THE NUMBER PRECEDING THE MULTIPLICATION SIGN THE
NUMBER OF PLACES INDICATED BY THE EXPONENT
OF THE 10 (TO THE LEFT IF IT IS NEGATIVE, OR
TO THE RIGHT IF IT IS POSITIVE) ADDING ZEROS AS NEEDED.

IN WHICH DIRECTION WILL YOU MOVE THE DECIMAL POINT?
?RIGHT
SO WHAT IS 4.35X10↑4 WRITTEN IN REGULAR FORM?
?43500
GOOD WORK, STEVE, THAT'S EXACTLY RIGHT.
TRY ANOTHER PROBLEM.
WHICH WILL YOU BE INPUTTING FIRST: A NUMBER IN
'SCIENTIFIC NOTATION' OR A NUMBER IN 'REGULAR
```

```
FORM'?
?SCIENTIFIC N.
INPUT THE GIVEN NUMBER IN SCIENTIFIC NOTATION.
?-4.35X1Ø↑Ø
TYPE IN YOUR ANSWER IN REGULAR FORM.
?-43.5

IN WHICH DIRECTION WILL YOU MOVE THE DECIMAL POINT?
?RIGHT
THAT WAS REALLY AN UNFAIR QUESTION.
SINCE THE EXPONENT WAS ZERO, YOU WILL NOT
MOVE THE DECIMAL POINT AT ALL.

SO WHAT IS -4.35X1Ø↑Ø WRITTEN IN REGULAR FORM?
?-435.
THE CORRECT ANSWER IS-4.35.
SORRY, STEVE, BUT YOU MADE ANOTHER ERROR.
LOOK AT THE ANSWER CAREFULLY TO SEE WHERE YOU WENT WRONG.
TRY ANOTHER PROBLEM.
WHICH WILL YOU BE INPUTTING FIRST: A NUMBER IN
'SCIENTIFIC NOTATION' OR A NUMBER IN 'REGULAR
FORM'?
?SCIENTIFIC
INPUT THE GIVEN NUMBER IN SCIENTIFIC NOTATION.
?4.35X1Ø↑-1
TYPE IN YOUR ANSWER IN REGULAR FORM.
?4.35

IN WHICH DIRECTION WILL YOU MOVE THE DECIMAL POINT?
?RIGHT
YOU WILL MOVE TO THE LEFT.
SO WHAT IS 4.35X1Ø↑-1 WRITTEN IN REGULAR FORM?
?.435
GOOD WORK, STEVE, THAT'S EXACTLY RIGHT.
THAT'S ALL FOR NOW. BYE.
-------------------------------------------
```

FRENCH AND U.S. HISTORY

Hangman (#0215)

The versatility of this program is one of its most valuable features. Currently a German class uses it, but with suitable files it can be modified to any subject. Presently, a revision is being written to allow a refresher arithmetic class to use it. The program is a good tutorial because it reveals clues to the student. Those who execute the program enjoy watching it fill in the string of characters already identified and drawing the partially filled gallow. The usual rules for the game are followed. The program requires no programming knowledge to execute.

Mark Pilant, 10th Grade

```
>RUN
TYPE YOUR FIRST NAME
?MLL
THIS IS A PROGRAM TO PLAY HANGMAN.
IF YOU THINK YOU KNOW THE WORD TYPE 'I KNOW' WHEN I ASK FOR THE LETTER.
YOU CAN CHOOSE WORDS WITH 5 TO 7 LETTERS INCLUSIVE.
TYPE THE NUMBER OF LETTERS YOU WOULD LIKE TO HAVE IN YOUR  WORD.
?7
ALL RIGHT, I'M READY. I HAVE THE PERFECT WORD. YOU'LL NEVER GUESS IT.
OH! BY THE WAY, I GUESS I'D BETTER TELL YOU NOW, YOU ONLY GET 10 GUESSES
OR S E V E N MISTAKES.
WHAT LETTER DO YOU CHOOSE.
?A
INCORRECT
THAT'S YOUR FIRST MISTAKE, MLL
YOU HAVE  9 TRIES LEFT.
CLUE: A NOUN


         -------
         I     I
         O     I
               I
               I
               I
               I
               I
    --------------


WHAT LETTER DO YOU CHOOSE.
?B
THAT'S NOT RIGHT
THAT'S YOUR SECOND MISTAKE, MLL
YOU HAVE  8 TRIES LEFT.
CLUE:  A DIE WORD
```

```
         -------
         1     1
         0     1
         !     1
               1
               1
               1
               1
   ---------------
```

```
WHAT LETTER DO YOU CHOOSE.
?C
INCORRECT
THAT'S YOUR THIRD MISTAKE, MLL
YOU HAVE  7 TRIES LEFT.
CLUE: MINUS THE SUFFIX IT BECOMES A MASCULINE NOUN
```

```
         -------
         1     1
         0     1
        \!     1
               1
               1
               1
               1
   ---------------
```

```
WHAT LETTER DO YOU CHOOSE.
?D
WHY DID YOU GIVE ME THE WRONG LETTER?
THAT'S YOUR FOURTH MISTAKE, MLL
YOU HAVE  6 TRIES LEFT.
CLUE: PLURAL FORMED BY ADDING NEN
```

```
         -------
         1     1
         0     1
        \!/    1
               1
               1
               1
               1
   ---------------
```

```
WHAT LETTER DO YOU CHOOSE.
?E
GREAT
THAT'S YOUR FIFTH TRY, MLL
YOUR WORD LOOKS LIKE THIS

E - - E - - -
```

```
WHAT LETTER DO YOU CHOOSE.
?LEHRERIN
YOU CAN'T CHOOSE MORE THAN ONE LETTER AT A TIME, MLL
WHAT LETTER DO YOU CHOOSE.
?F
INCORRECT
THAT'S YOUR FIFTH MISTAKE, MLL
YOU HAVE  4 TRIES LEFT.
CLUE: FIRST LETTER IS THE SAME AS THE WORD FOR FEVER IN GERMAN


          -------
          1     1
          0     1
         \!/    1
          !     1
                1
                1
                1
   ---------------


WHAT LETTER DO YOU CHOOSE.
?I KNOW
WHAT DO YOU THINK THE WORD IS, MLL
?ENKEL
HA!HA!HA! I TOLD YOU THAT YOU WOULDN'T GET IT.
I SHALL BE MERCIFUL AND LET YOU CONTINUE.
WHAT LETTER DO YOU CHOOSE.
?I
FIGHT
THAT'S YOUR SEVENTH TRY, MLL
YOUR WORD LOOKS LIKE THIS

E  -  -  E  -  I  -

WHAT LETTER DO YOU CHOOSE.
?Z
INCORRECT
THAT'S YOUR SIXTH MISTAKE, MLL
YOU HAVE  2 TRIES LEFT.
CLUE: A MEMBER OF THE FAMILY


          -------
          1     1
          0     1
         \!/    1
          !     1
         /      1
                1
                1
   ---------------


WHAT LETTER DO YOU CHOOSE.
?I KNOW
WHAT DO YOU THINK THE WORD IS, MLL
?ENKELIN
DARN IT. YOU GOT IT. I GUESS YOU'RE SMARTER THAN I FIGURED
YOU TO BE, MLL
GOOD-BYE, MLL
```

French Literature and History (#0216)

This program has been designed to test students on French literature and history. The program uses four files. The first two store the questions and the correct answers respectively. The third contains phrases of encouragement in French that are randomly written at appropriate points of the tutorial. The fourth file stores eight extra questions and answers for those students who wish extra credit. This program, because it is in French, determines the sex of the student so the appropriate gender and adjective endings are used. In addition, the program keeps a tally of the problems missed, and at its conclusion gives the student the number of questions missed and then a statement regarding his (or her) performance.

Dale Klopfer, 12th Grade

```
>RUN
VOICI UNE PROGRAMME EN FRANCAIS.  MALHEUREUSEMENT, CETTE MACHINE
N'EST PAS EN FRANCAIS.  DONC IL N'Y AURA PAS D' ACCENTS OU
D' AUTRES CHOSES FRANCAIS.
BONNE CHANCE!!
COMMENT VOUS APPELEZ-VOUS?
?MARIE DUPONT

QUELLE PHRASE CELEBRE DANS <<LA FARCE DE MAITRE PATHELIN>>VEUT DIRE
<<RENTRONS A NOTRE  SUJET>>?
?BE
EXACTEMENT!!
QUEL ROI ETAIT <<SAINT LOUIS>>?
?LU↑OUIS IX
EXACTEMENT CORRECTE!!

QUEL POET DU MOYEN AGE A EU LE MAL DU PAYS?
 (SUGGESTIONS:CHARLES;LOUISIANA)
?DU BELLAY
NON. TU AS TORT, MARIE.  VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?NO
ENCORE UNE FOIS, S'IL TE PLAIT.
?NON
CA M'EST BIEN EGAL, MARIE
QU'EST-CE QUI ETAIT LA SYMBOLE DE FRANCOIS I?
?UNE SALAMANDRE
CE N'EST PAS MAL!
QUI ETAIT ALCOFRIBAS NASIER?
?RABELAIS
TU AS RAISON COMME D'HABITUDE!
```

QUI ETAIT UN GRAND ERUDIT ET HUMANISTE HOLLANDAIS QUI A ECRIT
<<L'ELOGE DE LA FOLIE>>? (C'ETAIT PENDANT LA RENAISSANCE)
?REMBRANDT
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?OUI
LA REPONSE EST ERASME
QU'EST CE QUE C'ETAIT LA DEVISE OFFICIELLE DE L'ABBAYE DE THELEME?
?FAIT CE QU'UN VEUT
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?OUI
LA REPONSE EST FAIS CE QUE TU VOUDRAS
QUI A ECRIT LA <<DISCOURS SUR LA METHODE>>?
?DESCARTS'ES
TRES TRES BIEN!
QUI ETAIT JEAN-BAPTISTE POQUELIN?
?VOLTAIRE
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?OUI
LA REPONSE EST MOLIERE

QUI A DIT QUE LA POSITION DE L'HOMME DANS L'UNIVERS EST ENTRE
L'INFINIMENT GRAND ET L'INFINIMENT PETIT?
?BLAISE PASCAL
VRAI
QUI A ECRIT <<LE CID>>?
?CORNEILLE
EXACTEMENT CORRECTE!!
QUI ETAIT LE CHEF DU REGNE DE LA TERREUR?
?MARAT
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?NON
CA M'EST BIEN EGAL, MARIE

QU'EST-CE QUE C'EST QU'UN AUTRE NOM POUR L'EGLISE DE SAINTE-GENEVIEVE
DE SOUFFLOT?
?NOTRE DAME DE PARIS
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?OUI
LA REPONSE EST PANTHEON
QUI A DIT <<APRES MOI; LE DELUGE>>?
?LOUIS XV
TU AS RAISON COMME D'HABITUDE!
QUI A PRONONCE CETTE PHRASE:<<PARIS VAUT BIEN UNE MASSE>>?
?FRANCOIS I
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?NON
CA M'EST BIEN EGAL, MARIE
DANS <<LES ROMAN DE RENARD>>; QUI ETAIT LE TROMPEUR TROMPEE?
?CHANTECLIN
NON. TU AS TORT, MARIE. VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?M'NON
CA M'EST BIEN EGAL, MARIE

```
QUI ETAIT LE COMPOSITEUR DES OPERAS ET DES BALLETS POUR LA COUR ROYALE
PENDANT LA RENAISSANCE?
?RAMEAU
EXCELLENT!

QU'EST-CE QUE C'ETAIT LA PERIODE ENTRE LA MORT DE LOUIS XIV ET LE
COMMENCEMENT DU REGNE PERSONNEL DU LOUIS XV?
?LE 'S CENT JOURS
NON. TU AS TORT, MARIE.  VEUX TU SAVOIR LA REPONSE CORRECTE?
TAPEZ A LA MACHINE OUI OU NON
?OUI
LA REPONSE EST REGENCE

QUELS STYLES SONT SUGGERES PAR LES STUCS; LE BOIS DORE; DES CUPIDONS;
ETC?
?BAROQUE ET ROCCOCO
CORRECTE!
QUI ETAIT LA FEMME DE LOUIS XVI?
?MARIE ANTOINETTE
EXACTEMENT!!
VEUX-TU DU CREDIT SUPPLEMENTAIRE? (TAPEZ A LA MACHINE OUI OU NON)
?NON
ALORS, VOYONS TA NOTE,MARIE
MOINS 9 IL FAUT QUE TU CULTIVES TON JARDIN,
MARIE DUPONT
ADIEU, MARIE, MA BELLE AMIE!!
```

<u>French Grammar</u> (#0217)

This program was designed to review present subjective tense of regular and irregular French verbs. It uses any array and a simple data file to record the verb forms in the present tense and the correct responses in the subjective. At the beginning of the program, the instructions for running the program are given in French.

Ann Fritts, 12th Grade

```
COMMENT VOUS APPELEZ-VOUS?
?ANN FRITTS
BONJOUR, ANN. VOICI UN EXAMEN SUR LE SUBJONTIF DES VERBES
FRANCAIS. VOICI LE PREMIER VERBE. NE FAUT PAS ECRIRE LE PRONOM
OU LES ACCENTS,SIMPLEMENT LA FORME DU VERBE. BON CHANCE, ANN.
IL JOUE QUELLE EST LA REPONSE?
?JOUE
C'EST CORRECT!
NOUS PORTONS QUELLE EST LA REPONSE?
?PORTIONS
C'EST CORRECT!
JE REPONDS QUELLE EST LA REPONSE?
?REPONDE
C'EST CORRECT!
ILS ATTENDENT QUELLE EST LA REPONSE?
?ATTENDENT
C'EST CORRECT!
TU FINIS QUELLE EST LA REPONSE?
?FINNISSES
NON, LA REPONSE EST FINISSES.
VOUS CHOISISSEZ QUELLE EST LA REPONSE?
?CHOISSIEZ
```

NON, LA REPONSE EST CHOISISSIEZ.
J'APPELLE QUELLE EST LA REPONSE?
?APPELLE
C'EST CORRECT!
IL VA QUELLE EST LA REPONSE?
?AILLE
C'EST CORRECT!
TU APERCOIS QUELLE EST LA REPONSE?
?APERCOIVES
C'EST CORRECT!
J'ASSIEDS QUELLE EST LA REPONSE?
?ASSEYE
C'EST CORRECT!
ILS ONT QUELLE EST LA REPONSE?
?AIENT
C'EST CORRECT!
NOUS CONNAISSONS QUELLE EST LA REPONSE?
?CONNAISSIONS
C'EST CORRECT!
VOUS CROYEZ QUELLE EST LA REPONSE?
?CROYIEZ
C'EST CORRECT!
TU DOIS QUELLE EST LA REPONSE?
?DOIVES
C'EST CORRECT!
IL ECRIT QUELLE EST LA REPONSE?
?ECRITE
NON, LA REPONSE EST ECRIVE.
J'ETEINS QUELLE EST LA REPONSE?
?ETI↑EIGNE
C'EST CORRECT!
VOUS ETES QUELLE EST LA REPONSE?
?SOYEZ
C'EST CORRECT!
NOUS FAISONS QUELLE EST LA REPONSE?
?FAISIONA↑S
NON, LA REPONSE EST FASSIONS.
ILS METTENT QUELLE EST LA REPONSE?
?METTENT
C'EST CORRECT!
JE PARS QUELLE EST LA REPONSE?
?PARTE
C'EST CORRECT!
VOUS POUVEZ QUELLE EST LA REPONSE?
?PO↑UISSIEZ
C'EST CORRECT!
TU PRENDS QUELLE EST LA REPONSE?
?PRENNES
C'EST CORRECT!
NOUS SAVONS QUELLE EST LA REPONSE?
?SACHIONS
C'EST CORRECT!
IL VAUT QUELLE EST LA REPONSE?
?VAILLE
C'EST CORRECT!
JE VIENS QUELLE EST LA REPONSE?
?VIENNE
C'EST CORRECT!
ILS VOIENT QUELLE EST LA REPONSE?
?VOIENT
C'EST CORRECT!
TU VEUX QUELLE EST LA REPONSE?
?VEUILLES
C'EST CORRECT!
AU REVOIR, ANN.

<u>Presidents</u> (#0218)

This program allows a teacher to create a file of questions and answers in any area where quick identification is sought. A series of questions and their answers about U.S. Presidents has already been typed in for this demonstration.

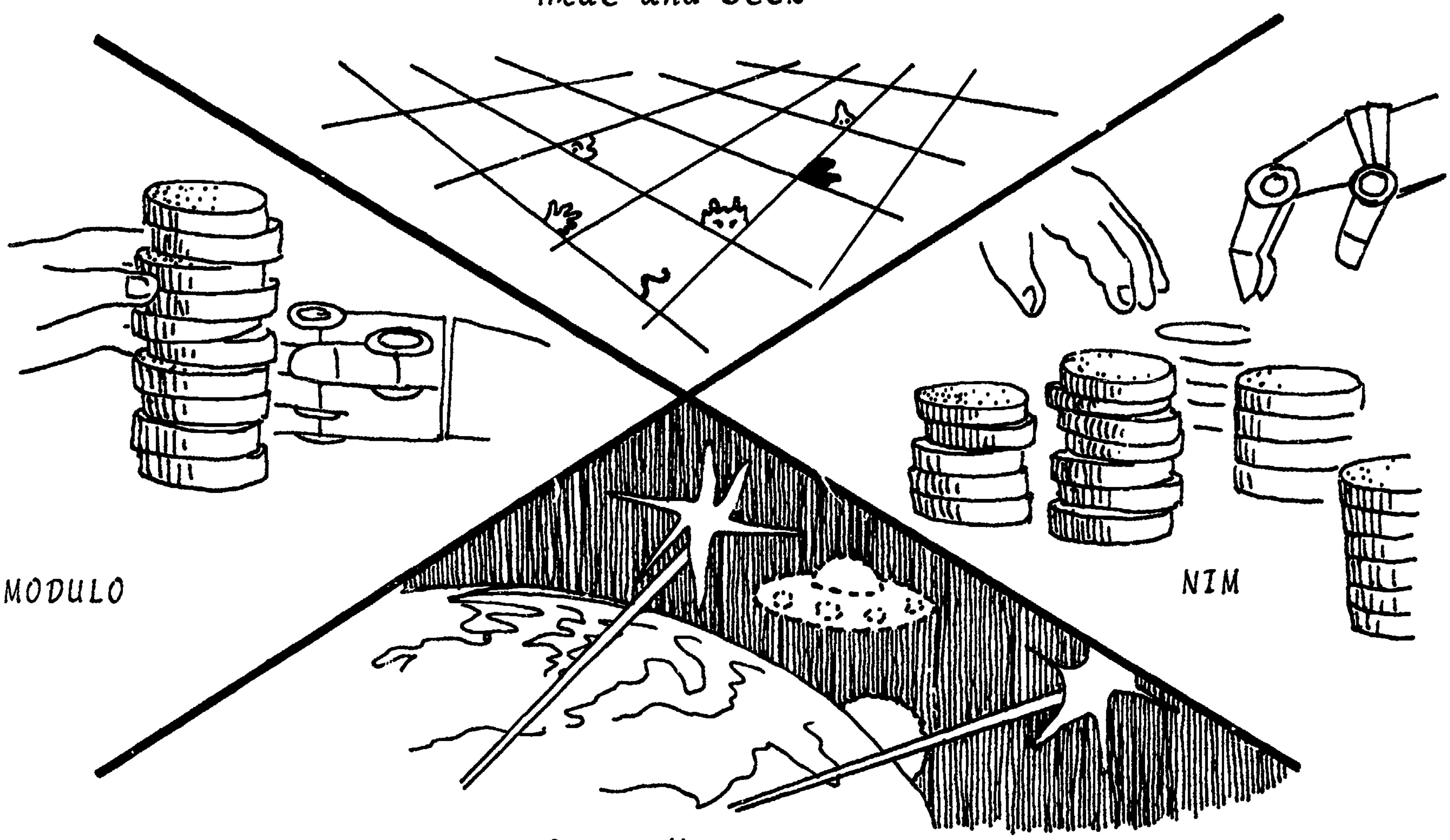
Ed Satler, 11th Grade

```
                         PRESIDENTS

HELLO,I WOULD LIKE TO KNOW YOUR NAME(FIRST AND LAST).
WHAT IS IT?ED

I DON'T LIKE TO REPEAT MYSELF,BUT I GUESS I MUST.
WHAT IS YOUR FULL NAME?ED SATLER


THIS PROGRAM WILL TEST YOUR KNOWLEDGE OF THE PRESIDENTS
OF THE UNITED STATES.YOU WILL BE ASKED SIX QUESTIONS
ABOUT THE PRESIDENTS AND BE SCORED ACCORDINGLY.
ARE YOU READY,ED?AE YOU

?--PLEASE RESPOND AGAIN
?ARE YOU

??--PLEASE RESPOND AGAIN
?OF COURSE


OKAY,LET US BEGIN.WHEN ANSWERING A QUESTION,
MAKE SURE THAT YOUR SPELLING IS CORRECT.

FIRST QUESTION,ED.
WHO WAS THE FIRST PRESIDENT OF THE UNITED STATES?
?HIS NAME WAS GEORGE WASHINGTON
RIGHT
NEXT QUESTION.

WHICH PRESIDENT WAS ASSASSINATED BY JOHN WILKES BOOTHE?
?ABRAHAM LINCOLN
GOOD
NEXT QUESTION.

WHO WAS PRESIDENT DURING THE START OF WORLD WAR 1?
?WOODROW
ED,YOU MUST INCLUDE THE CORRECT LAST NAME IN YOUR ANSWER.
??--PLEASE RESPOND AGAIN
?WOODROW WILSON
THAT'S RIGHT
NEXT QUESTION.
```

```
WHICH PRESIDENT WON THE DISPUTED ELECTION OF 1876?
?HARRISON
TOO BAD,ED,YOUR ANSWER IS INCORRECT.NEXT QUESTION.

WHO WAS ELECTED PRESIDENT IN 1968
?NIXON
CORRECT
NEXT QUESTION.

WHO WAS PRESIDENT DURING THE BEGINNING OF THE GREAT DEPRESSION?
?HUMPHREY
TOO BAD,ED,YOUR ANSWER IS INCORRECT.QUESTIONING IS OVER.

OUT OF THE SIX QUESTIONS,YOUR TOTAL NUMBER RIGHT IS 4
OR 66% CORRECT,WHICH IS FAIR.

ED SATLER,DO YOU WANT TO RUN THIS PROGRAM AGAIN?WHAT

??--PLEASE RESPOND AGAIN
?NOPE
```

COMPUTER GAMES

in BASIC



The four games in this booklet were written and tested on a PDP-10, using release 17 BASIC.

The games are:

PROJECT SOLO / Computer Science Dept. / Univ. of Pittsburgh (15213)
Modules #0201-4 / Jeff Lederer (PS)

(Hide & Seek and MODULO were originated by students of mathematics teacher, Bud Valenti.)

HIDE AND SEEK

The following program is a game called "Hide and Seek". This game encourages students to become familiar with the Cartesian Coordinate system by asking them to find four players hidden at grid points on a ten by ten graph. Allowable grid points have the form (J,K) where J and K are integers between 0 and 9 inclusive. The student is given ten tries in which to find the player's position. After each try, he is told how close his guess was to each player. At the end of the game the student can ask to see where the players he didn't find were hidden. If the student wants to play another game the players move to new hidden locations.

After playing the game "in his head", the student should try to play the game with the aid of graph paper and a compass. A good student can find all the players in six or seven moves by "triangulation", using an approach that is similar to some radio navigational systems (e.g. LORAN).

A sample run of the program is given below. This is followed by a flow chart which explains the logic of the program. The numbers on the flow chart correspond to the statement numbers in the program listing. This listing is given on page 4. This program was written in BASIC for a PDP-10.

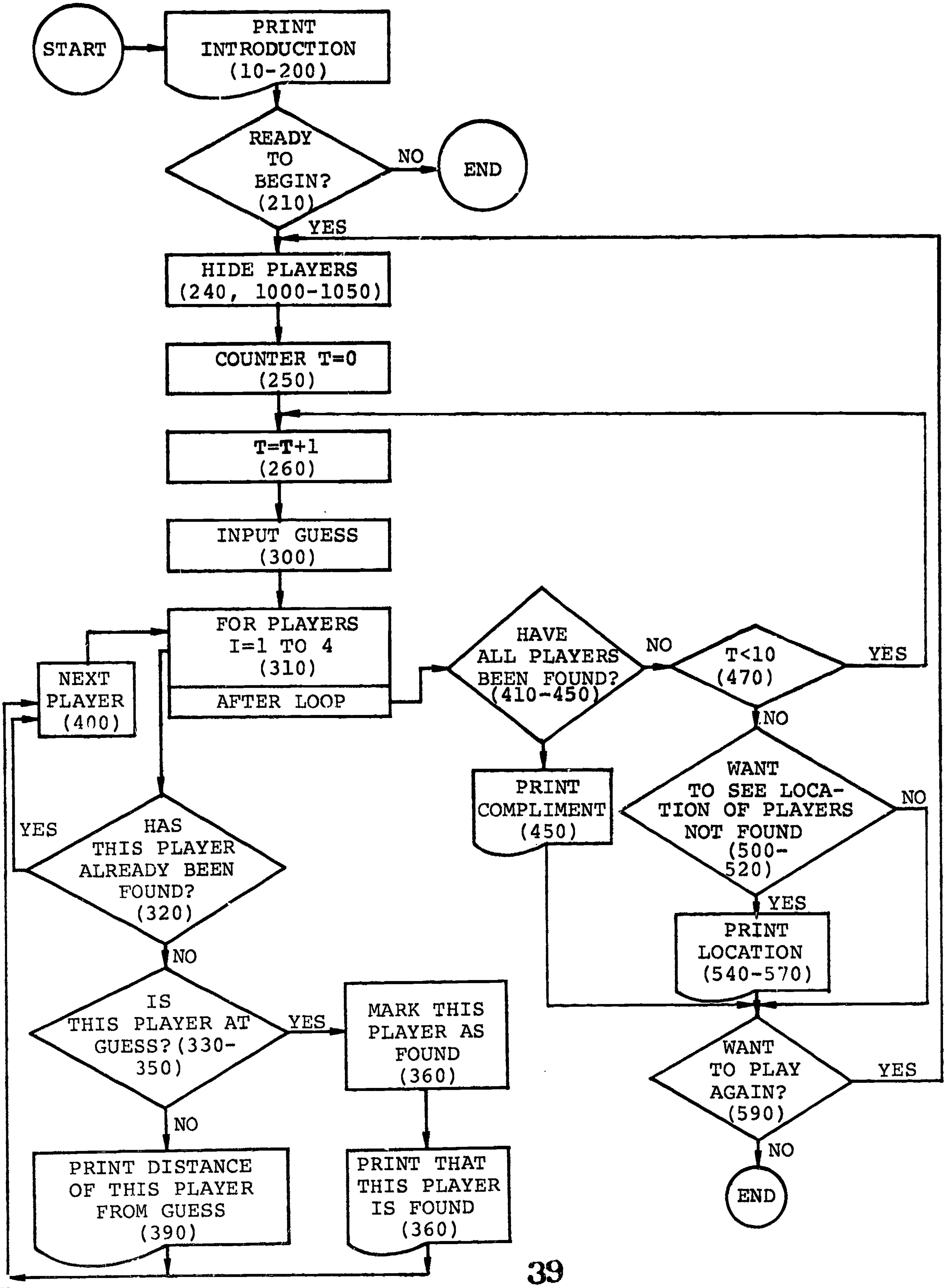
Sample RUN of /HIDE AND SEEK/

```
THIS IS THE GAME ØF HIDE AND SEEK.
THE ØBJECT ØF THE GAME IS TØ FIND THE FØUR PLAYERS
WHØ ARE HIDDEN ØN A 10 BY 10 GRID.

HØMEBASE WILL BE THE PØSITIØN AT (0,0) AND ANY GUESS
YØU MAKE SHØULD CØNTAIN TWØ NUMBERS.  THE FIRST GIVES
THE UNIT DISTANCE RIGHT ØF THE HØMEBASE AND THE SECØND
IS THE UNIT DISTANCE ABØVE HØMEBASE.

YØU WILL HAVE TEN ATTEMPTS TØ LØCATE THESE PLAYERS
AND WILL BE TØLD HØW CLØSE YØUR GUESS IS
TØ EACH PLAYER.

IF AFTER 10 TRIES YØU ARE UNABLE TØ CARRY ØUT THIS TASK
YØU MAY CØNTINUE TØ BE 'IT', BUT THE PLAYERS WILL
BE PERMITTED TØ MØVE TØ NEW LØCATIØNS.


ARE YØU READY TØ BEGIN?
? YES


TURN NUMBER 1 , WHAT IS YØUR GUESS?
? 5,5
YØUR DISTANCE FRØM PLAYER 1 IS 3.1 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 2.2 UNIT(S).
YØUR DISTANCE FRØM PLAYER 3 IS 4.2 UNIT(S).
YØUR DISTANCE FRØM PLAYER 4 IS 5.6 UNIT(S).
```

2

```
DØ YØU WANT TØ PLAY AGAIN?
? YES


TURN NUMBER 1 , WHAT IS YØUR GUESS?
? 5,5
YØUR DISTANCE FRØM PLAYER 1 IS 5.6 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 5.8 UNIT(S).
YØUR DISTANCE FRØM PLAYER 3 IS 4.4 UNIT(S).
YØUR DISTANCE FRØM PLAYER 4 IS 3.1 UNIT(S).


TURN NUMBER 2 , WHAT IS YØUR GUESS?
? 8,0
YØUR DISTANCE FRØM PLAYER 1 IS 11.4 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 11.3 UNIT(S).
YØUR DISTANCE FRØM PLAYER 3 IS 9 UNIT(S).
YØUR DISTANCE FRØM PLAYER 4 IS 8.2 UNIT(S).


TURN NUMBER 3 , WHAT IS YØUR GUESS?
? 8,9
YØUR DISTANCE FRØM PLAYER 1 IS 7 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 8 UNIT(S).
YØUR DISTANCE FRØM PLAYER 3 IS 1 UNIT(S).
YØUR DISTANCE FRØM PLAYER 4 IS 2.2 UNIT(S).


TURN NUMBER 4 , WHAT IS YØUR GUESS?
? 7,9
YØUR DISTANCE FRØM PLAYER 1 IS 6 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 7 UNIT(S).
YØU HAVE FØUND PLAYER 3
YØUR DISTANCE FRØM PLAYER 4 IS 1.4 UNIT(S).


TURN NUMBER 5 , WHAT IS YØUR GUESS?
? 6,8
YØUR DISTANCE FRØM PLAYER 1 IS 5 UNIT(S).
YØUR DISTANCE FRØM PLAYER 2 IS 6 UNIT(S).
YØU HAVE FØUND PLAYER 4


TURN NUMBER 6 , WHAT IS YØUR GUESS?
? 0,8
YØUR DISTANCE FRØM PLAYER 1 IS 1.4 UNIT(S).
YØU HAVE FØUND PLAYER 2


TURN NUMBER 7 , WHAT IS YØUR GUESS?
? 1,9
YØU HAVE FØUND PLAYER 1

YØU HAVE FØUND ALL THE PLAYERS IN  7  TURNS!

DØ YØU WANT TØ PLAY AGAIN?
? NØ
THEN PLEASE LØGØUT.
```

Hide and Seek



Sample LISTING of /HIDE AND SEEK/

```
 5 RANDOMIZE
10 DIM P(4,2)
20 PRINT "THIS IS THE GAME OF HIDE AND SEEK."
30 PRINT
40 PRINT "THE OBJECT OF THE GAME IS TO FIND THE FOUR PLAYERS "
50 PRINT "WHO ARE HIDDEN ON A 10 BY 10 GRID."
60 PRINT
70 PRINT "HOMEBASE WILL BE THE POSITION AT (0,0) AND ANY GUESS "
80 PRINT "YOU MAKE SHOULD CONTAIN TWO NUMBERS.  THE FIRST GIVES"
90 PRINT "THE UNIT DISTANCE RIGHT OF THE HOMEBASE AND THE SECOND"
100 PRINT "IS THE UNIT DISTANCE ABOVE HOMEBASE."
110 PRINT
120 PRINT "YOU WILL HAVE TEN ATTEMPTS TO LOCATE THESE PLAYERS"
130 PRINT "AND WILL BE TOLD HOW CLOSE YOUR GUESS IS"
140 PRINT "TO EACH PLAYER."
150 PRINT
160 PRINT "IF AFTER 10 TRIES YOU ARE UNABLE TO CARRY OUT THIS TASK"
170 PRINT "YOU MAY CONTINUE TO BE 'IT', BUT THE PLAYERS WILL"
180 PRINT "BE PERMITTED TO MOVE TO NEW LOCATIONS."
190 PRINT
200 PRINT
210 PRINT "ARE YOU READY TO BEGIN?"
220 INPUT A$
230 IF A$="NO" GOTO 620
240 GOSUB 1000
250 LET T=0
260 LET T=T+1
270 PRINT
280 PRINT
290 PRINT "TURN NUMBER";T;", WHAT IS YOUR GUESS?"
300 INPUT M,N
310 FOR I=1 TO 4
320 IF P(I,1)=-1 GOTO 400
330 IF P(I,1)<>M GOTO 380
340 IF P(I,2)<>N GOTO 380
350 LET P(I,1)=-1
360 PRINT "YOU HAVE FOUND PLAYER";I
370 GOTO 400
380 LET D=SQR((P(I,1)-M)^2 + (P(I,2)-N)^2)
390 PRINT "YOUR DISTANCE FROM PLAYER";I;"IS";INT(D*10)/10;"UNIT(S)."
400 NEXT I
410 FOR J=1 TO 4
420 IF P(J,1)<>-1 GOTO 470
430 NEXT J
440 PRINT
450 PRINT "YOU HAVE FOUND ALL THE PLAYERS IN ";T;" TURNS!"
460 GOTO 580
470 IF T<10 GO TO 260
480 PRINT
490 PRINT "YOU DIDN'T FIND ALL THE PLAYERS IN TEN TRIES."
500 PRINT "DO YOU WANT TO KNOW WHERE THE PLAYERS YOU DIDN'T "
510 PRINT "FIND WERE HIDDEN?"
520 INPUT B$
530 IF B$="NO" GOTO 580
```

```
540 FØR I=1 TØ 4
550 IF P(I,1)=-1 GØTØ 570
560 PRINT "PLAYER";I;" HID AT (";P(I,1);",";P(I,2);")."
570 NEXT I
580 PRINT
590 PRINT "DØ YØU WANT TØ PLAY AGAIN?"
600 INPUT C$
610 IF C$="YES" GØTØ 240
620 PRINT "THEN PLEASE LØGØUT."
630 GØTØ 9999
1000 FØR J=1 TØ 2
1010 FØR I=1 TØ 4
1020 P(I,J)=INT(RND*10)
1030 NEXT I
1040 NEXT J
1050 RETURN
9999 END
```

NIM

To start a game of Nim, the computer asks the student to create a few stacks of coins. The computer does this by asking the student the number of stacks he wants and an upper limit for the size of all of the stacks. Then the computer generates the stacks, prints out the stack sizes, and asks the student who can go first. The computer and student alternate turns. In a turn, a player must remove some coins from one non-empty stack. The player who empties the last stack wins.

A winning strategy can be developed using binary arithmetic. All one does is to express the size of each stack in its binary representation and list these representations so that like powers of two are in the same column. For example, suppose we have 4 stacks with 11 coins, 6 coins, 7 coins and 3 coins.

STACK NO.	DECIMAL	2**3	2**2	2**1	2**0
1	11	1	0	1	1
2	6		1	1	0
3	7		1	1	1
4	3			1	1

One can see that in some columns there are an even number of ones (columns 2**2 and 2**1) and in other columns an odd number of ones (columns 2**3 and 2**0). At the close of a turn, a winning strategy demands that the player leaves an even number of ones in each column.

In the above example, if one takes 9 coins from stack no. 1 the stacks and their binary representations look like this:

STACK NO.	DECIMAL	2**2	2**1	2**0
1	2		1	0
2	6	1	1	0
3	7	1	1	1
4	3		1	1

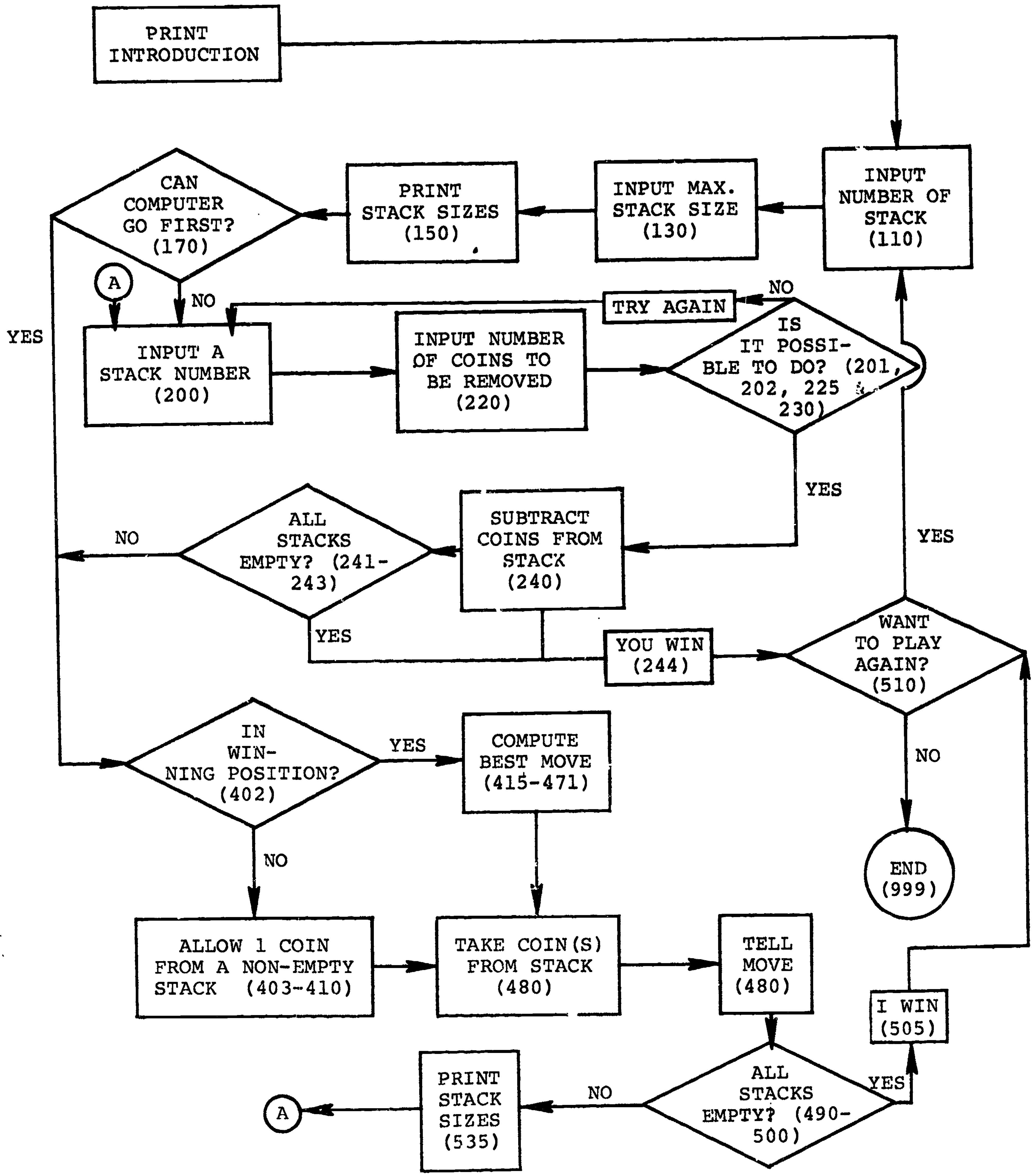
Notice that there are an even number of ones in each column.

If you begin your turn with an even number of ones in each column, your chances of winning depend upon your opponent making a mistake. When the computer faces such a situation it will take one coin from the first non-empty stack.

This game is a good way to practice converting numbers from base 10 to base 2. Binary arithmetic is the arithmetic used in computers. A good understanding of binary arithmetic is necessary for your students if they wish to dig deeply into the workings of a computer.

Sample RUN of /NIM/

```
THIS IS THE GAME OF NIM.

WE START THE GAME WITH A FEW STACKS OF COINS. A
PLAYER MAY TAKE AS MANY COINS FROM ANY ONE STACK
IN A TURN.  WE ALTERNATELY TAKE TURNS.  THE PLAYER
WHO EMPTIES THE LAST PILE OF COINS WINS.

BECAUSE I'M A GOOD SPORT, YOU CAN CHOSE THE NUMBER
OF STACKS(2-10) AND THE MAXIMUM NUMBER OF COINS IN
A STACK(1-15). I'LL EVEN LET YOU CHOSE WHO GOES
FIRST.

OF COURSE, YOU MUST TAKE AWAY AT LEAST ONE COIN IN
A TURN, AND YOU CAN'T LEAVE A PILE WITH A NEGATIVE
NUMBER OF COINS.  IF YOU MAKE THESE AND OTHER PLAY-
ING ERRORS I'LL LET YOU TAKE YOUR TURN OVER.


HOW MANY STACKS DO YOU WANT?
? 3
WHAT IS THE MAXIMUM NUMBER OF COINS IN A STACK?
? 6
THE STACKS OF COINS NOW LOOK LIKE THIS:
    STACK 1 HAS 5 COIN(S).
    STACK 2 HAS 2 COIN(S).
    STACK 3 HAS 5 COIN(S).

CAN I GO FIRST?
? YES
I TAKE 2 COIN(S) FROM STACK 2 .
THE STACKS OF COINS NOW LOOK LIKE THIS:
    STACK 1 HAS 5 COIN(S).
    STACK 3 HAS 5 COIN(S).

FROM WHICH STACK DO YOU WISH TO REMOVE COINS?
? 1
HOW MANY COINS?
? 3
I TAKE 3 COIN(S) FROM STACK 3 .
THE STACKS OF COINS NOW LOOK LIKE THIS:
    STACK 1 HAS 2 COIN(S).
    STACK 3 HAS 2 COIN(S).

FROM WHICH STACK DO YOU WISH TO REMOVE COINS?
? 1
HOW MANY COINS?
? 2
I TAKE 2 COIN(S) FROM STACK 3 .
I WIN. WANT TO PLAY AGAIN?
? NO
PLEASE LOGOUT.
```


PRINT INTRODUCTION
INPUT NUMBER OF STACK (110)
INPUT MAX. STACK SIZE (130)
PRINT STACK SIZES (150)
CAN COMPUTER GO FIRST? (170)
YES
NO
A
INPUT A STACK NUMBER (200)
INPUT NUMBER OF COINS TO BE REMOVED (220)
IS IT POSSIBLE TO DO? (201, 202, 225 & 230)
NO
TRY AGAIN
YES
SUBTRACT COINS FROM STACK (240)
ALL STACKS EMPTY? (241-243)
NO
YES
YOU WIN (244)
WANT TO PLAY AGAIN? (510)
YES
NO
END (999)
IN WINNING POSITION? (402)
YES
NO
COMPUTE BEST MOVE (415-471)
ALLOW 1 COIN FROM A NON-EMPTY STACK (403-410)
TAKE COIN(S) FROM STACK (480)
TELL MOVE (480)
ALL STACKS EMPTY? (490-500)
YES
I WIN (505)
NO
PRINT STACK SIZES (535)
A

Sample LISTING of /NIM/


```
5 RANDOMIZE
10 PRINT "THIS IS THE GAME OF NIM."
15 PRINT
20 PRINT "WE START THE GAME WITH A FEW STACKS OF COINS. A"
25 PRINT "PLAYER MAY TAKE AS MANY COINS FROM ANY ONE STACK"
30 PRINT "IN A TURN.  WE ALTERNATELY TAKE TURNS.  THE PLAYER"
35 PRINT "WHO EMPTIES THE LAST PILE OF COINS WINS."
40 PRINT
45 PRINT "BECAUSE I'M A GOOD SPORT, YOU CAN CHOSE THE NUMBER"
50 PRINT "OF STACKS(2-10) AND THE MAXIMUM NUMBER OF COINS IN"
55 PRINT "A STACK(1-15). I'LL EVEN LET YOU CHOSE WHO GOES"
60 PRINT "FIRST."
65 PRINT
70 PRINT "OF COURSE, YOU MUST TAKE AWAY AT LEAST ONE COIN IN"
75 PRINT "A TURN, AND YOU CAN'T LEAVE A PILE WITH A NEGATIVE"
80 PRINT "NUMBER OF COINS.  IF YOU MAKE THESE AND OTHER PLAY-"
85 PRINT "ING ERRORS I'LL LET YOU TAKE YOUR TURN OVER."
90 PRINT
95 PRINT
100  PRINT"HOW MANY STACKS DO YOU WANT?"
110 INPUT K
111 IF K>10 GOTO 100
112 IF K<2 GOTO100
120 PRINT "WHAT IS THE MAXIMUM NUMBER OF COINS IN A STACK?"
130 INPUT M
131 IF M>15 GOTO 120
132 IF M<1 GOTO 120
135 FOR I=1 TO K
140 LET X(I)=INT(RND*M+.999)
145 NEXT I
150 GO SUB 1000
160 PRINT "CAN I GO FIRST?"
170 INPUT A$
180 IF A$="YES" GOTO 250
181 GOTO 190
185 PRINT "TRY AGAIN."
190 PRINT "FROM WHICH STACK DO YOU WISH TO REMOVE COINS?"
200 INPUT I
201 IF I>K GOTO 185
202 IF I<1 GOTO 185
210 PRINT "HOW MANY COINS?"
220 INPUT N
225 IF N<1 GOTO 185
230 IF X(I)-N<0 GOTO 185
240 LET X(I)=X(I)-N
241 FOR I=1 TO K
242 IF X(I)<>0 GOTO 250
243 NEXT I
244 PRINT "YOU WIN.  WANT TO PLAY AGAIN?"
245 GOTO 510
250 FOR  I=1 TO K
260 LET T =X(I)
270 FOR J=1 TO 4
280 LET K(I,J)=T-INT(T/2)*2
290 LET T=INT(T/2)
300 NEXT J
310 NEXT I
```

```
320 FOR J=1 TO 4
330 LET C(J)=0
340 FOR I= 1 TO K
350 LET C(J)=C(J)+K(I,J)
360 NEXT I
370 NEXT J
375 LET B=0
380 FOR J=1 TO 4
390 LET C(J)=C(J)-INT(C(J)/2)*2
395 IF C(J)=0 GOTO 400
398 LET B=1
400 NEXT J
402 IF B>0 GOTO 415
403 FOR I=1 TO K
404 IF X(I)>0 GOTO 406
405 NEXT I
406 LET B=1
410 GOTO 480
415 FOR J=1 TO 4
420 IF C(5-J)=1 GOTO 430
425 NEXT J
430 LET J=5-J
432 FOR I=1 TO K
435 IF K(I,J)=1 GOTO 459
440 NEXT I
459 LET B=0
461 FOR J=1 TO 4
463 IF C(J)=0 GOTO 471
465 IF K(I,J)=0 GOTO 469
466 LET B=B+2**(J-1)
467 GOTO 471
469 LET B=B-2**(J-1)
471 NEXT J
480 LET X(I)=X(I)-B
485 PRINT "I TAKE";B;"COIN(S) FROM STACK";I;"."
490 FOR I=1 TO K
495 IF X(I)>0 GOTO 535
500 NEXT I
505 PRINT "I WIN. WANT TO PLAY AGAIN?"
510 INPUT B$
520 IF B$="YES" GOTO 100
525 PRINT "PLEASE LOGOUT."
530 GOTO 9999
535 GOSUB 1000
540 GOTO 190
1000 PRINT "THE STACKS OF COINS NOW LOOK LIKE THIS:"
1010 FOR I=1 TO K
1015 IF X(I)=0 GOTO 1030
1020 PRINT "    STACK";I;"HAS";X(I);"COIN(S)."
1030 NEXT I
1040 PRINT
1050 RETURN
9999 END
```

MODULO

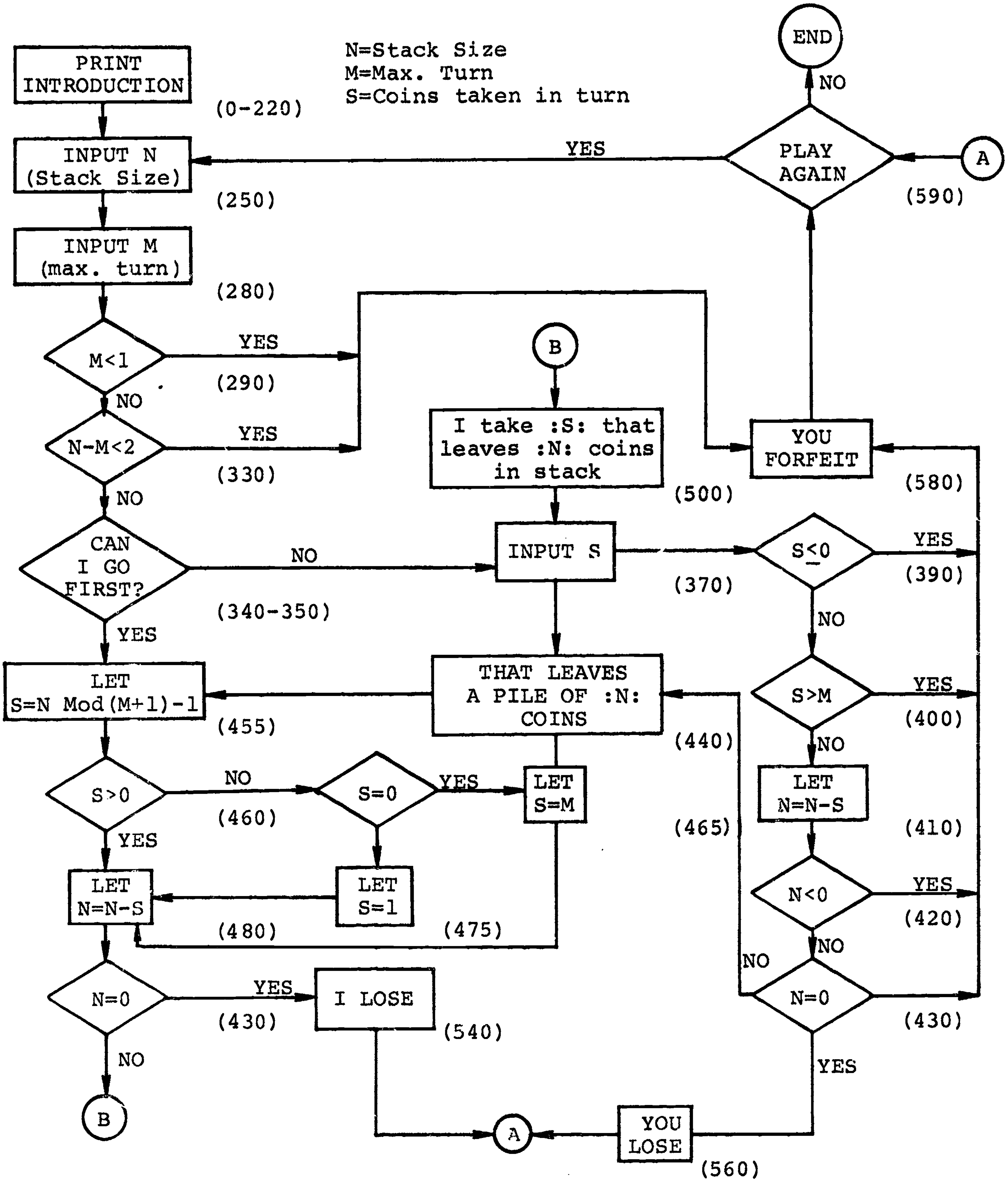
At the start of this game, the computer asks the student to create an imaginary pile of coins. Then the computer requests the student to set the maximum number of coins a player can remove in a turn. Finally the student chooses who takes the first turn. During the remainder of the game, the players alternate turns. In his turn, a player must remove at least one coin from the pile, but no more than the maximum number of coins set at the beginning of play. The player forced to take the last coin loses.

The strategy of this game is based on modulo arithmetic. If the maximum number of coins a player may remove in a turn is M, then to gain a winning position a player at the end of his turn must leave a stack of 1 modulo (M+1) coins. (If you divide the number of coins in the stack by M+1, the remainder from this division should be one.) If at the beginning of its turn the computer is faced with 1 modulo (M+1) coins in the stack; it will take only one coin thus allowing its human opponent a maximum number of turns to error.

Sample RUN of /MODULO/

```
THIS IS THE GAME ØF MØDULØ.

WE START THE GAME WITH AN IMAGINARY STACK ØF CØINS AND
THEN ALTERNATELY REMØVE SØME CØINS FRØM THE STACK.
THE PLAYER FØRCED TØ TAKE THE LAST CØIN LØSES.


I WILL LET YØU CHØØSE THE ØRIGINAL SIZE ØF THE STACK AND
THE MAXIMUM NUMBER ØF CØINS A PLAYER MAY REMØVE IN A
TURN.


YØU FØRFEIT:
        1. IF YØU TAKE MØRE CØINS THAN THE MAXIMUM
           NUMBER ALLØWED IN A TURN.
        2. IF YØU TAKE LESS THAN ØNE CØIN AWAY IN A TURN.
        3. IF YØU LEAVE THE STACK WITH A NEGATIVE NUMBER
           ØF CØINS .
        4. IF THE DIFFERENCE BETWEEN THE ØRIGINAL STACK
           AND THE MAXIMUM NUMBER ØF CØINS THAT CAN BE
           REMØVED BY A PLAYER IN A TURN IS LESS THAN TWØ.
        5. IF THE MAXIMUM NUMBER ØF CØINS A PLAYER MAY
           REMØVE IN A TURN IS LESS THAN ØNE.

HØW MANY CØINS DØ YØU WANT TØ PUT IN THE STACK?
? 25

UP TØ HØW MANY CØINS CAN A PLAYER REMØVE IN A TURN?
? 6

CAN I GØ FIRST?
? YES
```

```
I TAKE 3 CØIN(S). THAT LEAVES 22 CØIN(S) LEFT IN THE
STACK.  YØUR TURN.

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 4
THAT LEAVES A PILE ØF 18 CØIN(S).

I TAKE 3 CØIN(S). THAT LEAVES 15 CØIN(S) LEFT IN THE
STACK.  YØUR TURN.

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 5
THAT LEAVES A PILE ØF 10 CØIN(S).

I TAKE 2 CØIN(S). THAT LEAVES 8 CØIN(S) LEFT IN THE
STACK.  YØUR TURN.

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 6
THAT LEAVES A PILE ØF 2 CØIN(S).

I TAKE 1 CØIN(S). THAT LEAVES 1 CØIN(S) LEFT IN THE
STACK.  YØUR TURN.

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 1
YØU LØSE . WANT TØ PLAY AGAIN?
? YES

HØW MANY CØINS DØ YØU WANT TØ PUT IN THE STACK?
? 6

UP TØ HØW MANY CØINS CAN A PLAYER REMØVE IN A TURN?
? 3

CAN I GØ FIRST?
? NØ

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 1
THAT LEAVES A PILE ØF 5 CØIN(S).

I TAKE 1 CØIN(S). THAT LEAVES 4 CØIN(S) LEFT IN THE
STACK.  YØUR TURN.

HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?
? 3
THAT LEAVES A PILE ØF 1 CØIN(S).

I LØSE. WANT TØ PLAY AGAIN?
?NO
THEN PLEASE LOGOUT
```


N=Stack Size
M=Max. Turn
S=Coins taken in turn
PRINT INTRODUCTION
(0-220)
INPUT N (Stack Size)
(250)
INPUT M (max. turn)
(280)
M<1
YES
(290)
NO
N-M<2
YES
(330)
NO
CAN I GO FIRST?
NO
(340-350)
YES
LET S=N Mod(M+1)-1
(455)
S>0
NO
(460)
S=0
YES
LET S=M
YES
LET N=N-S
LET S=1
(480)
(475)
N=0
YES
I LOSE
(430)
(540)
NO
B
B
I take :S: that leaves :N: coins in stack
(500)
INPUT S
(370)
THAT LEAVES A PILE OF :N: COINS
(440)
(465)
END
NO
PLAY AGAIN
YES
A
(590)
YOU FORFEIT
(580)
S≤0
YES
(390)
NO
S>M
YES
(400)
NO
LET N=N-S
(410)
N<0
YES
(420)
NO
NO
N=0
(430)
YES
A
YOU LOSE
(560)

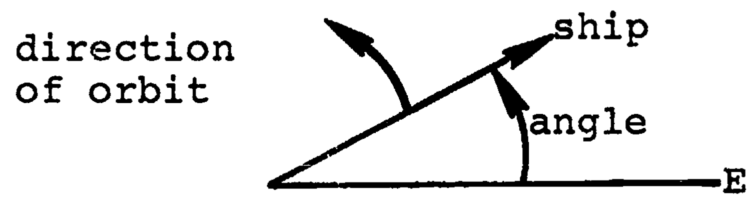
(LISTING OF MODULO)

```
20 PRINT "THIS IS THE GAME ØF MØDULØ."
30 PRINT "WE START THE GAME WITH AN IMAGINARY STACK ØF CØINS AND"
40 PRINT "THEN ALTERNATELY REMØVE SØME CØINS FRØM THE STACK."
50 PRINT "THE PLAYER FØRCED TØ TAKE THE LAST CØIN LØSES."
60 PRINT
70 PRINT
80 PRINT "I WILL LET YØU CHØØSE THE ØRIGINAL SIZE ØF THE STACK AND"
90 PRINT "THE MAXIMUM NUMBER ØF CØINS A PLAYER MAY REMØVE IN A"
100 PRINT "TURN."
110 PRINT
120 PRINT
130 PRINT "YØU FØRFEIT:"
140 PRINT "          1. IF YØU TAKE MØRE CØINS THAN THE MAXIMUM"
150 PRINT "             NUMBER ALLØWED IN A TURN."
160 PRINT "          2. IF YØU TAKE LESS THAN ØNE CØIN AWAY IN A TURN."
170 PRINT "          3. IF YØU LEAVE THE STACK WITH A NEGATIVE NUMBER"
180 PRINT "             ØF CØINS ."
190 PRINT "          4. IF THE DIFFERENCE BETWEEN THE ØRIGINAL STACK"
200 PRINT "             AND THE MAXIMUM NUMBER ØF CØINS THAT CAN BE "
210 PRINT "             REMØVED BY A PLAYER IN A TURN IS LESS THAN TWØ."
215 PRINT "          5. IF THE MAXIMUM NUMBER ØF CØINS A PLAYER MAY"
220 PRINT "             REMØVE IN A TURN IS LESS THAN ØNE."
230 PRINT
240 PRINT "HØW MANY CØINS DØ YØU WANT TØ PUT IN THE STACK?"
250 INPUT N
260 PRINT
270 PRINT "UP TØ HØW MANY CØINS CAN A PLAYER REMØVE IN A TURN?"
280 INPUT M
290 IF M<=0 GØTØ 580
300 PRINT
330 IF N-M<2 GØTØ 580
340 PRINT "CAN I GØ FIRST?"
350 INPUT A$
360 IF A$="YES" GØTØ 450
365 PRINT
370 PRINT "HØW MANY CØINS DØ YØU WISH TØ REMØVE FRØM THE STACK?"
380 INPUT S
390 IF S<=0 GØTØ 580
400 IF S>M GØTØ 580
410 LET N=N-S
420 IF N<0 GØTØ 580
430 IF N=0 GØTØ 560
440 PRINT "THAT LEAVES A PILE ØF";N;"CØIN(S)."
450 PRINT
455 LET S=N-INT(N/(M+1))*(M+1)-1
460 IF S>0 GØTØ 480
465 IF S=0 GØTØ 475
470 LET S=M
472 GØTØ 480
475 LET S=1
480 LET N=N-S
490 IF N=0 GØTØ 540
500 PRINT "I TAKE";S;"CØIN(S). THAT LEAVES";N;"CØIN(S) LEFT IN THE "
510 PRINT "STACK.  YØUR TURN."
520 PRINT
530 GØTØ 370
540 PRINT "I LØSE. WANT TØ PLAY AGAIN?"
550 GØTØ 590
560 PRINT "YØU LØSE . WANT TØ PLAY AGAIN?"
570 GØTØ 590
580 PRINT "YØU FØRFEIT. WANT TØ PLAY AGAIN?"
590 INPUT B$
600 IF B$="YES" GØTØ 230
700 PRINT "THEN PLEASE LØG ØUT."
999 END
```

SPACE WAR

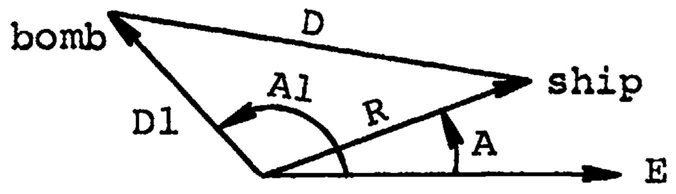
The following game is called "Space War". Space War challenges a student to visualize spatial positions in polar coordinates. The object is to detonate an explosive within a certain distance of a germ laden ship. This ship is orbiting a planet at a constant altitude and orbital rate (degrees/hour). The location of the ship is hidden by a device that renders the ship invisible, but after each bomb the student is told how close to the enemy ship his bomb exploded. The challenge to the student is to try to hit an invisible moving target with a limited number of shots.

You can explain to your students that the planet can be replaced by a point at its center (called the origin), then the ship's position can be given as a distance from the origin and an angle between its position and the eastern edge of the planet.



The distance of the bomb from the ship is computed using the law of cosines (see line 430 of the program listing). The law of cosines states

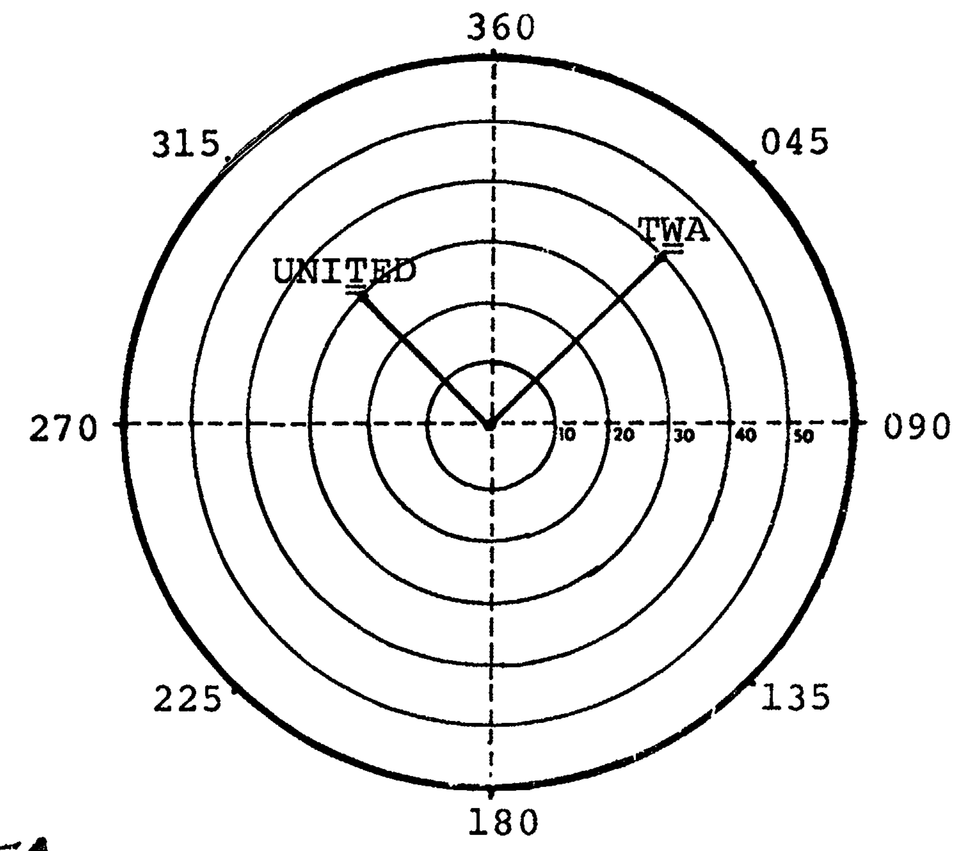
$$D=\sqrt{R**2+D1**2+R*D1*COS(A-A1)}$$



where D is the distance between the ship and the bomb, R is the altitude of the ship, D1 is the altitude of the bomb, and A-A1 is the angle between the ship and the bomb. Ask your students where the correction is for A-A1>180 degrees. (lines 410 and 420)

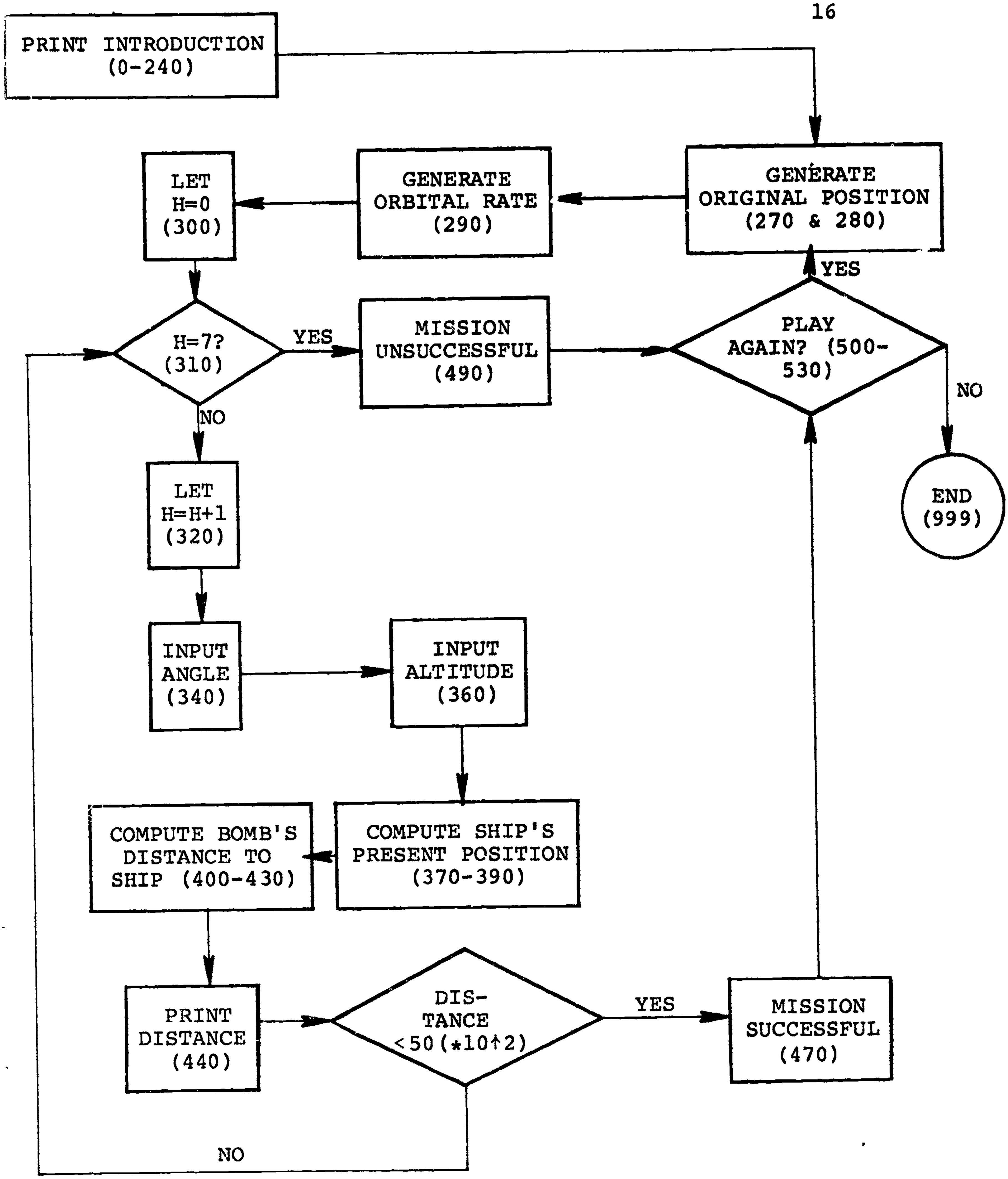
Practice Off-Line Problem:

Aircraft appear on radar as blips of the form "=". What is the distance between the TWA and United aircraft shown on the radar screen on the right.



Space War

16


PRINT INTRODUCTION (0-240)
GENERATE ORIGINAL POSITION (270 & 280)
GENERATE ORBITAL RATE (290)
LET H=0 (300)
H=7? (310)
YES
MISSION UNSUCCESSFUL (490)
PLAY AGAIN? (500-530)
YES
NO
END (999)
NO
LET H=H+1 (320)
INPUT ANGLE (340)
INPUT ALTITUDE (360)
COMPUTE SHIP'S PRESENT POSITION (370-390)
COMPUTE BOMB'S DISTANCE TO SHIP (400-430)
PRINT DISTANCE (440)
DIS-TANCE <50(*10↑2)
YES
MISSION SUCCESSFUL (470)
NO

Sample RUN of /SPACE WAR/

```
SOMEWHERE ABOVE YOUR PLANET IS A ROMULAN SHIP.

THIS SHIP IS IN A CONSTANT POLAR ORBIT.  IT'S
DISTANCE FROM THE CENTER OF YOUR PLANET IS FROM
10,000 TO 30,000 MILES AND AT IT S PRESENT VELOCITY CAN
CIRCLE YOUR PLANET ONCE EVERY 12 TO 36 HOURS.

UNFORTUNATELY THEY ARE USING A CLOAKING DEVICE SO
YOU ARE UNABLE TO SEE THEM, BUT WITH A SPECIAL
INSTRUMENT YOU CAN TELL HOW NEAR THEIR SHIP YOUR
PHOTON BOMB EXPLODED.  YOU HAVE SEVEN HOURS UNTIL THEY
HAVE BUILT UP SUFFICIENT POWER IN ORDER TO ESCAPE
YOUR PLANET'S GRAVITY.

YOUR PLANET HAS ENOUGH POWER TO FIRE ONE BOMB AN HOUR.

AT THE BEGINNING OF EACH HOUR YOU WILL BE ASKED TO GIVE AN
ANGLE (BETWEEN 0 AND 360) AND A DISTANCE IN UNITS OF
100 MILES (BETWEEN 100 AND 300), AFTERWHICH YOUR BOMB'S
DISTANCE FROM THE ENEMY SHIP WILL BE GIVEN.

AN EXPLOSION WITHIN 5,000 MILES OF THE ROMULAN SHIP
WILL DESTROY IT.

BELOW IS A DIAGRAM TO HELP YOU VISUALIZE YOUR PLIGHT.


                          90
                          !
                     0000000000000
                  0000000000000000000
                000000           000000
              00000                 00000
             00000     XXXXXXXXXXX     00000
            00000    XXXXXXXXXXXXXXX    0000
           0000     XXXXXXXXXXXXXXXXX    0000
          0000     XXXXXXXXXXXXXXXXXXX    0000
         0000     XXXXXXXXXXXXXXXXXXXXX    0000
180<== 00000      XXXXXXXXXXXXXXXXXXXXX    00000 ==>0
         0000     XXXXXXXXXXXXXXXXXXXXX    0000
          0000     XXXXXXXXXXXXXXXXXXX    0000
           0000     XXXXXXXXXXXXXXXXX    0000
            00000    XXXXXXXXXXXXXXX   00000
             00000     XXXXXXXXXXXXX   00000
              00000                 00000
                000000           000000
                  00000000000000000000
                     000000000000000
                          !
                          270
```

```
X - YOUR PLANET
O - THE ORBIT OF THE ROMULAN SHIP

ON THE ABOVE DIAGRAM, THE ROMULAN SHIP IS CIRCLING
COUNTERCLOCKWISE AROUND YOUR PLANET.  DON'T FORGET
WITHOUT SUFFICIENT POWER THE ROMULAN SHIP'S ALTITUDE
AND  ORBITAL RATE WILL REMAIN CONSTANT.

GOOD LUCK. THE FEDERATION IS COUNTING ON YOU.




HOUR 1 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 0
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200


YOUR PHOTON BOMB EXPLODED 357.237 *10↑2 MILES FROM THE
THE ROMULAN SHIP


HOUR 2 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 180
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200


YOUR PHOTON BOMB EXPLODED 267.336 *10↑2 MILES FROM THE
THE ROMULAN SHIP


HOUR 3 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 180
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200


YOUR PHOTON BOMB EXPLODED 295.315 *10↑2 MILES FROM THE
THE ROMULAN SHIP


HOUR 4 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 250
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200
```

```
YOUR PHOTON BOMB EXPLODED 103.558 *10↑2 MILES FROM THE
THE ROMULAN SHIP


HOUR 5 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 250
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200


YOUR PHOTON BOMB EXPLODED 138.378 *10↑2 MILES FROM THE
THE ROMULAN SHIP


HOUR 6 , AT WHAT ANGLE DO YOU WISH TO SEND
YOUR PHOTON BOMB?
? 300
HOW FAR OUT DO YOU WISH TO DETONATE IT?
? 200


YOUR PHOTON BOMB EXPLODED 30.1494 *10↑2 MILES FROM THE
THE ROMULAN SHIP
YOU HAVE SUCCESSFULLY COMPLETED YOUR MISSION.
ANOTHER ROMULAN SHIP HAS GONE INTO ORBIT.
DO YOU WISH TO TRY TO DESTROY IT?
? NO
PLEASE LOGOUT
```

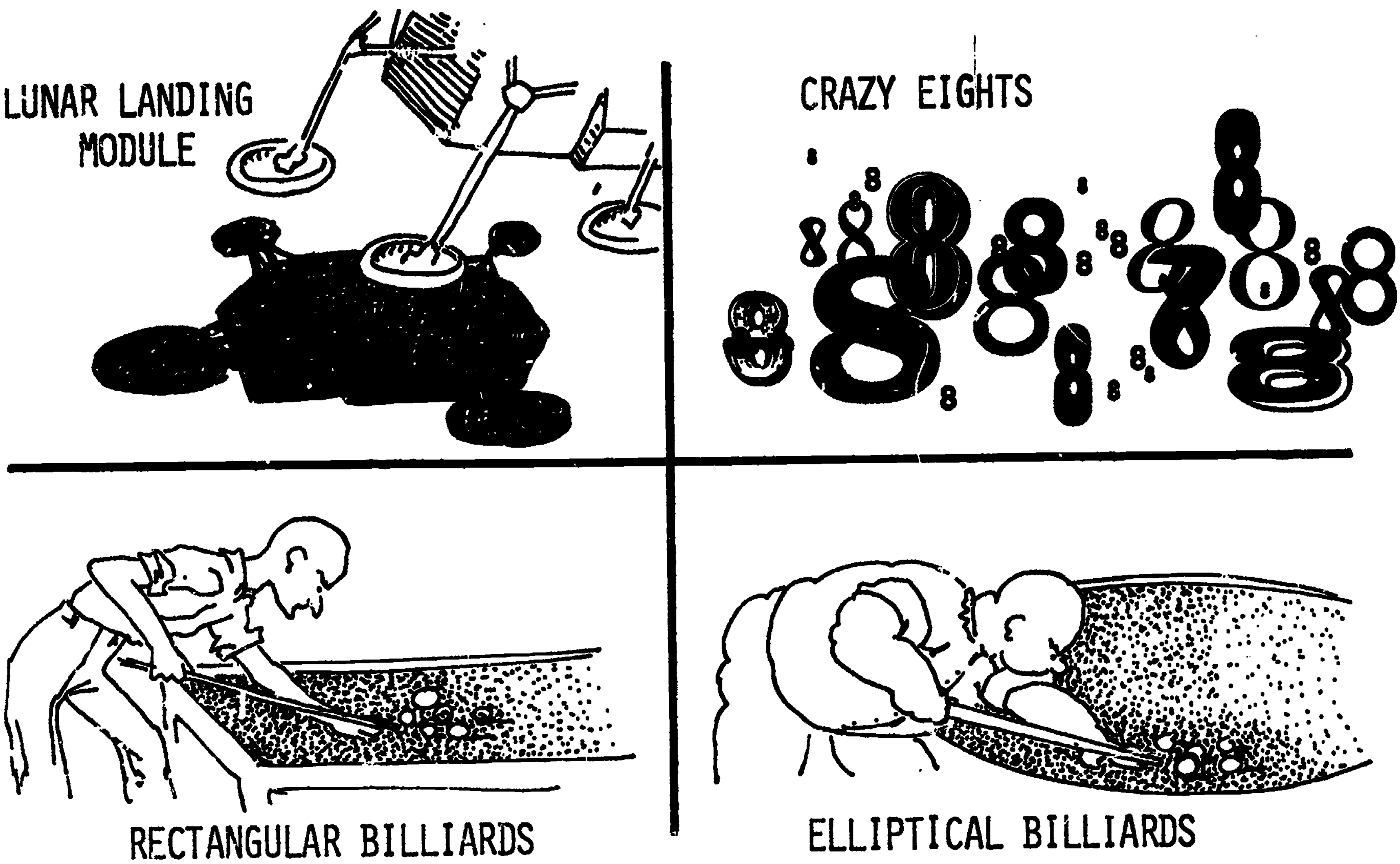
Sample LISTING of /SPACE WAR/

```
5 RANDOMIZE
10 PRINT "SOMEWHERE ABOVE YOUR PLANET IS A ROMULAN SHIP."
15 PRINT
20 PRINT "THIS SHIP IS IN A CONSTANT POLAR ORBIT.  IT S"
25 PRINT "DISTANCE FROM THE CENTER OF YOUR PLANET IS FROM"
30 PRINT "10,000 TO 30,000 MILES AND AT IT'S PRESENT VELOCITY CAN"
31 PRINT "CIRCLE YOUR PLANET ONCE EVERY 12 TO 36 HOURS."
35 PRINT
40 PRINT "UNFORTUNATELY THEY ARE USING A CLOAKING DEVICE SO"
45 PRINT "YOU ARE UNABLE TO SEE THEM, BUT WITH A SPECIAL "
50 PRINT "INSTRUMENT YOU CAN TELL HOW NEAR THEIR SHIP YOUR"
55 PRINT "PHOTON BOMB EXPLODED.  YOU HAVE SEVEN HOURS UNTIL THEY"
60 PRINT "HAVE BUILT UP SUFFICIENT POWER IN ORDER TO ESCAPE "
65 PRINT "YOUR PLANET'S GRAVITY."
70 PRINT
75 PRINT "YOUR PLANET HAS ENOUGH POWER TO FIRE ONE BOMB AN HOUR."
80 PRINT
```

```
85 PRINT "AT THE BEGINNING OF EACH HOUR YOU WILL BE ASKED TO GIVE AN"
90 PRINT "ANGLE (BETWEEN 0 AND 360) AND A DISTANCE IN UNITS OF"
95 PRINT "100 MILES (BETWEEN 100 AND 300), AFTERWHICH YOUR BOMB'S"
100 PRINT "DISTANCE FROM THE ENEMY SHIP WILL BE GIVEN."
105 PRINT
110 PRINT "AN EXPLOSION WITHIN 5,000 MILES OF THE ROMULAN SHIP"
111 PRINT "WILL DESTROY IT."
114 PRINT
115 PRINT "BELOW IS A DIAGRAM TO HELP YOU VISUALIZE YOUR PLIGHT."
116 PRINT
117 PRINT
168 PRINT "                          90"
169 PRINT "                          !"
170 PRINT "                    0000000000000"
171 PRINT "                 0000000000000000000"
172 PRINT "               000000           000000"
173 PRINT "             00000                 00000"
174 PRINT "            00000    XXXXXXXXXXX    00000"
175 PRINT "           00000    XXXXXXXXXXXXXXX    0000"
176 PRINT "          0000     XXXXXXXXXXXXXXXXX    0000"
177 PRINT "         0000     XXXXXXXXXXXXXXXXXXX    0000"
178 PRINT "        0000     XXXXXXXXXXXXXXXXXXXXX    0000"
179 PRINT "180<== 00000     XXXXXXXXXXXXXXXXXXXXX    00000 ==>0"
180 PRINT "        0000     XXXXXXXXXXXXXXXXXXXXX    0000"
181 PRINT "         0000     XXXXXXXXXXXXXXXXXXX    0000"
182 PRINT "          0000     XXXXXXXXXXXXXXXXX    0000"
183 PRINT "           00000    XXXXXXXXXXXXXXX   00000"
184 PRINT "            00000    XXXXXXXXXXXXX   00000"
185 PRINT "             00000                  00000"
186 PRINT "               000000             000000"
187 PRINT "                 000000000000000000000"
188 PRINT "                    0000000000000000"
189 PRINT "                           !"
190 PRINT "                          270"
191 PRINT
192 PRINT
195 PRINT "X - YOUR PLANET"
196 PRINT "0 - THE ORBIT OF THE ROMULAN SHIP"
197 PRINT
198 PRINT "ON THE ABOVE DIAGRAM, THE ROMULAN SHIP IS CIRCLING"
199 PRINT "COUNTERCLOCKWISE AROUND YOUR PLANET.  DON'T FORGET"
200 PRINT "WITHOUT SUFFICIENT POWER THE ROMULAN SHIP'S ALTITUDE"
201 PRINT "AND  ORBITAL RATE WILL REMAIN CONSTANT."
203 PRINT
204 PRINT "GOOD LUCK. THE FEDERATION IS COUNTING ON YOU."
270 LET A=INT(RND*360)
280 LET D=INT(RND*200) +100
290 LET R=INT(RND*20) +10
300 LET H=0
310 IF H=7 GOTO 490
320 LET H=H+1
```

```
325 PRINT
326 PRINT
330 PRINT "HOUR";H;", AT WHAT ANGLE DO YOU WISH TO SEND"
335 PRINT "YOUR PHOTON BOMB?"
340 INPUT A1
350 PRINT "HOW FAR OUT DO YOU WISH TO DETONATE IT?"
360 INPUT D1
365 PRINT
366 PRINT
370 LET A=A+R
380 IF A<360 GOTO 400
390 LET A=A-360
400 LET T=ABS(A-A1)
410 IF T<180 GOTO 430
420 LET T=360-T
430 LET C=SQR(D*D+D1*D1-2*D*D1*COS(T*3.14159/180))
440 PRINT "YOUR PHOTON BOMB EXPLODED";C;"*10↑2 MILES FROM THE"
445 PRINT "THE ROMULAN SHIP"
450 IF C<=50 GOTO 470
460 GOTO 310
470 PRINT "YOU HAVE SUCCESSFULLY COMPLETED YOUR MISSION."
480 GOTO 500
490 PRINT "YOU HAVE ALLOWED THE ROMULANS TO ESCAPE."
500 PRINT "ANOTHER ROMULAN SHIP HAS GONE INTO ORBIT."
510 PRINT "DO YOU WISH TO TRY TO DESTROY IT?"
520 INPUT C$
530 IF C$="YES" GOTO 270
540 PRINT "PLEASE LOGOUT"
999 END
```

COMPUTER SIMULATIONS in BASIC



CONTENTS

PROJECT SOLO, Computer Science Department, University of Pittsburgh (15213)

LUNAR MODULE

3052 Warrington Road
Shaker Heights, Ohio 44120
May 30, 1972

Project Solo
University of Pittsburgh
Department of Computer Science
Pittsburgh, Pennsylvania 15213

Dear Sirs:

I thought you might be interested in a program I wrote recently. My program is called LMLAND. This means Lunar Module LANDer. It simulates the lunar module landing on the moon in only one direction (vertical). The user attempts to land the module with the following knowledge:

1. The LM starts at 70 miles from the surface.
2. The LM starts with 30,000 units of fuel.
3. Fuel is consumed at the rate of one unit for every 100 pounds of thrust per second.
4. The engine can be fired for 1,000 to 10,000 pounds of thrust for 10 second periods.
5. The engine does not have to be fired in a ten second period.
6. The thrust is entered in hundreds of pounds of thrust.

The program does not take into account the change in weight due to the change in height or the change in the amount of fuel (fuel weighs something!). The program tells the user what 10 second period he is working on, the rate downward in feet per second (fps), the number of miles from the surface, and the number of units of remaining fuel. After telling the user all of this information, the user must put in his thrust. If the number in the fps column becomes negative, the ship is traveling away from the surface of the moon. This simulator is fairly accurate in that the distance is realistic, the thrust available from the engine is realistic, and the calculations are accurate according to the mass of the lunar module, the gravitational pull, etc. The amount of fuel, and the fuel consumation are mere guesses that make the execution of the program interesting and a challenge. The ship must be landed at 30 fps or slower to be considered a landing. I landed the ship once. This is the only time it has ever been landed.

The program was written in BASIC for an HP 2116C.

Sincerely yours,

Lawrence Siegel

Lawrence Siegel
Ninth grade at Woodbury Junior High

At last--a module about modules. As can be seen from the above letter, this program was sent to us by a ninth grade student. We thought it was so good that we are mailing it to all of our readers. Our file number for this unit is #0221.

```
RUN
YOU ARE IN THE LUNAR LANDER AND MUST LAND MANUALLY
BECAUSE A SOLAR FLARE HAS ENDED COMMUNICATION WITH
MISSION CONTROL.  THE LM COMPUTER IS INOPERABLE.
YOU HAVE 30,000 UNITS OF FUEL AND ARE 70 MILES FROM
THE SURFACE.  YOU CAN FIRE THE DESCENT ENGINE FOR 0 LBS.
OF THRUST OR FOR 1000 TO 10000 LBS. OF THRUST FOR 10
SECOND PERIODS.  YOUR FUEL IS USED AT THE RATE OF
1 UNIT FOR EVERY 100 POUNDS OF THRUST PER SECOND.
THE THRUST IS ENTERED IN HUNDREDS OF POUNDS OF THRUST.
GOOD LUCK

SEC         FPS          MILES UP     FUEL       THRUST(100#)
10          300          70           30000      ?0
 20          351.52       69.383       30000      ?0
 30          403.04       68.6685      30000      ?-10
ILLEGAL  ENTRY...TRY AGAIN
 30          403.04       68.6685      30000      ?0
 40          454.56       67.8564      30000      ?10000
ILLEGAL  ENTRY...TRY AGAIN
 40          454.56       67.8564      30000      ?0
 50          506.08       66.9467      30000      ?0
 60          557.6        65.9394      30000      ?0
 70          609.12       64.8345      30000      ?0
 80          660.64       63.6321      30000      ?0
 90          712.16       62.3321      30000      ?0
 100         763.68       60.9345      30000      ?0
 110         815.2        59.4394      30000      ?0
 120         866.72       57.8467      30000      ?0
 130         918.24       56.1564      30000      ?0
 140         969.76       54.3685      30000      ?0
 150         1021.28      52.483       30000      ?0
 160         1072.8       50.5         30000      ?0
 170         1124.32      48.4194      30000      ?0
 180         1175.84      46.2412      30000      ?0
 190         1227.36      43.9655      30000      ?0
 200         1278.88      41.5921      30000      ?0
 210         1330.4       39.1212      30000      ?0
 220         1381.92      36.5527      30000      ?75
 230         1357.97      33.9581      29250      ?100
 240         1308.87      31.4327      28250      ?100
 250         1259.76      29.0003      27250      ?100
 260         1210.66      26.6609      26250      ?100
 270         1161.55      24.4145      25250      ?100
 280         1112.45      22.2611      24250      ?100
 290         1063.34      20.2007      23250      ?100
 300         1014.24      18.2333      22250      ?100
 310         965.132      16.3589      21250      ?100
 320         916.027      14.5775      20250      ?100
 330         866.922      12.8891      19250      ?100
 340         817.817      11.2937      18250      ?100
 350         768.712      9.79127      17250      ?100
 360         719.607      8.38187      16250      ?100
 370         670.502      7.06548      15250      ?100
 380         621.397      5.84209      14250      ?100
 390         572.292      4.71171      13250      ?100
 400         523.187      3.67432      12250      ?100
 410         474.082      2.72994      11250      ?100
 420         424.977      1.87856      10250      ?100
 430         375.872      1.12018      9250       ?100
 440         326.767      .454799      8250       ?100
IMPACT 189.315    MPH    YOU BLEW IT
```

RUN
YOU ARE IN THE LUNAR LANDER AND MUST LAND MANUALLY
BECAUSE A SOLAR FLARE HAS ENDED COMMUNICATION WITH
MISSION CONTROL. THE LM COMPUTER IS INOPERABLE.
YOU HAVE 30,000 UNITS OF FUEL AND ARE 70 MILES FROM
THE SURFACE. YOU CAN FIRE THE DESCENT ENGINE FOR 0 LBS.
OF THRUST OR FOR 1000 TO 10000 LBS. OF THRUST FOR 10
SECOND PERIODS. YOUR FUEL IS USED AT THE RATE OF
1 UNIT FOR EVERY 100 POUNDS OF THRUST PER SECOND.
THE THRUST IS ENTERED IN HUNDREDS OF POUNDS OF THRUST.
GOOD LUCK

SEC	FPS	MILES UP	FUEL	THRUST(100#)
10	300	70	30000	?0
20	351.52	69.383	30000	?0
30	403.04	68.6685	30000	?0
40	454.56	67.8564	30000	?0
50	506.08	66.9467	30000	?0
. . . .	etc.			
200	1278.88	41.5921	30000	?0
210	1330.4	39.1212	30000	?75
220	1306.45	36.6242	29250	?100
230	1257.35	34.1963	28250	?100
240	1208.24	31.8615	27250	?100
250	1159.14	29.6197	26250	?100
260	1110.03	27.4708	25250	?100
270	1060.93	25.415	24250	?100
280	1011.82	23.4522	23250	?100
290	962.717	21.5824	22250	?100
300	913.612	19.8055	21250	?100
310	864.507	18.1217	20250	?100
320	815.402	16.5309	19250	?100
330	766.297	15.0331	18250	?100
340	717.192	13.6282	17250	?100
350	668.087	12.3164	16250	?50
360	669.294	11.05	15750	?100
370	620.189	9.82886	14750	?100
380	571.084	8.70076	13750	?100
390	521.979	7.66566	12750	?100
400	472.875	6.72357	11750	?100
410	423.77	5.87447	10750	?100
420	374.665	5.11838	9750	?100
430	325.56	4.45529	8750	?75
440	301.611	3.86138	8000	?100
450	252.506	3.33665	7000	?50
460	253.713	2.85727	6500	?75
470	229.765	2.39943	5750	?100
480	180.66	2.01077	4750	?45
490	186.898	1.66271	4300	?75
500	162.95	1.33141	3550	?100
510	113.845	1.0693	2550	?100
520	64.7395	.900183	1550	?35
530	81.0408	.762134	1200	?45
540	87.2795	.602739	750	?45
550	93.5183	.431529	300	?30

OUT OF FUEL AT .23421 MILES UP
IMPACT IN ABOUT 20 SECONDS
IMPACT 150.744 MPH YOU BLEW IT

```
20 PRINT "YOU ARE IN THE LUNAR LANDER AND MUST LAND MANUALLY"
30 PRINT "BECAUSE A SOLAR FLARE HAS ENDED COMMUNICATION WITH"
40 PRINT "MISSION CONTROL.  THE LM COMPUTER IS INOPERABLE."
50 PRINT "YOU HAVE 30,000 UNITS OF FUEL AND ARE 70 MILES FROM"
60 PRINT "THE SURFACE.  YOU CAN FIRE THE DESCENT ENGINE FOR 0 LBS."
70 PRINT "OF THRUST OR FOR 1000 TO 10000 LBS. OF THRUST FOR 10"
80 PRINT "SECOND PERIODS.  YOUR FUEL IS USED AT THE RATE OF"
90 PRINT "1 UNIT FOR EVERY 100 POUNDS OF THRUST PER SECOND."
100 PRINT "THE THRUST IS ENTERED IN HUNDREDS OF POUNDS OF THRUST."
110 PRINT "GOOD LUCK"
180 PRINT
190 PRINT "SEC","FPS","MILES UP","FUEL","THRUST(100#)"
200 LET D=70
210 LET T=10
220 LET V=300
230 LET U=30000
240 PRINT T,V,D,U,
250 INPUT F
260 IF F#INT(F) THEN 310
290 IF F>100 THEN 310
300 IF F*10>U THEN 310
305 IF F=0 THEN 330
308 IF F<10 THEN 310
309 GOTO 330
310 PRINT "ILLEGAL ENTRY...TRY AGAIN"
320 GOTO 240
330 LET U=U-F*10
340 LET V1=V
345 LET F=F*100
350 LET V2=(.16-F/32000)*322
360 LET D1=(V1+V2/2)/528
370 LET D=D-D1
380 LET V=V+V2
390 IF D <= 0 THEN 580
400 IF U <= 0 THEN 430
410 LET T=T+10
420 GOTO 240
430 PRINT "OUT OF FUEL AT"D;"MILES UP"
440 LET J=0
450 LET J=J+1
460 LET V1=V
470 LET V2=.16*332
480 LET D=D-(V1-V2/2)/528
485 LET V=V+V2
490 IF D>0 THEN 450
500 PRINT "IMPACT IN ABOUT"J*10;"SECONDS"
580 IF V <= 30 THEN 630
590 PRINT "IMPACT"V*15/22;"MPH   YOU BLEW IT"
595 IF V<666 THEN 620
601 PRINT "THE SHIP WAS SABOTAGED"
620 STOP
630 PRINT "YOU MADE IT"
640 PRINT "TOUCH DOWN AT"V*15/22"MPH.  YOU HAVE"U;"UNITS OF EXTRA"
650 PRINT "FUEL.  BEFORE YOU REPORT TO NASA, PLEASE INFORM LARRY"
660 PRINT "SIEGEL, 3052 WARRINGTON RD., SHAKER HTS., OHIO  44120,"
665 PRINT "AS TO HOW YOU SUCCEEDED.  CONGRATULATIONS"
670 END
```

CRAZY EIGHTS

This module shows you how to write a program that simulates a 2-player card game based on a standard deck of fifty-two cards. Each player receives seven cards. The rest of the pack is placed face down to form the stack. The top card of the stack is turned face up and placed beside it to form the starter. The first player must place on the starter a card that matches it either in suit or in rank. Each in turn must thus play one card on the starter pile, matching the last played.

If unable to play in turn, a player must draw cards one by one from the top of the stack until he is able and willing to play or until he exhausts the stack. After the stack is exhausted, a hand must play in turn if it can; when it cannot, the turn passes.

All eights are wild. An eight may be played regardless of the last previous card, and regardless of whether the hand is able to play a natural card at that time. In playing an eight, the owner must name a suit, and the next card played must be of that suit (or another eight). No limitation as to rank may be made in playing an eight.

The player first to get rid of all his cards wins. If the stack is exhausted and no hand can play, the game ends as a block.

In scoring, the winner collects for all cards remaining in the other hand: 50 for each eight, 10 for each face card, 1 for each ace, and the index value for each other card. If the deal ends in a block, the player with the lowest count collects the difference of counts.

Here's a run of a program which simulates the playing of "Crazy Eights". The program is interactive, matching the computer's strategy against a human player.

```
CRAZY EIGHTS PROGRAM

WHEN ASKED WHICH CARD YOU WISH TO PLAY YOUR ANSWER
SHOULD BE TWO NUMBERS. THE FIRST NUMBER IS THE
VALUE OF THE CARD AND THE SECOND IS THE SUIT.
FOR VALUES: 2=2,3=3,4=4,5=5,6=6,7=7,8=8,9=9,
10=10,JACK=11,QUEEN=12,KING=13,AND ACE=14
FOR SUITS: CLUBS=1,DIAMONDS=2,HEARTS=3,AND SPADES=4
```

PROJECT SOLO / Dept. of Computer Science / Univ. of Pittsburgh (15213)
Module #0220 / Frank Wimberly & Carolyn Len (PS)

```
 YOUR HAND IS
4             C
7             C
5             D
3             H
9             H
5             S
Q             S

 THE LAST CARD PLAYED WAS
A             H
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 9,3

 YOUR HAND IS
4             C
7             C
5             D
3             H
5             S
Q             S

 THE LAST CARD PLAYED WAS
9             S
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 12,4

 YOUR HAND IS
4             C
7             C
5             D
3             H
5             S

 THE LAST CARD PLAYED WAS
8             C
THE SUIT CALLED FOR IS C
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 7,1

 YOUR HAND IS
4             C
5             D
3             H
5             S

 THE LAST CARD PLAYED WAS
J             C
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 4,1

 YOUR HAND IS
5             D
3             H
5             S

 THE LAST CARD PLAYED WAS
4             H
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 3,3
```

```
 YOUR HAND IS
5              D
5              S

 THE LAST CARD PLAYED WAS
6              H
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 0
YOUR NEW CARD IS 9    D
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 0
YOUR NEW CARD IS 10   H
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 10,3

 YOUR HAND IS
5              D
9              D
5              S

 THE LAST CARD PLAYED WAS
10             D
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 9,2

 YOUR HAND IS
5              D
5              S

 THE LAST CARD PLAYED WAS
2              D
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 5,2

 YOUR HAND IS
5              S

 THE LAST CARD PLAYED WAS
5              C
DO YOU WISH TO PLAY A CARD (1=YES,0=NO)? 1
WHICH OF YOUR CARDS DO YOU WISH TO PLAY? 5,4

CARDS LEFT IN MY HAND
6              C
9              C
K              C
K              H
YOUR SCORE IS  35
MY SCORE IS  0
YOU WON THAT HAND.

DO YOU WISH TO PLAY AGAIN (1=YES,0=NO)? 0

OUT OF  1  HANDS YOU WON  1 . YOUR TOTAL SCORE IS  35  ; MINE IS  0 .
LOOKS LIKE YOUR HIGH SCORER.
BYE.
```

```
10 RANDOMIZE
50 PRINT "CRAZY EIGHTS PROGRAM"
60 PRINT
100 DIM A(52),D(52),G(52),H(52)
110 DIM T(52),S(52),V(52)
120 DIM Z(52),U$(4),C$(13)
130 READ U$(1),U$(2),U$(3),U$(4)
140 FOR I=1 TO 13
150 READ C$(I)
160 NEXT I
170 FOR I=1 TO 52
180 READ Z(I)
190 NEXT I
200 DATA C,D,H,S
210 DATA "2","3","4","5","6","7","8","9","10",J,Q,K,A
220 DATA 2,3,4,5,6,7,50,9,10,10,10,10,1
230 DATA 2,3,4,5,6,7,50,9,10,10,10,10,1
240 DATA 2,3,4,5,6,7,50,9,10,10,10,10,1
250 DATA 2,3,4,5,6,7,50,9,10,10,10,10,1
300 PRINT "WHEN ASKED WHICH CARD YOU WISH TO PLAY YOUR ANSWER"
301 PRINT "SHOULD BE TWO NUMBERS. THE FIRST NUMBER IS THE "
302 PRINT "VALUE OF THE CARD AND THE SECOND IS THE SUIT."
303 PRINT "FOR VALUES: 2=2,3=3,4=4,5=5,6=6,7=7,8=8,9=9,"
304 PRINT "10=10,JACK=11,QUEEN=12,KING=13,AND ACE=14"
305 PRINT "FOR SUITS: CLUBS=1,DIAMONDS=2,HEARTS=3,AND SPADES=4"
320 LET L=0
330 FOR I=1 TO 4
340 FOR J=1 TO 13
350 LET L=L+1
360 LET S(L)=I
370 LET V(L)=J
380 NEXT J
390 NEXT I
400 LET N1=0
410 LET N2=0
420 LET N3=0
430 LET S1=0
440 LET S2=0
450 FOR I=1 TO 52
460 LET G(I)=0
470 LET H(I)=0
480 LET A(I)=RND(Z)
490 NEXT I
500 LET W1=0
510 LET B1=0
520 LET P1=1
530 LET P2=1
540 LET J=0
550 LET M1=99999
560 FOR I=1 TO 52
570 IF A(I)>=M1 THEN 600
580 LET M1=A(I)
590 LET I1=I
600 NEXT I
```

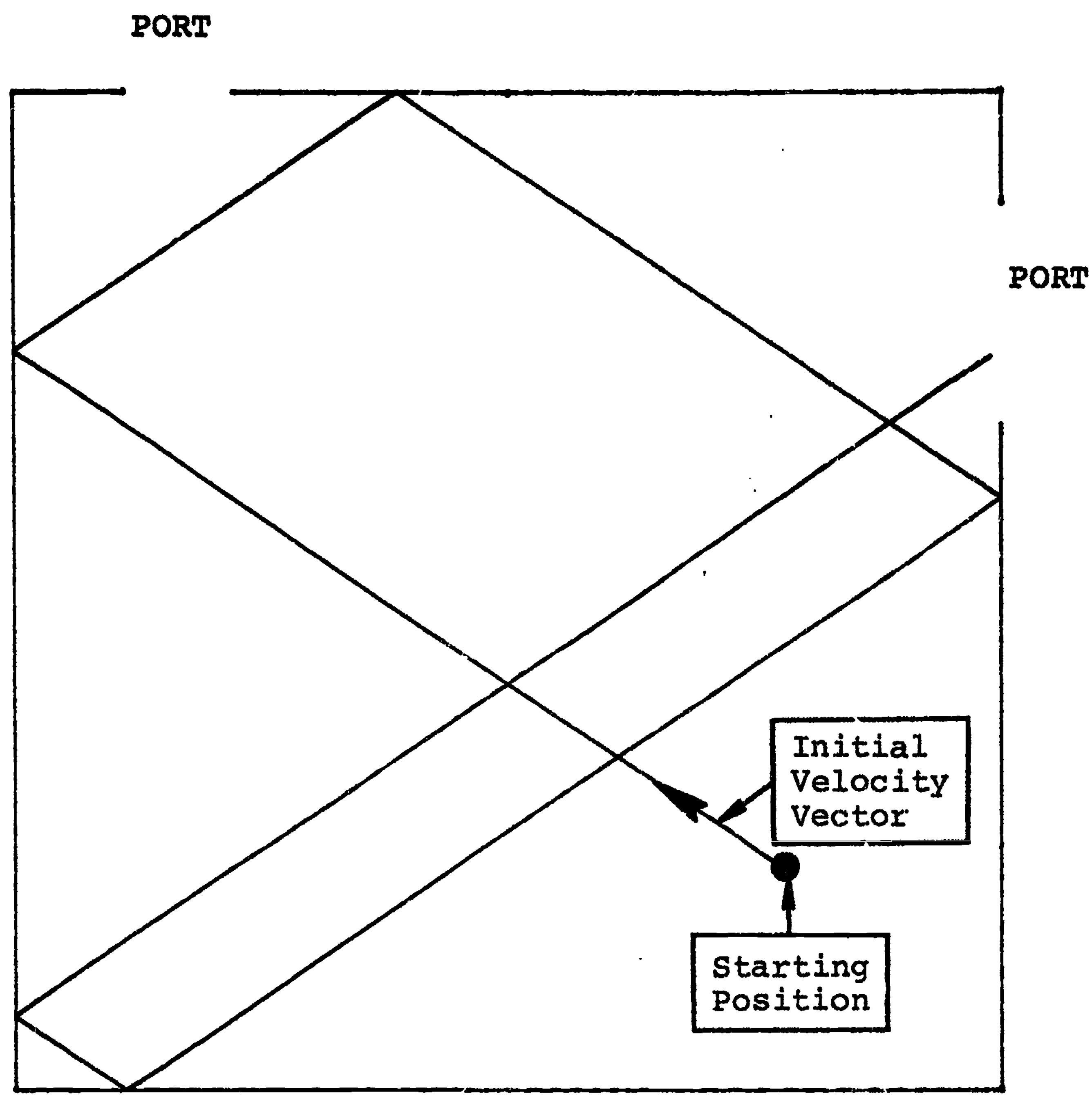
```
610 LET J=J+1
620 LET D(J)=I1
630 LET A(I1)=99999
640 IF J<52 THEN 550
650 FOR I=1 TO 7
660 LET G(D(52-2*I+2))=1
670 LET H(D(52-2*I+1))=1
680 NEXT I
690 LET T(1)=D(38)
700 LET T1=1
710 LET P=S(T(1))
720 LET H1=7
730 LET H2=7
740 LET D1=37
760 GOSUB 3000
770 GOSUB 5000
780 IF W1=1 THEN 860
790 IF B1=1 THEN 860
820 GOSUB 6000
830 IF W1=1 THEN 860
840 IF B1=1 THEN 860
850 GOTO 760
860 GOSUB 2000
870 PRINT "YOUR SCORE IS ";C1
880 PRINT "MY SCORE IS ";C2
890 IF C2>=C1 THEN 930
900 PRINT "YOU WON THAT HAND."
910 LET N1=N1+1
920 GOTO 950
930 PRINT "I WIN THAT HAND."
940 LET N2=N2+1
950 PRINT
960 LET N3=N3+1
970 PRINT "DO YOU WISH TO PLAY AGAIN (1=YES,0=NO)";
975 INPUT Z9
980 IF Z9=1 THEN 450
990 PRINT
991 PRINT "OUT OF ";N3;" HANDS YOU WON ";N1;
993 PRINT ". YOUR TOTAL SCORE IS ";S1;" ; MINE IS ";S2;"."
995 IF S1<=52 THEN 998
996 PRINT "LOOKS LIKE YOUR HIGH SCORER."
997 GOTO 999
998 PRINT "LOOKS LIKE I AM THE HIGH SCORER."
999 PRINT "BYE."
1000 STOP
2000 LET C1=0
2010 LET C2=0
2020 IF H2=0 THEN 2070
2024 PRINT
2025 PRINT "CARDS LEFT IN MY HAND"
2030 FOR I=1 TO 52
2040 IF H(I)=0 THEN 2060
2045 PRINT C$(V(I)),U$(S(I))
2050 LET C1=C1+Z(I)
2060 NEXT I
2070 IF H1=0 THEN 2140
2080 FOR I=1 TO 52
2090 IF G(I)=0 THEN 2110
2100 LET C2=C2+Z(I)
2110 NEXT I
```

```
2120 IF C1=0 THEN 2140
2122 IF C1>=C2 THEN 2130
2124 LET C2=C2-C1
2126 LET C1=0
2128 GOTO 2140
2130 LET C1=C1-C2
2132 LET C2=0
2140 LET S1=S1+C1
2150 LET S2=S2+C2
2160 RETURN
3000 PRINT
3005 PRINT " YOUR HAND IS"
3010 FOR I=1 TO 52
3020 IF G(I)=0 THEN 3040
3030 PRINT C$(V(I)),U$(S(I))
3040 NEXT I
3110 PRINT
3120 PRINT " THE LAST CARD PLAYED WAS "
3130 PRINT C$(V(T(T1))),U$(S(T(T1)))
3133 IF V(T(T1))<>7 THEN 3150
3136 PRINT "THE SUIT CALLED FOR IS ";U$(P)
3150 RETURN
5000 PRINT "DO YOU WISH TO PLAY A CARD
5010 INPUT Z9
5020 IF Z9=1 THEN 5080
5025 IF D1<>0 THEN 5030
5026 PRINT "ALL THE CARDS HAVE BEEN DEALT-
5027 GOTO 5330              --YOU ARE BLOCKED"
5030 LET I=D(D1)
5040 LET D1=D1-1
5050 LET H1=H1+1
5060 LET G(I)=1
5062 PRINT "YOUR NEW CARD IS ";C$(V(I));"   ";U$(S(I))
5070 GOTO 5000
5080 PRINT "WHICH OF YOUR CARDS DO YOU WISH TO PLAY";
5090 INPUT V9,S9
5091 LET V9=V9-1
5100 IF V9=7 THEN 5150
5110 IF S9=P THEN 5150
5120 IF V9=V(T(T1)) THEN 5150
5130 PRINT "THAT IS NOT A LEGAL PLAY."
5140 GOTO 5000
5150 FOR I=1 TO 52
5160 IF G(I)=0 THEN 5190
5170 IF V(I)<>V9 THEN 5190
5180 IF S(I)=S9 THEN 5220
5190 NEXT I
5200 PRINT "YOU DO NOT HAVE THAT CARD."
5210 GOTO 5000
5220 LET G(I)=0
5230 LET H1=H1-1
5240 LET T1=T1+1
5245 LET T(T1)=I
5250 IF V9<>7 THEN 5280
5260 PRINT "WHAT SUIT DO YOU WISH";
5270 INPUT S9
5280 LET P=S9
5290 LET P1=1
5300 IF H1<>0 THEN 5320
5310 LET W1=1
5320 RETURN
5330 IF P2<>0 THEN 5350
5340 LET B1=1
5350 LET P1=0
5360 RETURN
6000 LET I=53
6010 LET I=I-1
6020 IF H(I)=0 THEN 6060
6030 IF V(I)=7 THEN 6060
6040 IF S(I)=P THEN 6160
6050 IF V(I)=V(T(T1)) THEN 6160
6060 IF I>1 THEN 6010
6061 FOR I=1 TO 52
6062 IF V(I)<>7 THEN 6064
6063 IF H(I)<>0 THEN 6160
6064 NEXT I
6070 IF D1=0 THEN 6370
6080 LET I=D(D1)
6090 LET D1=D1-1
6100 IF V(I)=7 THEN 6180
6110 IF S(I)=P THEN 6180
6120 IF V(I)=V(T(T1)) THEN 6180
6130 LET H2=H2+1
6140 LET H(I)=1
6150 GOTO 6070
6160 LET H(I)=0
6170 LET H2=H2-1
6180 LET T1=T1+1
6190 LET T(T1)=I
6195 LET P=S(I)
6200 IF V(I)<>7 THEN 6330
6210 FOR J=1 TO 4
6220 LET Y(J)=0
6230 LET K1=13*(J-1)+1
6235 LET K2=13*J
6240 FOR K=K1 TO K2
6245 IF H(I)=0 THEN 6260
6250 LET Y(J)=Y(J)+1
6260 NEXT K
6270 NEXT J
6280 LET P=1
6290 FOR J=2 TO 4
6300 IF Y(J)<=Y(P) THEN 6320
6310 LET P=J
6320 NEXT J
6330 LET P2=1
6340 IF H2<>0 THEN 6360
6350 LET W1=1
6360 RETURN
6370 IF P1<>0 THEN 6390
6380 LET B1=1
6390 LET P2=0
6400 RETURN
6410 END
```

SIMULATING MOTION ON A

Rectangular Billiard Table

- The program described in this module simulates the "dynamics" of a particle acting under inelastic collision. We can think of the particle as a billiard ball on an ideal table, i.e. a table on which no loss of energy takes place.
- Physics students can give other interpretations to the mathematical model which underlies the program.
- Information on how to use an X-Y plotter to display the output of this program can be obtained from module #0131.



PROJECT SOLO, Computer Science Department, University of Pittsburgh (15213)
Module #9007, Gary Lindstrom and Carolyn Len (PS)

In the billard simulation, we are using an elementary law of physics. When an inelastic particle hits an immovable wall it bounces off in a special way.

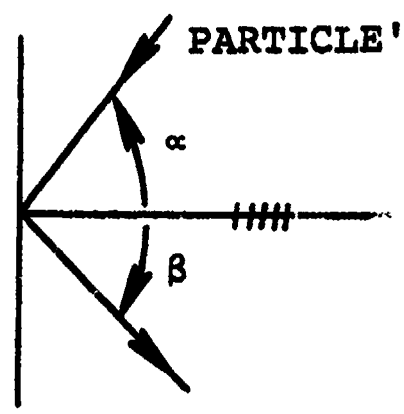


Figure 1.

In Figure 1 the straight lines with arrowheads represent the path of a particle hitting a wall. The line with the slashes is a perpendicular to the wall at the point where the particle hit the wall. Angle α is called the angle of incidence (i.e. the angle between the perpendicular and the particle's incoming path). Angle β is called the angle of reflection (i.e. the angle between the perpendicular and the particle's outgoing path.) An elementary physics law (the law of reflection) states that the angle of incidence (angle α) and the angle of reflection (angle β) are equal. Further, if the particle is inelastic and the wall is unyielding the speed of the particle will remain constant.

Here is another way of looking at this collision.

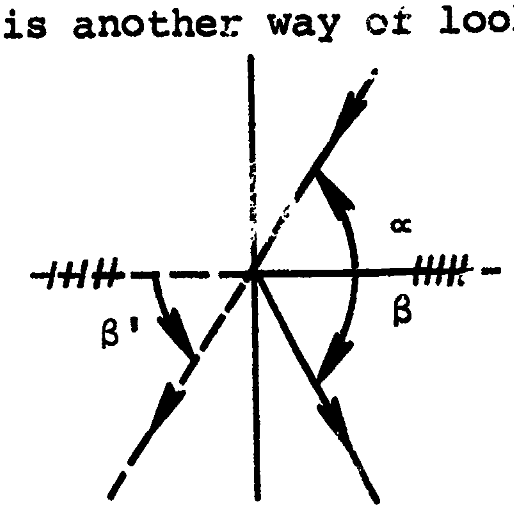


Figure 2.

In Figure 2 an additional path is dotted in. This path is called a virtual path (i.e. an imaginary path). Also the perpendicular has been extended into the wall. The virtual path is the path the particle would have followed if the wall did not exist. Notice that angle β' is equal to angle α (Can you prove this?) and therefore equal to angle β. The virtual path is the particle's trajectory in the mirror world of the particle's actual path in the real world.

Now imagine there are two walls in our real world.

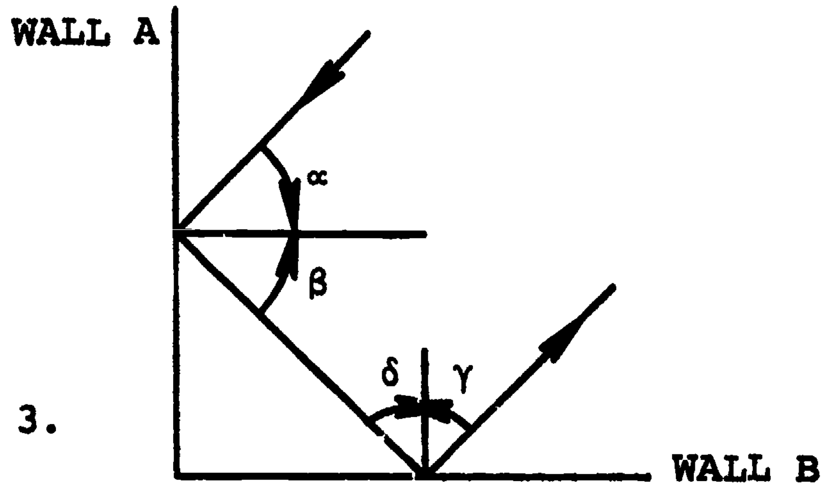


Figure 3.

This time the particle hits wall A bounces off, and hits wall B. When it hits wall A the angle of incidence is α and the angle of reflection is β. When it hits wall B the angle of incidence is δ and the angle of reflection is γ. What can be said of α, β, δ, and γ? From the law of reflection we know that $<\alpha = <\beta$ and $<\delta = <\gamma$.

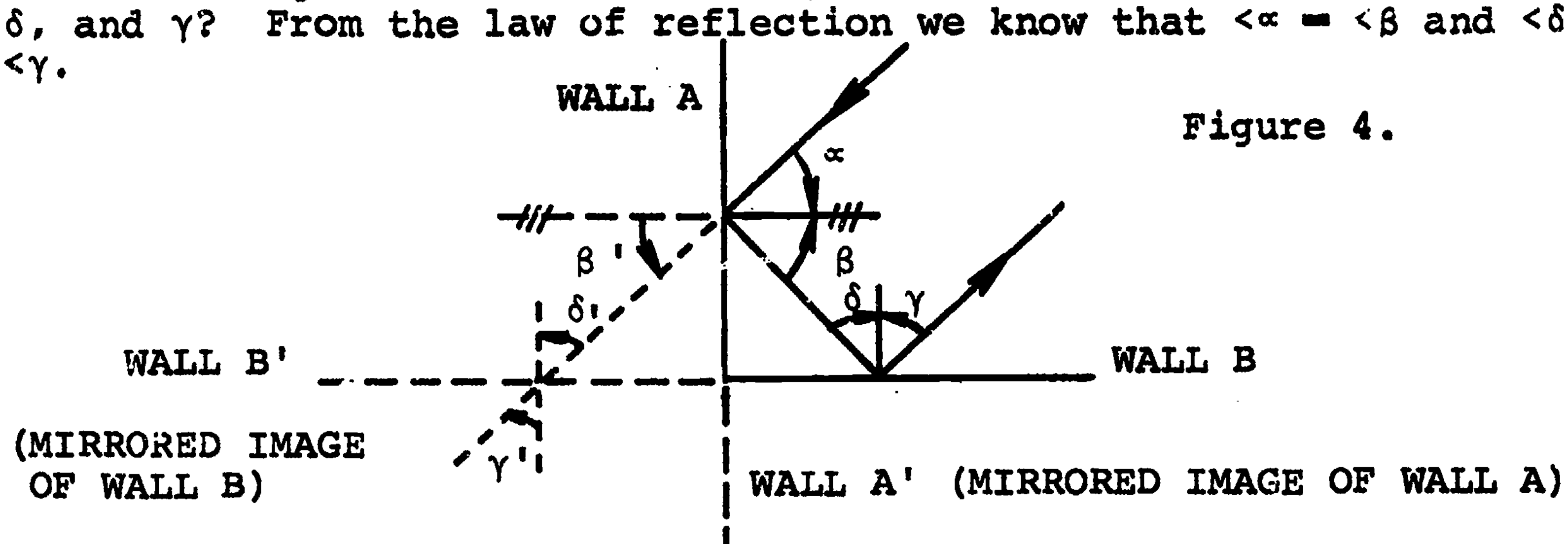


Figure 4.

In Figure 4 we drew in the virtual path of the particle's trajectory in the mirror world. We already know that $<\beta = <\beta'$, but can you prove the $<\delta = <\delta'$? You should be able to using plane geometry. Also can you show $<\gamma = <\gamma'$? $<\gamma'$ lies in a different mirror imaged world than either $<\alpha$ or $<\beta'$. Can you see how this mirror image was produced?

If you stretch your imagination you can see that this virtual path method can be extended to as many bounces off as many walls as you like. In Figure 5 we used four walls and extended the reflections (mirrored images) to fill a plane. This idea of reflected images is the method used by the billard simulation program to calculate the particle's movement.

Figure 5 is a sketch of the plane of reflected tables for the sample run of the billard simulation program. The little boxes on the table's sides are the table's pockets. The solid line extending through the different reflected tables is the particle's virtual path. The broken line in the box on the lower left is the real path of the particle on the billard table.

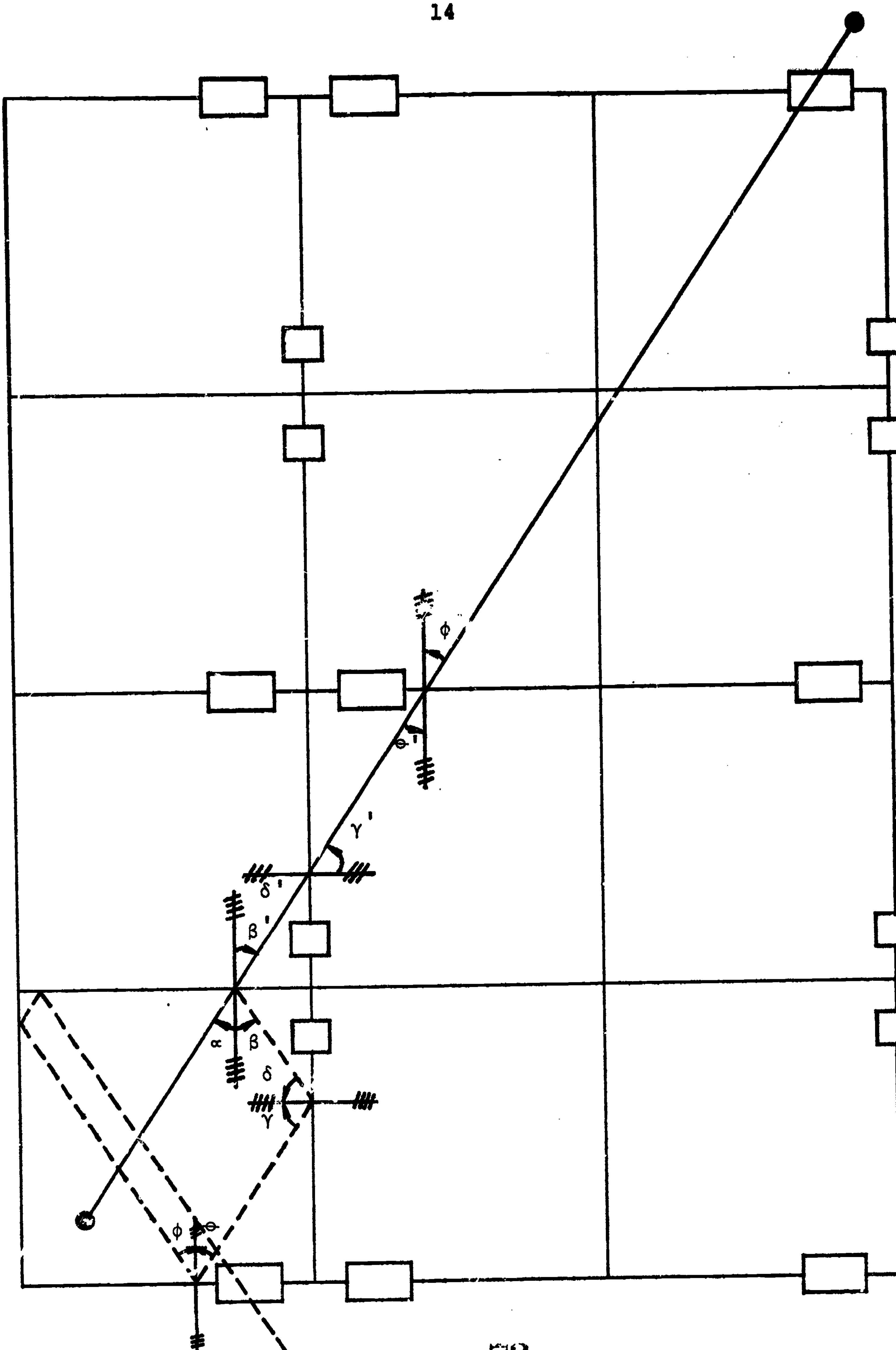


Figure 5

There are three major parts to the Billard Simulation program.

1. The program prompts you to give the following information:

    a. the length and width of the rectangular billard table
    b. the position and size of pockets (these are openings on the sides of the table through which the particle can exit)
    c. the starting position of the particle on the table
    d. the initial x and y components of the velocity vector for the particle's movement

    The table and pockets are then drawn by the plotter and the plotter pen is positioned at the particle's starting location.

2. The program controls the simulation by:

    a. finding the side of the table the particle will hit next
    b. computing the impact coordinates of the particle
    c. plotting in line mode the calculated path of the particle's movement
    d. checking to see if the particle exited a pocket
        (1) if it did not exit go to step 2a.
        (2) if it did exit go to the third part of the program

3. The program types a summary of what happened during the simulation.

    a. the number of the pocket the particle exited
    b. the number of times the particle bounced off of a side of the table before exiting
    c. the exit coordinates of the particle
    d. the distance the particle traveled on the table
    e. the time units elapsed between the time of the particle's initial velocity on the table and the time of its exit into a pocket.

    3d and 3e assume the initial velocity is per unit time.

    The person running the program is then given the choice of starting another particle on the same table or ending the program.

An Explanation of the Program Parts

Here is a description of what statements 100 to 260 do. These lines set up or initialize the particular simulation to be done.

100	All variables are set to zero.
105	Dimensions are set for pockets in the table--a maximun of 10 pockets is allowed.
110	I, N, and B are made integer variables. I, N, and B are going to be used for counters within the program.
115 - 117	The forms are set for use in the print statements.
120 - 136	The length and width of the billard table are set.
137	A scale factor for the plotting statements is determined.
140 -215	These statements ask the user to describe the positions and sizes of the pockets; they also check to make sure that no "impossible" pockets are requested.
218	Subroutine 600 is called. It draws the billard table on the plotter.
220	The bounce counter is set to zero.
230 - 250	These statements ask the user to state the initial position and components of the velocity vector for the particle; they also check that the starting position is on the table and that the particle is not stationary.
258	The plotter is put in line mode.
260	The pen is positioned at the particle's starting location on the table (the x component is scaled to two-thirds of the y component) (both x and y are scaled for coordinate system 0 to 9999) (both x and y are scaled for a non-square table)

Variables introduced in lines 100-260.

P(N,1) is the opening x coordinate of pocket N.
Q(N,1) is the opening y coordinate of pocket N.
P(N,2) is the closing x coordinate of pocket N.
Q(N,2) is the closing y coordinate of pocket N.
A$,B$, and C$ are string variables containing format definitions.
R is the length of the rectangular table.
T is the width of the rectangular table.
A is used to scale the coordinates for plotting.
N is the number of pockets in the table.
B is the counter for the number of times a particle bounces off of a side before it exits a pocket.
X is the x coordinate of the particle's starting position.
Y is the y coordinate of the particle's starting position.
S1 is the x component of the velocity vector of the particle.
S2 is the y component of the velocity vector of the particle.

An Explanation of the Calculations for the Simulation of the Particle's Movement

Lines 265 to 285 control the calling of subroutines to calculate the particle's movement and plot this movement.

Subroutine 300 finds the next side of the table that is hit. We envision the table as being reflected until it covers the plane (See Figure 5). In this view the particle's trajectory is then a ray starting at an initial point and extending in one direction with a given velocity. The x and y coordinates for the particle are given values in this plane extension (See flowchart Figure 6). Then subroutine 400 translates each coordinate pair produced into cooresponding coordinate values in the original table. These values (M1 and M2) are used in subroutine 500 to check for pocket encounters or exits.

Relevant variables in subroutine 300

D2 is the distance from the particle's present position on the virtual path on the reflected tables to the next vertical side encountered on this path.
D1 is the distance from the particle's present position on the virtual path on the reflected tables to the next horizontal side encountered on this path.

H1/H2 are the x and y coordinates of the next horizontal side encountered by the particle on its virtual path on the reflected tables. This assumes that intermediate vertical side encounters are ignored.
V1/V2 correspond to H1 and H2 for the next vertical side encountered by the particle.

Subroutine 400 translates the x and y coordinates of the next side encountered on the particle's virtual path on the reflected tables to their corresponding x and y coordinates on the original table.

Relevant variables in subroutine 400

C1 is the number of complete table widths between the original table and the x coordinate of the next side encountered on the particle's virtual path on the reflected tables. The sign indicates right (+) or left (-) of the particle's starting position.
C2 is similar to C1 for the y coordinate. The sign indicates up (+) or down (-).

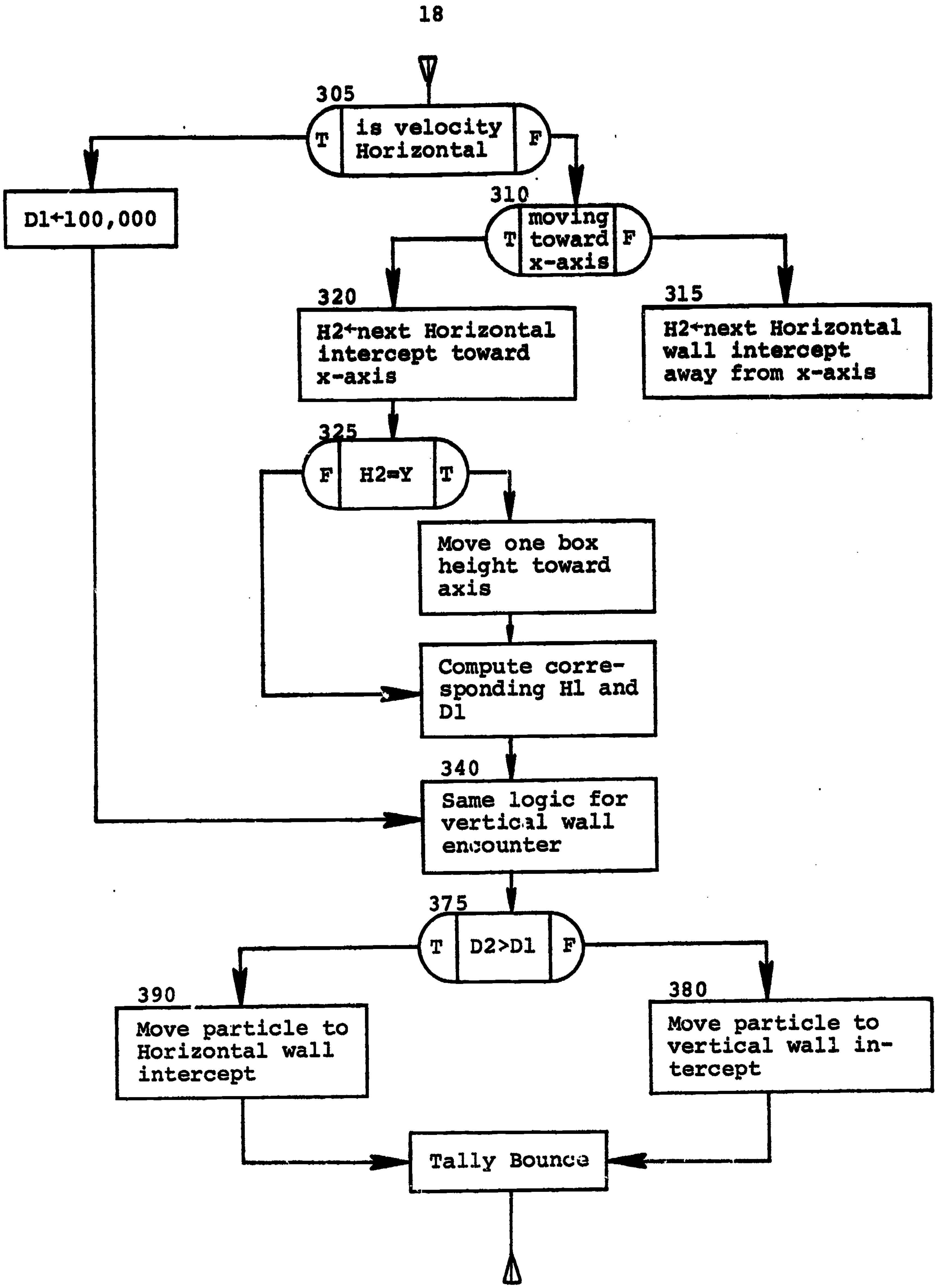

18
305
T
is velocity Horizontal
F
D1←100,000
310
T
moving toward x-axis
F
320
H2←next Horizontal intercept toward x-axis
315
H2←next Horizontal wall intercept away from x-axis
325
F
H2=Y
T
Move one box height toward axis
Compute corresponding H1 and D1
340
Same logic for vertical wall encounter
375
T
D2>D1
F
390
Move particle to Horizontal wall intercept
380
Move particle to vertical wall intercept
Tally Bounce

Figure 6

M1 is the displacement from the vertical boundary named in C1 to the x coordinate of the next side encountered on the virtual path.
M2 is similar to M1 for C2 and the y coordinate.

We convert M1 and M2 to be the coordinates normalized to the original table. There are four cases for M1 conversions; one for each different reflection of the original table the particle may encounter.

III IV III | I II I II

y-axis

I = M1 is unchanged. The reflected image is the same as the original.

II = M1 is changed to the width of the table minus M1. The reflected image is like the one to the right of the original.

III = M1 is negated. The x coordinate is moved to the positive side of the y axis. The reflected image is like the one on top of and to the right of the original.

IV = M1 is changed to the width of the table plus M1. The reflected image is like the one on top of the original.

Similar conversions are done for M2.

Subroutine 500 checks to see if the normalized x and y coordinates of the particle are included in the pocket for which it was called. If it is not included the subroutine returns. If it is included a summary of information is printed about the particle's movement.

Relevant variables

D is the distance traveled by the particle on the table.
T1 is the number of time units elapsed.

Here is a description of the statements that draw the table and its pockets. (lines 600-845)

600	Dimensions are set for the maximum number of pockets allowed on one side of the table.
605	X9, Y9, J, and J1 are made integer variables.
610	The plotter is set to line mode.

615	Y0 is set to find pockets along the side of the table where the y coordinate is zero.
620	A counter is initialized to zero.
625 - 635	All the pockets along the given y coordinate are found and put in the B matrix. J is the number of pockets found.
640	Subroutine 800 is called to sort the pockets in the B matrix. This is done in order to plot the side.
645	Y9 is set to the scaled y coordinate to be used in the plot statements.
650	The plotter pen is positioned to begin plotting.
655 - 675	These lines plot the horizontal sides of the table with spaces where the pockets are located.
680	If only one horizontal line is plotted, go back and plot the second horizontal line.
715 - 780	These lines are the same as lines 615 - 680 except for the vertical sides.
785	The plot line mode is ended.
790	This returns to the statement that called subroutine 600.
800 - 845	These are the statements that sort the pockets in the B matrix into ascending order before plotting.

A listing of the program /BILLARD/.

```
100 VAR=ZERO
105 DIM P(10,2),Q(10,2)
110 INTEGER I,N,B
115 A$="'PARTICLE EXITS POCKET ' ZD ' AFTER ' ZZZD' BOUNCES,'/"
116 B$="'COORDINATES (' ZZZD.DDD ',' ZZZD.DDD'),'/"
117 C$="'DISTANCE ' ZZZZZZZD.DD ', TIME ' ZZZD.DD'.'/"
120 PR. "PROGRAM TO SIMULATE PARTICLE DYNAMICS "
125 PR. "IN A POCKETED RECTANGLE."
130 PR. PR. "INPUT THE LENGTH AND WIDTH OF YOUR RECTANGLE":
135 INPUT R,T
136 IF R<0 OR T<0 PR. "LENGTH AND WIDTH MUST BE GREATER THAN ZERO."
GOTO 130
137 LET A=MAX(R,T)
140 PR. PR. "ENTER POCKET PAIRS."
145 PR."ENTER A NUMBER LESS THAN ZERO WHEN YOU WANT NO MORE POCKETS."
150 N=0
160 PR. "OPEN X COORDINATE=":
165 INPUT P(N+1,1)
170 IF P(N+1,1)<0 GOTO 218
175 PR. "OPEN Y COORDINATE=":
180 INPUT Q(N+1,1)
181 IF P(N+1,1)>R OR Q(N+1,1)>T OR Q(N+1,1)<0 GOTO 190
183 IF P(N+1,1)<>0 AND P(N+1,1)<>R GOTO 193
185 IF Q(N+1,1)<>P(N+1,1) LET P(N+1,2)=P(N+1,1) GOTO 187
186 PR. "CLOSE X COORDINATE =": INPUT P(N+1,2)
187 PR. "CLOSE Y COORDINATE =": INPUT Q(N+1,2)
188 IF P(N+1,2)>R OR P(N+1,2)<0  OR Q(N+1,2)>T OR Q(N+1,2)<0 GOTO 190
189 IF P(N+1,1)=P(N+1,2) OR Q(N+1,1)=Q(N+1,2) GOTO 205
190 PR. "ILLEGAL POCKET." GOTO 160
193 IF Q(N+1,1)<>0 AND Q(N+1,1)<>T GOTO 190
195 Q(N+1,2)=Q(N+1,1) PR. "CLOSE X COORDINATE=":
200 INPUT P(N+1,2)
202 IF P(N+1,2)>Z OR P(N+1,2)<0 GOTO 190
205 IF P(N+1,1)<=P(N+1,2) AND Q(N+1,1)<=Q(N+1,2) GOTO 215
208 LET E=P(N+1,1) LET P(N+1,1)=P(N+1,2) LET P(N+1,2)=E
210 LET E=Q(N+1,1) LET Q(N+1,1)=Q(N+1,2) LET Q(N+1,2)=E
215 N=N+1 GOTO 160
218 GOSUB 600
220 B=0
230 PR.PR."TYPE THE X,Y CORDINATES OF THE PARTICLES INITIAL POSITION.":
235 INPUT X,Y
238 IF X>R OR X<0 OR Y>T OR Y<0 PR. "IMPROPER STARTING POSITION "
GOTO 230
240 PR. "TYPE THE X AND Y COMPONENTS OF THE
VELOCITY VECTOR FOR THE PARTICLE.":
245 INPUT S1,S2
247 IF S1=0 AND S2=0 PR. "PARTICLE IS STATIONARY." GOTO 240
250 LET X1=X LET Y1=Y
258 PR. "PLTL"
260 PRINT INT(2./3.*(X/A*9999)), INT(Y/A*9999)
265 GOSUB 300
270 GOSUB 400
275 PRINT INT(2./3.*(M1/A*9999)), INT(M2/A*9999)
280 GOSUB 500 FOR I=1 TO N
285 GOTO 265
```

A listing of the program /BILLARD/ continued.

```
300 REM FIND THE NEXT WALL
305 IF S2=0 LET D1=100000 GOTO 340
310 IF S2*Y<0 GOTO 320
315 H2=(INT(Y/T)+ABS(S2)/S2)*T GOTO 330
320 H2=INT(Y/T)*T
325 IF H2=Y LET H2=H2+ABS(S2)/S2*T
330 H1=X+(H2-Y)*S1/S2
335 D1=(H2-Y)*(H2-Y)+(H1-X)*(H1-X)
340 IF S1=0 LET D2=100000 GOTO 375
345 IF S1*X<0 GOTO 355
350 V1=(INT(X/R)+ABS(S1)/S1)*R GOTO 365
355 V1=INT(X/R)*R
360 IF V1=X LET V1=V1+ABS(S1)/S1*R
365 V2=Y+(V1-X)*S2/S1
370 D2=(V1-X)*(V1-X)+(V2-Y)*(V2-Y)
375 IF D2>D1 GOTO 390
380 LET X=V1 LET Y=V2
385 LET B=B+1 RETURN
390 LET X=H1 LET Y=H2
395 LET B=B+1 RETURN
400 REM NORMALIZE PARTICLE COORDINATES
405 LET C1=INT(X/R) LET C2=INT(Y/T)
410 LET M1=X-C1*R LET M2=Y-C2*T
415 F=INT(C1/2)
420 IF X<0 GOTO 435
425 IF C1/2-F=0 GOTO 445
430 M1=R-M1 GOTO 445
435 IF C1/2-F=0 LET M1=-M1 GOTO 445
440 M1=R+M1
445 F=INT(C2/2)
450 IF Y<0 GOTO 465
455 IF C2/2-F=0 RETURN
460 M2=T-M2 RETURN
465 IF C2/2-F=0 LET M2=-M2 RETURN
470 M2=T+M2 RETURN
500 REM CHECK PARTICLE AT WALL FOR POCKET ENTRY
505 IF P(I,1)=P(I,2) GOTO 525
510 IF M1<P(I,1) RETURN
515 IF M1>P(I,2) RETURN
520 GOTO 530
525 IF M1<>P(I,1) RETURN
530 IF Q(I,1)=Q(I,2) GOTO 550
535 IF M2<Q(I,1) RETURN
540 IF M2>Q(I,2) RETURN
545 GOTO 555
550 IF M2<>Q(I,1) RETURN
555 D=SQRT((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1))
560 T1=D/SQRT(S1*S1+S2*S2)
565 PRINT "PLTT"
```

A listing of the program /BILLARD/ continued.

```
570 PR.
575 PRINT IN FORM A$:I,B-1
576 PRINT IN FORM B$:M1,M2
577 PRINT IN FORM C$:D,T1
585 PR. PR. "TYPE YES FOR MORE PARTICLES ON THIS TABLE ":
590 INPUT S$
595 IF S$="YES" GOTO 220
596 END
600 DIM B(5,2)
605 INTEGER X9,Y9,J
610 PR. "PLTL"
615 Y0=0
620 J=0
625 FOR I=1 TO N
630 IF Q(I,1)=Y0 LET J=J+1 LET B(J,1)=P(I,1) LET B(J,2)=P(I,2)
635 NEXT I
640 GOSUB 800
645 Y9=Y0/A*9999
650 PR. 0,Y9:"↑"
655 FOR I=1 TO J
660 PR. INT(2./3.*(B(I,1)/A*9999)),Y9
665 PR. INT(2./3.*(B(I,2)/A*9999)),Y9:"↑"
670 NEXT I
675 PR. INT(2./3.*(R/A*9999)),Y9
680 IF Y0=0 LET Y0=T GOTO 620
715 X0=0
720 J=0
725 FOR I=1 TO N
730 IF P(I,1)=X0 LET J=J+1 LET B(J,1)=Q(I,1) LET B(J,2)=Q(I,2)
735 NEXT I
740 GOSUB 800
745 X9=2./3.*(X0/A*9999)
750 PR. X9,0:"↑"
755 FOR I=1 TO J
760 PR. X9,INT(B(I,1)/A*9999)
765 PR. X9,INT(B(I,2)/A*9999):"↑"
770 NEXT I
775 PR. X9,INT(T/A*9999)
780 IF X0=0 LET X0=R GOTO 720
785 PR. "PLTT"
790 RETURN
800 J1=0
810 FOR I=1 TO J-1
815 IF B(I,1)<=B(I+1,1) GOTO 835
820 LET E=B(I,1) LET B(I,1)=B(I+1,1) LET B(I+1,1)=E
825 LET E=B(I,2) LET B(I,2)=B(I+1,2) LET B(I+1,2)=E
830 J1=1
835 NEXT I
840 IF J1=1 GOTO 800
845 RETURN
```

A sample run of the program /BILLARD/.

```
>RUN
PROGRAM TO SIMULATE PARTICLE DYNAMICS
IN A POCKETED RECTANGLE.

INPUT THE LENGTH AND WIDTH OF YOUR RECTANGLE?9,9

ENTER POCKET PAIRS.
ENTER A NUMBER LESS THAN ZERO WHEN YOU WANT NO MORE POCKETS.
OPEN X COORDINATE=?6
OPEN Y COORDINATE=?0
CLOSE X COORDINATE=?8
OPEN X COORDINATE=?9
OPEN Y COORDINATE=?7
CLOSE Y COORDINATE =?8
OPEN X COORDINATE=?-1
PLTL

TYPE THE X,Y CORDINATES OF THE PARTICLES INITIAL POSITION.?2,2
TYPE THE X AND Y COMPONENTS OF THE
VELOCITY VECTOR FOR THE PARTICLE.?2,3
PLTL

PARTICLE EXITS POCKET  1 AFTER   5 BOUNCES,
COORDINATES (   6.667,    0.000),
DISTANCE        40.86, TIME    11.33.

TYPE YES FOR MORE PARTICLES ON THIS TABLE ?NO
```

The plotter drawing of the billard table and the particle's movement in the above run is shown on the title page of this module.

Challenges

- Write a program that simulates inelastic collision on an elliptical billard table.

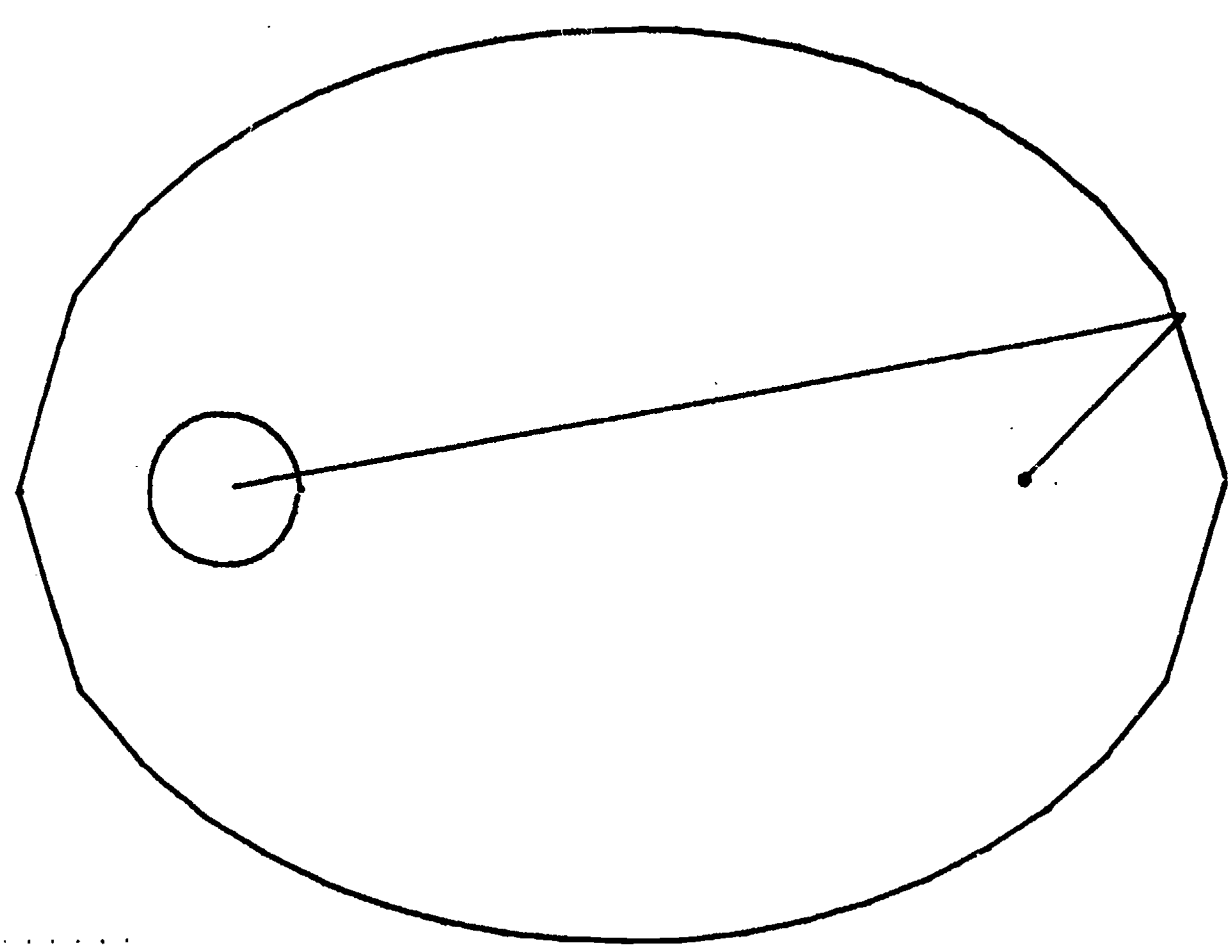
- If you have an x-y plotter, use your program to produce graphical output of the kind shown on the cover. See module #0131 for some help.

SIMULATING MOTION ON AN

ELLIPTICAL Billiard Table

- The program Elliptical Billiards simulates the motion of a ball on an "ideal" billiard table built in the shape of an ellipse. The table can have any number of holes, and each hole can have a different radius. The program assumes that a single ball has been placed at any position within the ellipse, and then given an initial velocity. The magnitude and direction of this velocity are specified by the person running the program.

- As in most simulations, the physical situation has been "idealized" in a number of ways. First, we assume that the table is frictionless, that is, that the ball's energy (and therefore its speed) does not change with time. The ball is assumed to be an infinitely small inelastic sphere. The sides of the elliptical table are also assumed to be inelastic.



PROJECT SOLO / Dept. of Computer Science / Univ. of Pittsburgh (15213)
Module #0113 / Gary Lindstrom & Carolyn Len (PS)

The elliptical billiard simulation is based on the law that the angle of incidence equals the angle of reflection. For the simulation the user specifies the size of an elliptical table, the number of holes in the table (at least one), and the position and size of each hole. After the table is specified the user positions a particle on the table and gives the components of the particle's initial velocity vector. The particle's movement on the table is then simulated until the particle falls through one of the holes in the table. The particle is idealized such that it is thought of as a point with no dimensions. This is done so that when the particle hits the side of the ellipse it hits at only one point on the ellipse. Also the particle is inelastic and the ellipse's sides are unyielding so that the speed of the particle remains constant.

The data needed for the simulation is requested and verified in lines 110 to 205.

A is the x-intercept of the ellipse; verified that A>0.
B is the y-intercept of the ellipse; verified that B>0.
{X(I), Y(I)} are the x and y coordinates for the center of the Ith hole; verified that $\left(\frac{X(I)}{A}\right)^2 + \left(\frac{Y(I)}{B}\right)^2 \leq 1$.
R(I) is the radius of the Ith hole; verified that R>0.
{U,V} are the x and y coordinates of the particle's starting position; verified that $\left(\frac{U}{A}\right)^2 + \left(\frac{V}{B}\right)^2 \leq 1$.
{V1,V2} are the x and y components of the initial velocity vector; verified that $|V1| + |V2| > 0$.

The geometric interpretation of the above is:

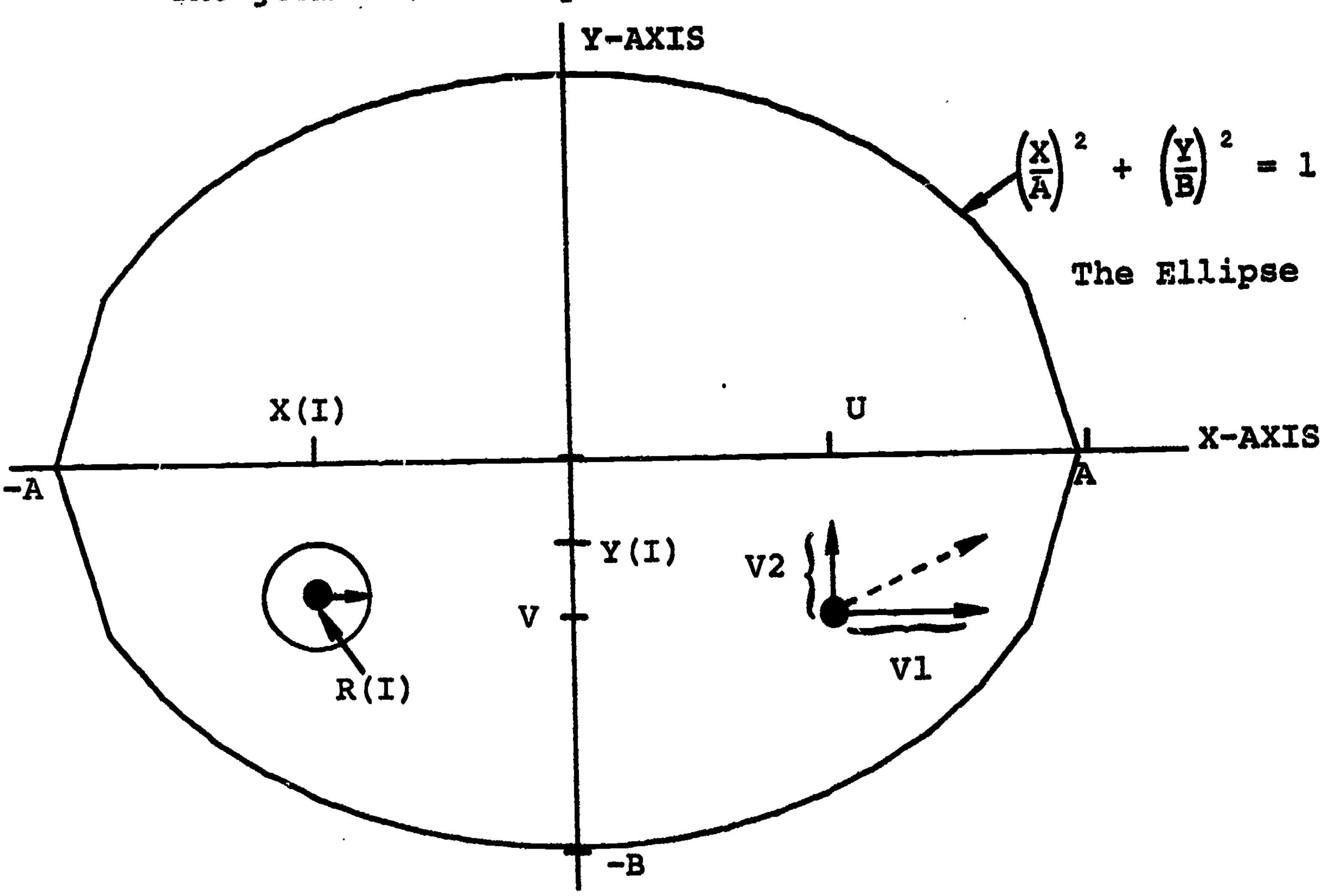


Figure 1

After the input of the data and the initialization of necessary variables the following steps are performed to calculate and plot the particle's movement.

215-220 These lines calculate the equation of the particle's current trajectory line.

The trajectory is calculated by using plane geometry. It is the line determined by two points. In this case, one point is the particle's present position (U, V) and the other is the particle's position after the components of the velocity vector have been added (U+V1, V+V2). The formula for two points determining a line is:

$$(y_2 - y_1)x - (x_2 - x_1)y + x_2y_1 - x_1y_2 = 0$$

In our problem x_1 = U+V1 $\quad y_1$ = V+V2

x_2 = U $\quad y_2$ = V

$\therefore$ X*(V -(V+V2)) - Y(U-(U+V1)) + (U*(V+V2) - (U+V1)*V) = 0

$\underbrace{\text{-V2}}_{\text{A1}}\text{*X} + \underbrace{\text{V1}}_{\text{B1}}\text{*Y} + \underbrace{\text{(U*V2 - V*V1)}}_{\text{C1}} = 0$ is the current trajectory line.

D1 is set to $\sqrt{A1^2+B1^2}$ to be used later in the program.

223 The particle's position is plotted.
225 Subroutine 300 is called to find the point where the particle will next hit the side of the ellipse.
230 Subroutine 400 is called to see if the particle passed over a hole and fell off of the table.
235 If the particle passed out of a hole and the user wants another particle on the same table this statement will reinitializes the bounce counter and go back to start another simulation.
240 If the holes are missed subroutine 500 is called to calculate the components of the new velocity vector.
245 The bounce counter is increased by one.
255 The x and y coordinate variables to be used in calculating the particle's movement are set to the particle's present position.
260 The cycle of calculating the particle's movement is restarted.

Subroutine 300 finds the next point on the table where the particle will hit. We seek the mutual solution to the equation for the ellipse and the equation of the current trajectory line. Variables P and Q will hold the x and y coordinates of the impact point.

These are calculated as follows:

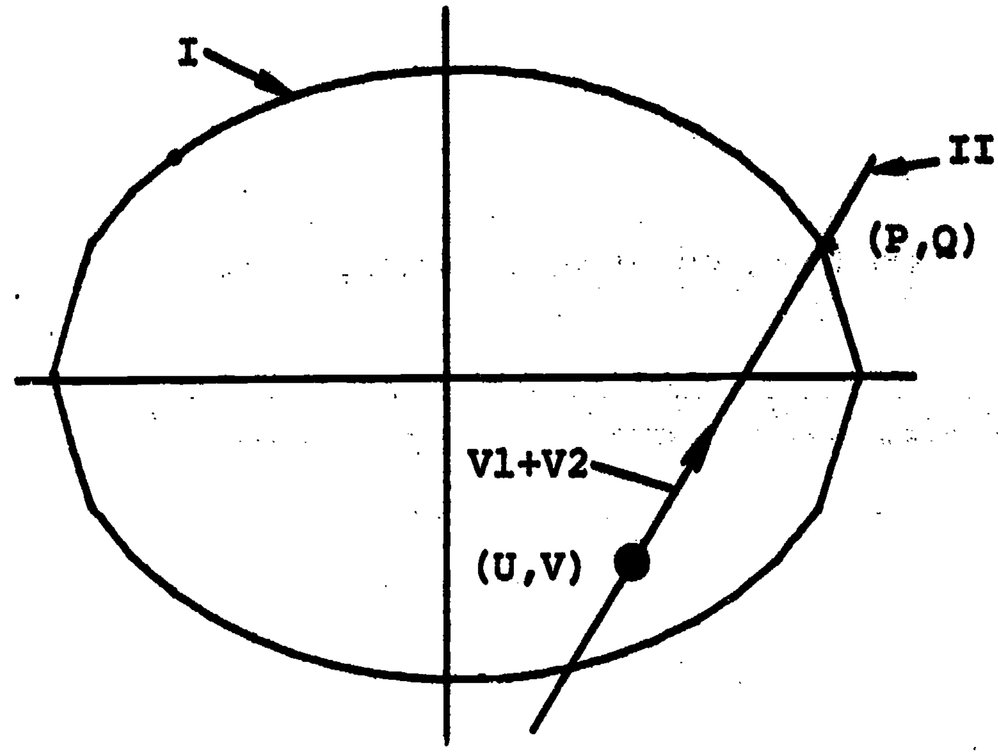
I: $\left(\frac{X}{A}\right)^2 + \left(\frac{Y}{B}\right)^2 = 1$ is the equation for the ellipse.

II: A1*X + B1*Y + C1 = 0 is the equation for the current trajectory line.

From I

$$Y = \pm B\sqrt{1 - \left(\frac{X}{A}\right)^2}$$

$$\therefore A1*X \pm B1*B\sqrt{1-\left(\frac{X}{A}\right)^2} + C1 = 0$$



$$(A1*X + C1)^2 = B1^2*B^2*(1-(X/A)^2)$$

$$A1^2*X^2 + 2*A1*C1*X + C1^2 = B1^2*B^2*(1-(X/A)^2)$$

$$\underbrace{\left(A1^2 + \frac{B1^2*B^2}{A^2}\right)}_{A2}*X^2 + \underbrace{(2*A1*C1)}_{B2}*X + \underbrace{(C1^2-B1^2*B^2)}_{G2} = 0$$

Using the quadratic formula:

$$P = \frac{-B2 \pm\sqrt{B2^2-4*A2*G2}}{2*A2}$$

A2 = 0 is not possible, since this would mean that II is tangent to I, an impossible trajectory. The square root term must be real; otherwise I and II would not meet.

We try the plus case of the plus-minus choice first. A test is made in line 330 to see if P is in the same direction from U as V1. If P is not in the same direction we have the wrong case and the minus choice is needed.

This is computed by:

V1*(P-U)>0 implies that + is the correct case.

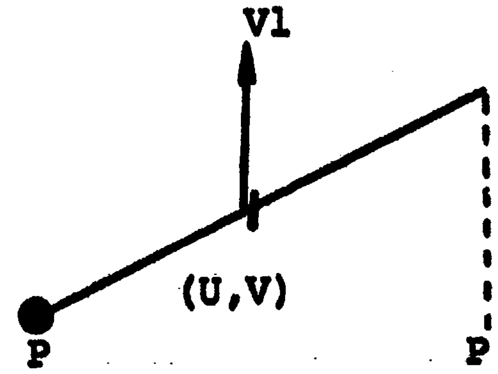


Figure 3

NOTE: A vertical vector, V1 = 0, requires a special case:

$$P = U$$
$$Q = B\sqrt{1-(U/A)^2} \quad \text{if } V2>0$$
$$\quad -B\sqrt{1-(U/A)^2} \quad \text{otherwise}$$

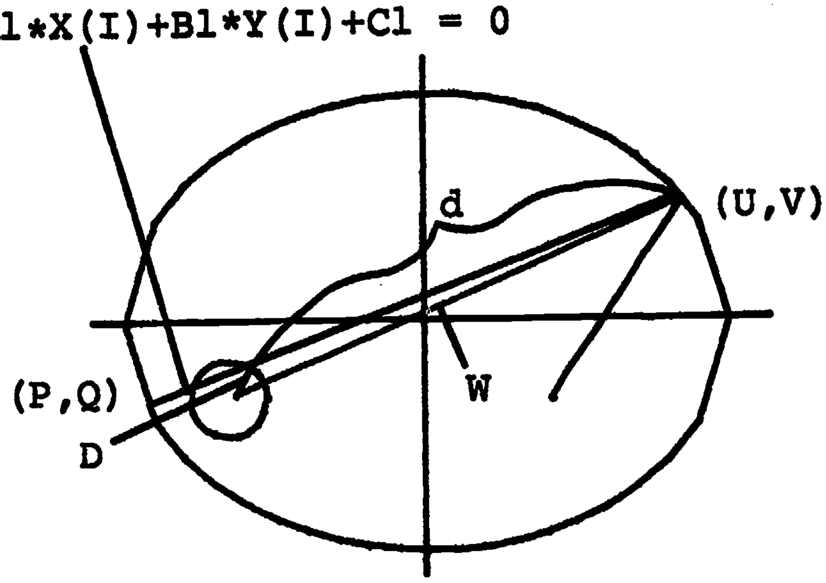
For V1≠0, Q is found by solving II.

As stated before, P and Q are the x and y coordinates of the point where the particle hit the side of the ellipse.

Subroutine 400 checks to see if the line of trajectory passes over the Ith hole. This is done by using the formula for the distance between a line and a point. If the distance between the trajectory line and the center of the hole is less than the radius of that hole the particle will fall off the table through that hole.

That is: $D = \dfrac{|A1*X(I) + B1*Y(I) = C1|}{\sqrt{A1^2 + B1^2}}$ If $D \leq R(I)$, ball passes over hole I.

A1*X(I)+B1*Y(I)+C1 = 0



D is taken to be the dis- traveled on the last bounce.

$$D = \sqrt{W^2 - D^2}$$

When the particle falls through a hole the following information is given:

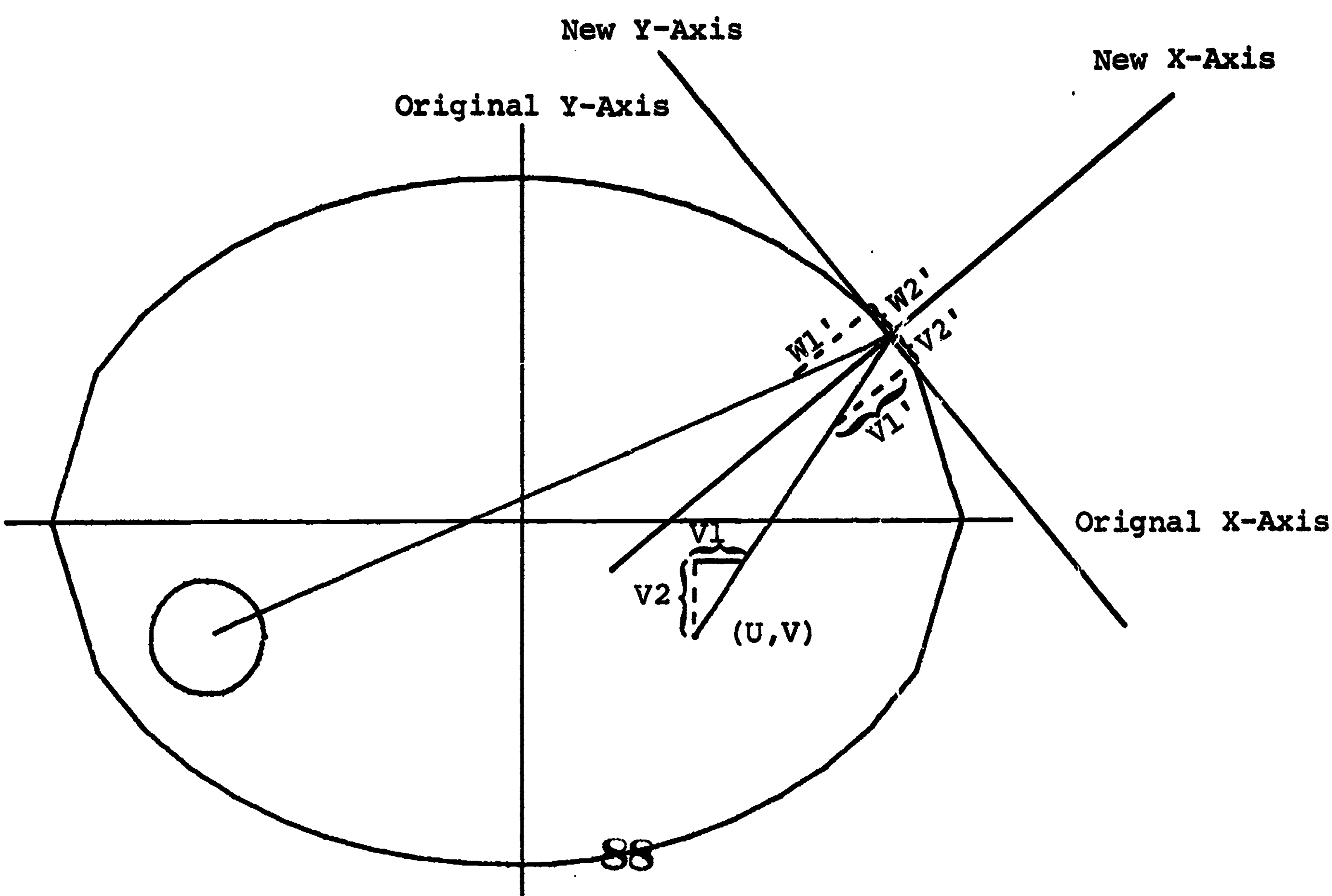
I is the number of the hole it exited.
T is the time for the simulation of this particle.

B5 is the number of times the particle bounced off of the side of the ellipse.
D2 is the distance the particle traveled on the table. Time and distance calculations assume the intial velocity is per unit time.

Subroutine 500 computes the x and y components of the new velocity vector for the particle's bounce. We do this by first finding the slope of the tangent to the ellipse at the point where the particle hit the side. Then we set up a new coordinate system by translating the orgin to the particle's impact point and rotating the axises until the y-axis coincides with the tangent. The bounce effect in this new coordinate system is found by negating the y component of the velocity vector and leaving the x component alone. Once the new components of the velocity vector are found in the translated and rotated coordinate system they are transformed (put back) into the original coordinate system. There is a special case to this method when the y-component is zero. In this case the slope is infinity. The new x-component is found by negating the old x-component and the y-component remains zero. For the special case there is no need to translate and rotate the axises. If the slope is not infinity we work with the new coordinate system as follows:

Let V ^{1}, V2^1 be impact velocity components in the new system.
Let W1^1,W2^1 be rebound velocity components in the new system.
Let W1, W2 be rebound velocity components in the original system.

See Figure 5.



By using polar coordinates we can discover the relationship between the Cartisian coordinates of any point $P(x,y)$ and its image $P^1(x^1,y^1)$ under the rotated and translated system.

The equations for translating between the original coordinate system and the new one are: (The coordinates of the new origin with respect to the old axis are $x = h$ and $y = k$; the angle of rotation is θ.)[1]

$$x^1 = (x-h)*\cos\theta + (y-k)*\sin\theta$$
$$y^1 = (y-k)*\cos\theta - (x-h)*\sin\theta$$

Since we are concerned with vector components and not exact points on the new coordinate system we can eliminate the origin translation from our calculations to find the vector components. The equations become:

$$\therefore\ x^1 = x*\cos\theta + y*\sin\theta$$
$$y^1 = y*\cos\theta - x*\sin\theta$$

$$\therefore\ V1^1 = V1*\cos\theta + V2*\sin\theta$$
$$V2^1 = V2*\cos\theta - V1*\sin\theta$$

$$W1^1 = V1^1 = V1*\cos\theta + V2*\sin\theta$$
$$W2^1 = -V2 = -V2*\cos\theta + V1*\sin\theta$$

Returning to the original system leaving off the translation of origin values again, we have

$$x = x^1*\cos\theta - y^1*\sin\theta$$
$$y = y^1*\cos\theta + x^1*\sin\theta$$

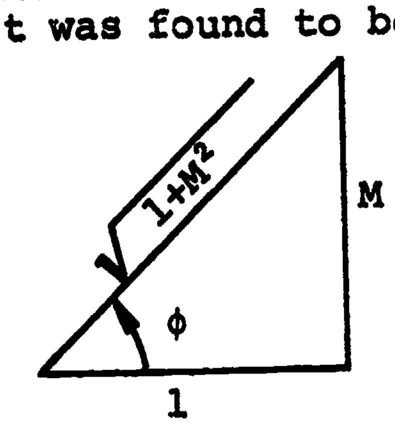
$$\begin{aligned}\therefore\ W1 &= W1^1\cos\theta - W2^1\sin\theta \\ &= (V1*\cos\theta + V2*\sin\theta)*\cos\theta - (-V2*\cos\theta + V1*\sin\theta)*\sin\theta \\ &= (\cos^2\theta - \sin^2\theta)*V2 + (2*\sin\theta\cos\theta)V1\end{aligned}$$

$$\begin{aligned}W2 &= W2^1*\cos\theta + W1^1*\sin\theta \\ &= (-V2*\cos\theta + V1*\sin\theta)*\cos\theta + (V1*\cos\theta + V2*\sin\theta)*\sin\theta \\ &= (\sin^2\theta - \cos^2\theta)*V2 + (2*\sin\theta\cos\theta)V1\end{aligned}$$

We can eliminate the θ from the preceding equations for W1 and W2 by using the slope of the tangent. The slope of the tangent at the point (,y) was found by taking the derivative of the ellipse and solving for dx/dy (the slope). It was found to be $-XB^2/A^2Y$.

Using M as the slope we have:

$$\sin\theta = \frac{M}{\sqrt{1+M}} \qquad \cos\theta = \frac{1}{\sqrt{1+M}}$$



[1] Samual M. Selby, CRC Standard Mathematical Tables, Student Edition, 14 Cleveland Ohio: The Chemical Rubber Co. 1965, p. 515

Thus:

$$W1 = \left(\frac{1}{1+M^2} - \frac{M^2}{1+M^2}\right)V1 - \left(2*\frac{M^2}{\sqrt{1+M^2}} * \frac{1}{\sqrt{1+M}}\right)*V2$$

$$= \left(\frac{1-M^2}{1+M^2}\right)*V1 - 2*\left(\frac{M}{1+M^2}\right)*V2$$

$$W2 = \left(\frac{M^2}{1+M^2} - \frac{1}{1+M^2}\right)*V2 - \left(2*\frac{M}{\sqrt{1+M^2}} * \frac{1}{\sqrt{1+M^2}}\right)*V1$$

$$= \left(\frac{M^2-1}{1+M^2}\right)*V2 - 2*\left(\frac{M}{1+M^2}\right)*V1$$

Subroutine 700 is used to plot the ellipse. Plotting coordinates are found by solving the ellipse equation for a y value give an x value.

$$\frac{X^2}{A^2} = \frac{Y^2}{B^2} = 1$$

$$B^2X^2 + A^2Y^2 = A^2B^2$$

$$Y^2 = \frac{A^2B^2 - B^2X^2}{A^2}$$

$$Y = \frac{SQRT(A^2B^2 - B^2X^2)}{A}$$

Here is a line description of Subroutine 700.

725 A variable is set to plot for the positive y coordinate of the ellipse.
730 The plotter is put into line mode.
740 A loop is set up for the number of points to be plotted for half the ellipse.
750 The x value is determined.
760 The y value is calculated.
770 This statement plots the ellipse.
780 The loop is continued until the termination criteria is reached.
785 The plot mode is ended.
790 If the negative values for y have not been plotted go back and plot them.
795 This statement returns control to the place from which the subroutine was called.

Subroutine 800 is used to plot the holes on the table. Plotting coordinates are found by calculating the Cartesian x and y coordinates using the conversions from polar coordinates. In the calculations R(I) is the radius, the angle θ is going from 0 to 2π to draw the complete circle, and the X(I), Y(I) are added for the circle's center displacement from the origin.

The x coordinate = K = R(I)*cosθ + X(I).
The y coordinate = L = R(I)*cosθ + Y(I).

Here is a line description of Subroutine 800.

810 The plotter is put into line mode.
820 The loop is set up for the number of points to be plotted for the circle.
830 The x coordinate is calculated.
840 The y coordinate is calculated.
850 If a point is out side the ellipse do not plot the value. This allows partial holes on the sides of the ellipse.
860 This statement plots the circle.
870 The loop is continued until the termination criteria is reached.
880 The plot mode is ended.
890 This statement returns control to the place from which the subroutine was called.

The following is a sample run of the program /ELLIP/.

```
 RUN
INPUT THE X-AXIS INTERCEPT ?4
INPUT THE Y-AXIS INTERCEPT ?3
PLTL
PLTL
INPUT HOLES: (X,Y)CENTER, R RADIUS, ABS(X)>X INTERCEPT ENDS LIST.
INPUT X ?-2.645751
INPUT Y ?-1
INPUT R ?.5
INPUT X ?-100
PLTL
INPUT PARTICLE STARTING POSITION X,Y. ?1.645751,-1
INPUT INITIAL VELEOCITY VECTOR: VX,VY. ?1,1.5
PLTL
PARTICLE LEAVES HOLE   1 AT TIME    5.38
AFTER   1 BOUNCE(S);  PATH LENGTH    9.69
IF YOU WISH ANOTHER PARTICLE IN THIS ELLIPSE TYPE YES. ?NO
```

The characters PLTL above indicate that the plotter is activated at this point in the run. The ellipse is drawn first, then the holes, and finally the path of the particle. The picture on the cover of this module (pg. 25) show an actual drawing made by an X-Y plotter. Further information on use of a plotter is found in module 0131.

The following is a listing of the program /ELLIP/.

```
MAY 22 18:55   /ELLIP/


100 REM ELLIPTICAL BILLARD SIMULATOR
102 LET F5=9999
103 A$="'PARTICLE LEAVES HOLE ' ZZD ' AT TIME ' ZZZD.DD/"
104 B$="'AFTER ' ZZD ' BOUNCE(S);  PATH LENGTH 'ZZZD.DD/"
105 DIM X(10),Y(10),R(10),K(10),L(10)
110 PR. "INPUT THE X-AXIS INTERCEPT ":
115 INPUT A
120 IF A<=0 PR. "AXES MUST BE POSITIVE." GOTO 110
125 PR. "INPUT THE Y-AXIS INTERCEPT ":
130 INPUT B
135 IF B<=0 PR. "AXES MUST BE POSITIVE." GOTO 125
136 F1=MAX(A,B) LET F2=2*F1 LET Z=2./3.
137 LET A9=A*A LET B9=B*B
138 GOSUB 700
140 LET H=0
143 PR. "INPUT HOLES: (X,Y)CENTER, R RADIUS, ABS(X)>X INTERCEPT ENDS L
IST."
145 PR. "INPUT X ":INPUT X(H+1)
150 IF ABS(X(H+1))>A GOTO 190
155 PR. "INPUT Y ": INPUT Y(H+1)
160 LET C=X(H+1) LET D=Y(H+1)
170 IF C*C/A9+D*D/B9>1 PR. "HOLE CENTER MUST BE INSIDE ELLIPSE."
GOTO 145
175 PR. "INPUT R ": INPUT R(H+1)
180 IF R(H+1)<=0 PR. "HOLE RADII MUSTBE POSITIVE. " GOTO 175
183 IF R(H+1)>F1 PR. "HOLE RADII MUST BE SMALLER THAN":
PR. " MAJOR AXIS INTERCEPT." GOTO 175
185 H=H+1 GOTO 145
190 GOSUB 800 FOR I=1 TO H
192 PR. "INPUT PARTICLE STARTING POSITION X,Y. ": INPUT U,V
195 IF U*U/A9+V*V/B9>1 PR. "STARTING POSITION OUTSIDE ELLIPSE."
GOTO 192
200 PR. "INPUT INITIAL VELEOCITY VECTOR: VX,VY. ": INPUT V1,V2
205 IF ABS(V1)+ABS(V2)<=0 PR. "NON-ZERO VECTOR, PLEASE. " GOTO 200
210 LET B5=0 LET D2=0 LET S=0
211 PR. "PLTL"
215 LET A1=-V2 LET B1=V1
220 LET C1=U*V2-V*V1 LET D1=SQRT(A1*A1+B1*B1)
223 PR. INT(Z*((U+F1)/F2*F5)), INT ((V+F1)/F2*F5)
225 GOSUB 300
230 GOSUB 400 FOR I=1 TO H
235 IF S=1 GOTO 192
240 GOSUB 500
245 B5=B5+1
250 PR. INT(Z*((P+F1)/F2*F5)), INT ((Q+F1)/F2*F5)
255 LET U=P LET V=Q
260 GOTO 215
300 REM FIND NEXT WALL ENCOUNTER
```

A listing of the program /ELLIP/ continued.

```
305 A2=A1*A1+B1*B1*B9/A9
310 B2=2*A1*C1 LET G2=C1*C1-B1*B1*B9
315 W=SQRT(B2*B2-4*A2*G2)
320 P=(W-B2)/(2*A2)
325 IF B1=0 GOTO 340
330 IF V1*(P-U)<.00001 LET P=(-W-B2)/(2*A2)
335 Q=(-C1-A1*P)/B1 RETURN
340 REM VERTICAL VELOCITY VECTOR
345 P=U LET Q=B*SQRT(1-U*U/A9)
350 IF V2<0 LET Q=-Q
355 RETURN
400 REM CHECK FOR PASSAGE OVER HOLE I
405 D=ABS(A1*X(I)+B1*Y(I)+C1)/D1
410 IF D>R(I) RETURN
415 C=X(I)-U LET E=Y(I)-V
420 IF V1*C+V2*E<.00001 RETURN
425 W=SQRT(C*C+E*E) LET D2=D2+SQRT(W*W-D*D)
430 T=D2/SQRT(V1*V1+V2*V2)
432 PR. INT(Z*((X(I)+F1)/F2*F5)), INT ((Y(I)+F1)/F2*F5)
433 PR. "PLTT"
435 PRINT IN FORM A$:I,T
437 PRINT IN FORM B$:B5,D2
440 PR. "IF YOU WISH ANOTHER PARTICLE IN THIS ELLIPSE TYPE YES. ":
445 INPUT Y$
450 IF Y$<>"YES" STOP
455 S=1 LET I=H RETURN
500 REM SINCE HOLES WERE MISSED, MOVE PARTICLE AND BOUNCE
505 D2=D2+SQRT((P-U)*(P-U)+(Q-V)*(Q-V))
510 IF ABS(Q)=0 LET V1=-V1 RETURN
515 M=-(P*B9/(A9*Q))
520 W1=(V1*(1-M*M)+2*V2*M)/(1+M*M)
525 V2=(V2*(M*M-1)+2*V1*M)/(1+M*M)
530 V1=W1 RETURN
700 REM PLOT ELLIPSE
725 I1=1.
730 PR. "PLTL"
740 FOR I=0 TO 2*A STEP A/10
750 K=I-A
760 L=(I1)*(SQRT(A9*B9-K*K*B9)/A)
770 PR. INT(Z*((K+F1)/F2*F5)), INT((L+F1)/F2*F5)
780 NEXT I
785 PR. "PLTT"
790 IF I1=1. LET I1=-1. GOTO 730
795 RETURN
800 REM PLOT HOLES
810 PR. "PLTL"
820 FOR I1=0 TO 6.28 STEP .2
830 K=R(I)*COS(I1)+X(I)
840 L=R(I)*SIN(I1)+Y(I)
850 IF (K*K)/A9+(L*L)/B9>1 PR. -1,-1 GOTO 870
860 PR. INT(Z*((K+F1)/F2*F5)), INT((L+F1)/F2*F5)
870 NEXT I1
880 PR. "PLTT"
890 RETURN
```