DOCUMENT RESUME

ED 116 927 SE 019 967

AUTHOR	Barnes, Bernis, Ed.; And Others
TITLE	It's a Computerized World: Basic Language for GE Time-Sharing System.
INSTITUTION	Institute for Services to Education, Inc., Washington, D.C.
SPONS AGENCY	National Inst. of Education (DHEW), Washington, D.C.
BUREAU NO	BR-7-0867
PUB DATE	70
CONTRACT	OEC-0-8-070867-0001
NOTE	53p.; Appendix material from ED 084 936; Occasional marginal legibility
EDRS PRICE	MF-$0.76 HC-$3.32 Plus Postage
DESCRIPTORS	College Mathematics; *Computer Programs; *Computer Science Education; Guides; Higher Education; Instruction; *Instructional Materials; Programing; Programing Languages; Secondary Education; *Secondary School Mathematics; *Teaching Guides
IDENTIFIERS	BASIC Language; Thirteen College Curriculum Program

ABSTRACT

This instructional unit of five lessons and four appendices is designed to acquaint both teacher and student with the elementary aspects of computer programming. The first two sections contain background information in computer processes and in BASIC language for a time-sharing system for those teachers who have limited backgrounds and experiences in computer science. Lessons I and II cover giving instructions in English and in BASIC; lesson III deals with translating instructions from English into BASIC; lesson IV introduces conditional control statements through simple programs; and lesson V looks into the use of subscripts in a BASIC program. Each lesson contains suggested teacher questions and related exercises for students. Appendix A contains two programs to be put on tape and checked during lessons II and IV. Appendix B contains six handouts for use with lessons II through VI. Appendix C contains the analysis of programs to find the roots of quadratic equations and also a summary of BASIC symbols. Appendix D contains programs for sums of series, dividing a line, and changing bases. (JBW)

PREFACE

Since one important aspect of this unit is getting the students -- not the teacher -- to initiate and follow through on a particular activity, we have let the teacher play the role of consultant and guide in conducting these classes. In most of these lessons it is only necessary for the teacher to ask one or two questions or set up the computer and pass out the work sheets. We believe that if students are permitted to work through these lessons with a minimum of help, they will be able to write simple programs that contain loops by the end of this unit.

The first two sections of this unit contain background information about computers. This information has been provided for those teachers who have limited backgrounds and experiences in computer science.

This unit contains five lessons which should be done in the order in which they appear. The amount of time necessary to complete any one of them will depend on the class. However, each of them can be completed in a normal class period.

There are handouts in Appendix B to be reproduced for lessons II, III, IV, V, and VI, and programs in Appendix A to be put on tape and checked out for lessons II and IV.

The appendices also contain the analysis of some programs that should be helpful to those students having difficulty with this unit, and several programs that can be used with other units which the students may work through.

CONTENTS

The ever-increasing use of high s eed digital computers is becoming an important factor which touches all areas of our daily lives. Banks, retail outlets, grocery chains, educational institutions, research and scientific investigators are making extensive use of computers and computerized services. Small organizations which find the cost of owning or renting a computer prohibitive can utilize the benefits of the computer on a time-sharing basis at a fraction of the cost of ownership or rental.

Because of its ability to operate at fantastic speed (the term nanosecond -- 10^{-9} seconds -- has been coined to describe it) the computer is especially useful where almost instantaneous answers are required or a great number of reiterative operations are necessary.

We feel that the modern teacher and student should have some familiarity with the concepts of computer operation and programming. This unit is designed to acquaint both teacher and student with the elementary aspects of computer programming. Additional self-study will result in facility in use of this modern device.

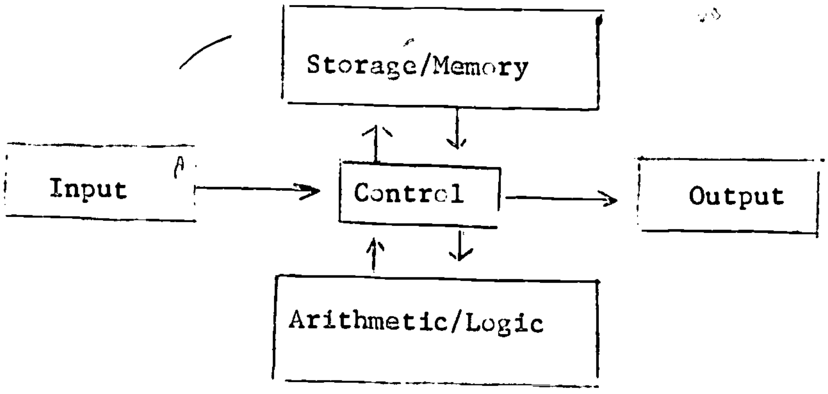
Contrary to popular belief, the computer is not an electronic brain capable of independent thought. Nor is it possible at the present time to talk to the computer and elicit a response. Communication with computer must be given in a language that can be understood by the computer.

Computers are designed to be operated by a machine language -- a sequence of coded commands which are stored in the memory unit until called into use by the control unit. Early programmers found it necessary to learn this machine language which bears no resemblance to any spoken tongues. Synthetic languages are being developed which are designed to bridge the gap between the spoken tongue and the machine language.

The purpose of this unit is to get a feel for what is involved in writing a set of instructions so a computer can give the answer to some problem. The instructions given to the computer constitute a computer program. The language we will use is BASIC (Beginner's All-purpose Symbolic Instruction Code) designed for use with the GE Time-Sharing service.

HOW THE COMPUTER WORKS

A diagram of the flow of information through a computer roughly resembles the one below.



Input consists of the set of statements which make up the program. Input to the computer may be transmitted by punched cards, tape, teletype, or form the keyboard on the console. The statements may be instructions to the computer to perform certain operations or data to be used in the execution of the instrúctions.

The storage/memory unit is where information is stored until needed in the execution of some part of the program.

The arithmetic/logic unit is where the actual computations take place. This unit may add, subtract, multiply or divide; higher-order mathematics is not done directly. This unit also performs certain logical operations; it can determine if a number is positive or negative, zero or non-zero.

The control unit acts in a supervisory capacity and exercises control over all units in the system. It coordinates the activities of the other units by timing and directing the flow of information from one unit to another.

Output may be in the form of punched cards, punched tape, magnatic tape or printed sheets. The output consists of the results obtained from the computer operations.

INTRODUCTION TO THE BASIC LANGUAGE FOR THE GE TIME-SHARING SYSTEM

It is important that students realize that a program is simply a set of statements in a computer language which specifies a sequence of instructions that can be executed by the computer just as instructions given in a human language (English, French, etc.) can be executed by humans. These statements in general will vary from one language to another in order to conform to the rules of the language.

I. Giving Instructions in English

To get an intuitive feeling for communicating with a computer, the teacher may play the role of a simple computer and let the students give him instructions to execute. Start by putting five boxes in a line on a table where all can see and drop face down in an end box the cards ace through five of each suit, already picked out. Say:

A. I am a simple computer and you can give[*] me instructions. Some things I can do, some I cannot. You will find out what I can do as we go along. What should this simple computer do with these cards?

Given some time to think about the problem, they usually will ask, What cards are those? When they do, show them the cards and say:

I have the ace through five of each of the four suits.

[*] This game was adopted from I Am A Simple Computer by Sam Neff and Alice Hankla of the Pre-College Program.

Someone will eventually say, "Sort them either by suit, by number, by color, or by odds/evens."

B. How would you instruct this simple computer to sort the cards?

Suppose they decide to sort them by suits, then they may give the following instructions:

Put the hearts in a box, the clubs in another, and the diamonds in still another.

But these instructions are too complex for a simple computer. Therefore, they must be broken down into several simple instructions, such as:

Pick up the top card.

Put it in box ?.

C. What techinque can we use to direct the computer to a given box?

They should say, "Label all the boxes." Then, "Pick up a card in box 1."

Put it in box 2."

D. Now, can you instruct this computer to sort the cards?

If the answer is yes, then start from the beginning to sort them. On instruction from the students, pick up a card from box 1 (without looking at it), and put it in box 2. If no one objects, pick up another card from box 1 (without looking at it,) and put it in box 2. Sooner or later someone will tell you to READ THE CARD.

At this time the teacher should say:

Suppose we start from the beginning and write the statements on the chalkboard.

1. PICK UP A CARD FROM BOX 1.
2. READ IT.
3. IF 5, PUT IT IN BOX 5. (Students usually have no problem with this statement.)
4. IF A, PUT IT IN BOX 4.
5. IF 3, PUT IT IN BOX 3.
6. IF 2, PUT IT IN BOX 2.

At this point, the students may argue about what to do with the ace. Usually, they will end up writing:

7. IF A, PUT IT IN BOX 5.

Execute these seven statements.

E. Have we finished?

"No! Repeat the steps." The students usually argue about the number of times the steps should be repeated.

8. REPEAT 19 TIMES.

Execute the statements nineteen times.

F. Now, are we finished?

If the students do not remember that the ace's and five's are in the same box, show them the cards that are in each box.

G. How can we separate the ace's and five's?

9. PICK UP A CARD FROM BOX 5.

10. READ IT.

11. IF A, PUT IN BOX 1.

12. IF 5, PUT IN BOX 4.

13. REPEAT 7 TIMES.

Execute steps 9, 10, 11, 12 and 13, then go and execute 1, 2, ...). Then they should want to change statements 8 and 13 to read:

8. REPEAT, STEPS 1 THROUGH 7, 19 TIMES.

13. REPEAT, STEPS 9 THROUGH 12, 7 TIMES.

Pause for someone to suggest the following statements:

14. PICK UP A CARD FROM BOX 4.

15. PUT IT IN BOX 5.

16. REPEAT, STEPS 14 AND 15, 3 TIMES.

H. How will the computer know when we are finished?

17. STOP.

I. Can we delete some of the words in these statements without changing the instructions?

Yes.

This was the final program:

```
 1.  PICK, BOX 1
 2.  READ
 3.  IF 5, PUT BOX 5
 4.  IF 4, PUT BOX 4
 5.  IF 3, PUT BOX 3
 6.  IF 2, PUT BOX 2
 7.  IF A, PUT BOX 5
 8.  REPEAT, 1 THROUGH 7, 19 TIMES
 9.  PICK, BOX 5
10.  READ
11.  IF A, PUT BOX 1
12.  IF 5, PUT BOX 4
13.  REPEAT, 9 THROUGH 12,. 7 TIMES
14.  PICK, BOX 4
15.  PUT BOX 5
16.  REPEAT, 14 AND 15,  3 TIMES
17.  STOP
```

Start from the beginning and execute the entire program.

Then check the cards at the end to see if they are in the right boxes.

Related Exercises

1. Write a program to sort by suit. Also, arrange the cards in each box in ascending order.

2. Write a set of statements that will direct me to your home from here. (This assignment should lead to some very interesting discussions.) Two very important ideas should come out of these discussions:

   (a) The instructions must be given in a mutually understood language.

   (b) Directions (left, right, etc.), and street and highway names should be used for precision.

   Ask some of the students to read their instructions and ask:

   (a) Why did you write your statements in English and not Spanish or French, etc.?

   (b) Why did you list your statements? Why did you not write them in paragraph form?

   (c) Why were street and highway names necessary?

   (d) What must one know to be able to execute the instructions?

II. Giving Instructions in Basic

1. Put Program P-1 (Appendix A) on tape to store in the computer at the beginning of this lesson. Distribute handout sheet, H-1 (Appendix B) and do the first part, one statement at a time. Each answer should be checked with that of the computer before going to the next statement. Ask the following questions, one at a time, to sum up what has been learned.

What have you learned about the Basic language from doing these statements?

Some of the following questions may be answered in the course of this discussion.

(a) What was stored in location A, B and C at the beginning of the program?

(b) Why are the statements numbered?

(c) What is the difference in meaning of the equal sign in Basic statements and in algebraic statements?

(d) What arithmetic operations are used in Basic?

(e) What sign is used to represent each operation in Basic?

Comment: You may change all the answere in the Exercise by assigning a different counting number to A in statement 10 of both the work sheet and program.

Related Exercises

If A = 12 and B = 4, write each of the following statements in Basic so that the computed value of each variable is as given:

45 1. $C = B \cdot A - A/B$

3 2. $D = \dfrac{C \cdot B - A^2}{A}$

6 3. $B = \dfrac{C \cdot B + A \cdot D}{A \cdot D}$

8 4. $A = \dfrac{A}{B} \cdot \dfrac{C + D}{A}$

2 5. $X = \dfrac{A(C + D^2)}{B^D}$

Run your program on the computer to see if your statements are correct.

2. The second part follows the same precedure as the first part. The program for the second part is also P-1. Ask the following questions, one at a time, to sum up what has been learned.

(a) In what order will the computer do the arithmetic operations?

(b) Why are parentheses used in arithmetic expressions?

Related Exercise

Write five arithmetic statements that are acceptable to the computer. Then, put them on the computer to see if they are.

III. Translating Instructions From English to Basic

1. Distribute handout sheet H-2 (Appendix B). Suggest two numbers for executing the English statements in Part 1.

(a) What answer did you get?

You may need to use two or three pairs of numbers before the class is ready to go on to the next question.

(b) Can you write a statement in Basic that will correspond to each of the above statements?

To help students get started, ask the following kinds of questions:

(1) How can you tell the computer to pick two numbers?

(2) How would you tell the computer to double the first? Etc.

When the Basic statements corresponding to the English statements have been written, ask:

(c) Will your set of Basic statements give the same answer as the English statements when they are both executed properly?

If the answer is no, ask:

How would you change the Basic statements to give the same answer you got by executing the English statements?

When they get the same answer with both sets of of instructions, have someone put his program on the computer. If the statement END and statement numbers have not been included, add them in the appropriate places. The program should look like the following:

```
 5 LET A = some number
10 LET B = some number
20 LET C = 2 * A
30 LET C = C + 7
40 LET C = C + 3 * B
50 LET C = C - 11
60 PRINT C
80 END
```

2. While the above program is being put on the computer, the rest may work on part two of the work sheet.

When these programs are understood, ask:

(a) How can the program be written so we can pick different pairs of numbers? (Answer: 10 INPUT A, B and delete statement 5 LET A = ? .)

(b) How can we get this program to work for more than one pair of numbers? (Answer: 70 GO TO 10.)

Related Exercises

1. For any given counting number, write a program that will find three consecutive odd numbers.

2. Write a program that will find the next odd number greater than any given counting number. (The students should be told about the integer function -- LET Y = INT (X) for X = 5/2 assigns to Y the value 2 instead of 2.5.)

IV. Conditional Control

1. Put program P-2 (Appendix A) on tape and store it in the computer. Distribute handout sheet H-3A (Appendix B) and do the statements, one statement at a time. Each answer should be checked with that of the computer before going to the next statement. Ask:

> What is the rule for determining which set of statements will be executed next?
>
> If there is no answer to this question, ask:
>
> (a) When will the computer go to the statement whose number follows the THEN?
>
> (b) When will the computer go the the next statement in the sequence?
>
> The agreed upon rule must work for all ten statements.

2. Distribute handout sheets H-3B (Appendix B) and have them work through and discuss the program one at a time. If the students would like to check their answers with those of the computer, they should put these programs on the computer and run them.

Before running Program 2, the following statements should be added:

```
15 PRINT "10"
25 PRINT "20"
28 PRINT "30"
45 PRINT "40"
48 PRINT "50"
57 PRINT "55"
58 PRINT "60"
65 PRINT "70"
77 PRINT "75"
78 PRINT "80"
```

By counting the number of times a statement number is printed, the studnet can determine the number of times that statement was executed. They may also use their printout to retrace the steps of the computer in the program.

Since some of the statements in Program 4 will be executed many times, suggest adding in counters instead of the PRINT statements as in Program 2 and PRINT statements before the END statement.

Add:

```
15 LET A = A + 1

25 LET B = B + 1

32 LET C = C + 1

34 LET D = D + 1

45 LET E = E + 1

65 LET G = G + 1

68 LET J = J + 1

85 LET K = K + 1

88 LET L = L + 1

92 PRINT "10="; A; ", 20="; B; ", 30="; C; ", 35="; D;", 40=" E

94 PRINT "60="; G;",70=";J;", 80="; K;", 90="; L;",95=1"
```

Related Exercises

1. Given four variables, A, B, C and D with previously defined values, write a sequence of "if, then" statements that will instruct the computer to execute statement number 80, if all four variables have the same value. If one or more variables have a different value, tell the computer to execute statement 45 instead.

2. Given three variables X, Y and Z with no two values the same, if the value of X is:

   (a) greater than both Y and Z, execute statement number 10 next,

   (b) less than Y but greater than Z, execute statement number 20 next,

   (c) greater than Y but less than Z, execute statement number 30 next, and

   (d) less than both Y and Z, execute statement number 40 next.

3. Write a program that will sum the first 20 counting numbers.

4. Write a program that will get the square root of the first 20 counting numbers.

5. Write a program that will find the first 20 odd positive integers.

6. Write a program that will find the prime numbers less than one hundred.

V. Subscripts

Distribute handout sheet H-4 (Appendix B). Work through the program as the computer would and record the printout. Allow students to check their answers with those of the computer.

Ask:

1. How are subscripts represented in Basic?

2. Can you think of a better way to handle this problem?

Related Exercises

1. Set each value of an array called ID equal to the order of the value in the array (that is, ID (1) = 1, ID (2) = 2, etc.). The array has length 20.
2. Store the elements of array A in array B.
3. Store the elements of array A in array B in reverse order.
4. Find the smallest number in an array of 10 numbers.
5. Write a program that will find the pythagorean numbers less than one hundred.
6. Write a program that will subdivide a line eight times.
7. Write a program that will find the number of divisors of a number.
8. Evaluate an N degree polynomial.
9. Arrange a sequence of numbers in ascending order.
10. Find the roots of a polynomial equation to three decimal places.

APPENDIX A

PROGRAM FOR LESSON II

```
  1  PRINT "ARE YOU READY TO DO PART 1 OR PART 2 OF H-1";
  2  INPUT L
  7  LET I = 10
  9  GO TO 110
 10  LET A = 5
 15  GO TO 135
 17  LET A = 5
 18  GO TO 135
 20  LET B = A - A * A
 25  GO TO 135
 27  LET B = -A
 28  GO TO 135
 30  LET C = B - B/A
 35  GO TO 135
 37  LET C = A/B - 1
 38  GO TO 135
 40  LET A = B + A * C/B
 45  GO TO 135
 47  LET C = C + 1
 48  GO TO 135
 50  LET C = B * (A + C)/A
 55  GO TO 135
 57  LET B = B * B + C
 58  GO TO 135
 60  LET A = A * C ↑ 2/B
 65  GO TO 135
 67  LET A = A ↑ 2 - B
 68  GO TO 135
 70  LET A = A/C * B
 75  GO TO 135
 77  LET B = B - C + A
 78  GO TO 135
 80  LET B = B ↑ (C/B) * C/A
 85  GO TO 135
 87  LET C = B * C
 88  GO TO 135
 90  LET A = B * A/C/(C - 2 * B) ↑ 2
 95  GO TO 135
 97  LET B = B/2
 98  GO TO 135
100  LET C = C ↑ 2/2 * (C - 2 * B)
105  GO TO 135
107  LET A = C/B + 12
108  GO TO 135
```

(Cont. on next page)

```
110  LET N = 1
115  PRINT //
120  PRINT "IF STATEMENT";I;"HAS JUST BEEN EXECUTED, WHAT NUMBERS ARE"
122  PRINT "STORED IN A, B, C";
125  LET M = 1
130  INPUT D, E, F
131  IF M = 1 THEN 190
132  IF L = 1 THEN 134
133  ON N GO TO 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
134  ON N GO TO 17, 27, 37, 47, 57, 67, 77, 87, 97, 107
135  IF A <> D THEN 170
140  IF B <> E THEN 170
145  IF C <> F THEN 170
150  PRINT "YOUR ANSWER IS CORRECT"
155  LET N= N + 1
158  LET I = I + 10
160  IF I <= 107 THEN 120
165  GO TO 200
170  PRINT "INCORRECT ANSWER.   TRY AGAIN."
175  LET M = M + 1
176  LET A = X
177  LET B = Y
178  LET C = Z
180  GO TO 130
190  LET X = A
193  LET Y = B
194  LET Z = C
195  GO TO 132
200  END
```

PROGRAM FOR LESSON IV

```
  3  LET M = 10
  5  GO TO 110
 10  LET I = 20
 15  GO TO 135
 20  LET I = 30
 25  GO TO 135
 30  LET I = 80
 35  GO TO 135
 40  LET I = 50
 45  GO TO 135
 50  LET I = 70
 55  GO TO 135
 60  LET I = 90
 65  GO TO 135
 70  LET I = 60
 75  GO TO 135
 80  LET I = 40
 85  GO TO 135
 90  LET I = 100
 95  GO TO 135
100  LET I = 20
105  GO TO 135
110  LET N = 1
120  PRINT "IF STATEMENT";M;"HAS JUST BEEN EXECUTED, WHAT IS THE"
121  PRINT "STATEMENT NUMBER OF THE STATEMENT TO BE EXECUTED NEXT";
131  INPUT J
133  ON N GO TO 10, 20, 30, 40,50, 60, 70, 80, 90, 100
135  IF I = J THEN 195
185  PRINT "INCORRECT ANSWER.   TRY AGAIN."
190  GO TO 131
195  PRINT "CORRECT ANSWER."
198  PRINT /
200  LET M = M + 10
210  LET N = N + 1
220  IF M < 101 THEN 120
230  END
```

APPENDIX B

INSTRUCTIONS IN BASIC

H - 1

Below is a sequence of arithmetic formula statements (Program) in the Basic Language to be executed in the order they are written. After each statement has been executed, what is the current value stored in each variable location assuming each was zero at the beginning of the program?

PART 1

A	B	C		
______	______	______	10	LET A = 5
______	______	______	20	LET B = -A
______	______	______	30	LET C = A/B - 1
______	______	______	40	LET C = C + 1
______	______	______	50	LET B = B * B + C
______	______	______	60	LET A = A ↑ 2 - B
______	______	______	70	LET B = B - C + A
______	______	______	80	LET C = B * C
______	______	______	90	LET B = B/2
______	______	______	100	LET A = C/B + 12

PART 2

A	B	C		
______	______	______	10	LET A = 5
______	______	______	20	LET B = A - A * A
______	______	______	30	LET C = B - B/A
______	______	______	40	LET A = B + A * C/B
______	______	______	50	LET C = B * (A + C)/A
______	______	______	60	LET A = A * C ↑ 2/B
______	______	______	70	LET A = A/C * B
______	______	______	80	LET B = C/A * B ↑ (C/B)
______	______	______	90	LET A = B * A/C/(C - 2 * B) ↑ 2
______	______	______	100	LET C = C ↑ 2/2 * (C - 2 * B)

ENGLISH TO BASIC

H - 2

1. Can you execute the following instructions?

   Pick two numbers.

   Double the first.

   Add seven.

   Add three times the second.

   Subtract 11.

   What do you get?

2. Can you write a statement in Basic that will correspond to each statement above?

3. Can you execute the following instructions?

   Pick two numbers.

   Triple the second.

   Add eight.

   Subtract the first.

   Add seven.

   What do you get?

4. Can you write a statement in Basic that will correspond to each statement above?

CONDITIONAL CONTROL

H - 3A

After each statement in the following sequence has been executed, what is the statement number of the statement that the computer will execute next when X = 10 and Y = 20?

```
__________      10   IF ( X ) < 5 THEN 50

__________      20   IF 2 * X = -Y THEN 70

__________      30   IF X + Y > 5 THEN 80

__________      40   IF Y - X  <= 5 THEN 60

__________      50   IF 4 * X - Y >= 15 THEN 70

__________      60   IF X - 2 * Y < -20 THEN 90

__________      70   IF  3 * X - X > 15 THEN 60

__________      80   IF  4 * Y/-X <= -8 THEN 40

__________      90   IF Y - 2 * X >= 10 THEN 20

__________     100   IF 2 * X + Y <> 10 THEN 20
```

CONDITIONAL CONTROL

H - 3B - 1

1. What is the output for the following program?

```
                                    I          N
10    LET N = 10
                                 _____      _____
20    LET I = I + 3
                                 _____      _____
30    IF I > N THEN 60
                                 _____      _____
40    PRINT I, N
                                 _____      _____
50    GO TO 20

60    END
```

2. Assume that the program shown below has been completely executed, how many times was each statement executed?

```
__________     10    LET N = 6

__________     20    LET I = I + 2

__________     30    IF N < I THEN 55

__________     40    LET N = N + 1

__________     50    GO TO 20

__________     55    LET L = N

__________     60    IF I - N > 10 THEN 75

__________     70    GO TO 20

__________     75    LET L = I

__________     80    END
```

CONDITIONAL CONTROL

H - 3B - 2

3. What is the output for the given data for the following program?

```
                                      M         K

10    LET K = 0                     _____     _____

15    LET J = 1                     _____     _____

20    READ M                        _____     _____

30    IF M = 0 THEN 60              _____     _____

40    LET K = M + J + K             _____     _____

50    GO TO 70

60    LET K = K - 1

70    PRINT M, K

80    LET J = J + 1

90    IF J <= M THEN 20

95    DATA 4, 0, 2

98    END
```

4. Assume that the program shown below has been completely executed, how many times was each statement executed?

```
__________     10    LET I = 20

__________     20    LET N = I

__________     30    LET I = I

__________     35    IF 2 * I - N < 0 THEN 80

__________     40    LET H = SQR(N ↑ 2 + I ↑ 2)

__________     60    LET N = N + I/2

__________     70    GO TO 30

__________     80    LET I = I + 10

__________     90    IF 100 - I > 0 THEN 20

__________     95    END
```

SUBSCRIPTS
H - 4

What is the printout of the following program?

```
10   LET N = 1
20   LET A(N) = N * (N + 1)/2
30   PRINT N, A(N)
40   LET N = N + 1
50   IF N <= 10 THEN 20
60   END
```

	N	A(N)
1st	____	____
2nd	____	____
3rd	____	____
4th	____	____
5th	____	____
6th	____	____
7th	____	____
8th	____	____
9th	____	____
10th	____	____

APPENDIX C

ANALYSIS OF SOME PROGRAMS

Suppose you wanted to solve the quadratic equation $2x^2 + 6x = 56$. You recall that the roots of a quadratic equation may be found by the formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The following program will give the required roots.

<u>Sample Program - 1</u>

```
10 LET A = 2
15 LET B = 6
20 LET C = -56
25 LET Y = SQR (B*B - 4*A*C)  or 25 LET Y = SQR (B ↑ 2 - 4*A*C).
30 LET X1 = (-B+Y) /(2 * A)
35 LET X2 = (-B-Y) / (2*A)
40 PRINT X1,X2
50 END
```

The computer will print:

```
4      -7
```

SQR ( ) must have what meaning?

What does the symbol "↑" mean?

Program No. 1 could be adapted to the solution of any quadratic equation with real roots by inserting the new values for A, B and C in 10, 15 and 20 and keeping the other statements the same.

Suppose, however, that the quantity b^2-4ac yields a negative number, which means that the roots are imaginary. This can be tested by inserting the following statements:

```
28 IF Y < 0 THEN 43
43/PRINT "ROOTS ARE IMAGINARY"
```

Now if the computation for Y yields a negative number, the computer will go to statement number 43 and print, "ROOTS ARE IMAGINARY" and then to statement 45 which ends the program. If Y is not a negative number the computer will proceed to statements 30, 35, 40, 43,50 and will print (for the above values of A,B and C):

```
              4          -7

        ROOTS ARE IMAGINARY
```

Obviously, we do not want statements 40 and 43 executed for the same set of values so we insert a statement which will cause the computer to miss statement 43 when 40 is executed. So we write "42 GO TO 50."

Our complete program now reads:

```
10 LET A = 2
15 LET B = 6
20 LET C = -56
25 LET Y = SQR (B ↑ 2 - 4*A*C*)
28 IF Y < 0 THEN 43
30 LET X1 = (-B+Y) / (2*A)
35 LET X2 = (-B-Y) / (2*A)
40 PRINT X1,X2
42 GO TO 50
43 PRINT "ROOTS ARE IMAGINARY"
50 END
```

To solve any quadratic equation we can insert new values for A, B and C in statements 10, 15, 20 and not alter the other steps. This requires a new program for each equation. With a few changes we can cause the program to handle a series of equation as follows:

```
10 READ A, B, C
```

delete statements 15, 20 and 42 and add statements

```
41 GO TO 10
44 GO TO 10
46 DATA 2, 6, -56
47 DATA 1, 3, 9
48 DATA 5, -4, 7
49 DATA -3, 2, -2
```

Our program now looks like this:

Sample Program - 1a

```
10 READ A, B, C
25 LET Y = SQR (B ↑ 2 - 4*A*C)
28 IF Y < 0 THEN 43
30 LET X1 = (-B+Y) / (2*A)
35 LET X2 = (-B-Y)/ (2*A)
40 PRINT X1,X2
41 GO TO 10
43 PRINT "ROOTS ARE IMAGINARY"
44 GO TO 10
46 DATA 2, 6, -56
47 DATA 1, 3, 9
48 DATA 5, -4,-7
49 DATA -3, 2,  2
50 END
```

The computer will now solve the following equations:

(1) $2x^2 + 6x - 56 = 0$

(2) $x^2 + 3x + 9 = 0$

(3) $5x^2 - 4x - 7 = 0$

(4) $-3x^2 + 2x + 2 = 0$

The computer will print the following:

```
          4            -7
ROOTS ARE I AGINARY
     1.65           -.85
     -.55           1.21656
OUT OF DATA IN 10
```

To avoid placing a long list of DATA statements in the program, we can change statement 10 to read, 10 INPUT A, B, C and delete statements 46, 47 and 49. When the computer executes statement 10, a question mark will be printed. Values for A, B and C can be entered on the teletypewriter separated by commas. For example: for the equation $6x^2 - 4x = 10$, we enter 6, -4, and -10. Our print out would look like this:

```
     ?  6, -4, -10
        1.6666      -1
     ?  3, 7, -5
        .57333      -2.9066
     ?  4, -3, 12
        ROOTS ARE IMAGINARY
     ?
```

This will continue as long as values are supplied for the variables A, B, and C. We get out of the program by typing STOP on the teletypewriter.

We mentioned earlier that there are several types of statements in elementary Basic. Seven of these are included in Sampel Program-la. Let us review the use and meaning of each.

Statement 10 READ A, B, C directs the computer to assign values to the variables listed according to the available numbers in a DATA statement. (A DATA statement must accompany a READ statement.) The computer will assign values as follows: A = 2, B = 6, C = -56; the next time around, A = 1, B = 3, C = 9, and etc., until all numbers presented as data have been used.

Statement 25 LET Y = SQR (B ↑ 2 - 4*A*C) directs the computer to find a value for the expression on the right and assign that value to the variable Y.

Statement 28 IF Y < 0 THEN 43 directs the computer to check to see if the condition Y < 0 exists and to go to a definite place in the program. In this case, if Y is negative, statement 43 is to be executed next, other wise statement 30.

Statement 40 PRINT X1, X2 directs the computer to print out the values for the variables X1 and X2 which follow. Any information enclosed within quotation marks following a PRINT statement (see No. 43) will be printed by the computer.

Statement 41 GO TO 10 causes the computer to go directly to the statement named. In this case the computer goes to Statement 10, which is a READ statement, and looks for data to assign to the variables A, B, and C.

Statement 46 DATA 2, 6, -56 supplies the computer with values to be assigned to variables contained in READ statements.

Statement 50 END informs the computer that the program is now completed and brings the operation to a halt.

Statement 10 INPUT A, B, and C (in our variation) causes the computer to print a question mark (?) and wait for three values to be typed on the teletypewriter.

FOR and NEXT statements are used to creat a "loop;" that is, a set of statements to be executed a number of times. Note the following examples: We want to print the squares and cubes of the first 50 integers. The program below will do this:

```
  4 LET A = 1 ↑ 2
  5 LET B = 1 ↑ 3
 10 PRINT "1,"; A, B
 12 LET A = 2 ↑ 2
 13 LET B = 2 ↑ 3
 15 PRINT "2"; A, B
      .
      .
 98 LET A = 50 ↑ 2
 98 LET B = 50 ↑ 3
200 PRINT "50;" A, B
205 END
```

Our program will contain 151 lines.

By using a FOR ... NEXT loop we can do the same program with only 6 lines.

```
10 FOR X = 1 TO 50
12 LET A = X ↑ 2
13 LET B = X ↑ 3
15 PRINT X, A, B
20 NEXT X
25 END
```

The computer sets X = 1, executes statements 12, 13 and 15, statement 20 causes it to increase X by 1 and return to 10. Then continue until X equals 50. The amount of increase (or decrease) can be changed by including a STEP with the FOR statement. For Example; 10 FOR X = 1 TO 50 STEP 2, will cause the computer to increase X by 2 each time through the loop and the values for the squares and cubes of the <u>odd</u> integers from 1 to 50 will be printed out. The value of a step may be negative or positive. To print the values of the numbers in reverse order we would program, 10 FOR X = 50 TO 1 STEP -1 and the computer will printout the values for 50 first, then 49, and continue until it has completed the loop.

SUMMARY OF SYMBOLS

Symbol	Example	Meaning
+	A + B	Addition, Add B to A.
-	A - B	Substraction, Subtract B from A.
*	A * B	Multiplication, Multiply B by A.
/	A /B	Division, Divide A by B.
↑	B ↑ 3	Raise B to 3rd power, B^3.
=	A = B	The value of B is assigned to A.
<	A < B	Less than, A is less than B.
<=	A <= B	Less than or equal to, A is less than or equal to B.
>	A > B	Greater than, A is greater than B.
>=	A >= B	Greater than or equal to, A is greater than or equal to B.
<>	A <> B	Is not less than or greater than, A = B.

The last six symbols are used in IF ,THEN statements for the comparison of two values. The computer does Basic operations in the following order: raise to a power, multiplication or division, then addition or subtraction. For example, in the expression A + B * C ↑ D/E - F, the computer would raise C to the D power, multiply that result by B, divide that result by E, add A to that value, then subtract F from the result. It is equivalent to the algebraic expression, $A + \frac{B . C^D}{E} - F$. The order can be changed by the use of parentheses as the computer executes the expression inside the parentheses first. To evaluate $\frac{((A + B) . C)^D}{E - F}$ we would write, ((A + B) * C) ↑ D/(E - F). Now the computer adds A and B, multiplies the results by C, raises that value to the D power and divides the result by the difference of E and F.

Certain mathematical functions can be executed by the computer, one of which we have already used, SQR ( ). These functions are given special three-letter names and have the following meanings:

Function	Meaning	
SIN (X)	Find the sine of x	X is an angle measured in radians.
COS (X)	Find the cosine of x	
TAN (X)	Find the tangent of x	
ATN (X)	Find the arctangent of x	
EXP (X)	Find e^X	
LOG (X)	Find the natural logarithm of x (ln X)	
ABS (X)	Find the absolute value of X	
SQR (X)	Find the square root of X	
INT (X)	Find the greatest integer not greater than X	

Given the following program, tell what each statement means or what action is taken by the computer and the values for the variables in each step.

```
 10   PRINT "THIS PROGRAM WILL TEST YOUR USE OF BASIC"
 20   READ X, Y
 25   IF X = Y THEN 45
 30   LET X = Y
 35   READ Y
 40   GO TO 25
 45   LET A = SQR (X)
 50   LET B = INT (A)
 55   FOR R = 1 TO B
 60   LET I = X - R ↑ 2
 65   NEXT R
 70   PRINT "WHAT DO YOU EXTIMATE FOR I";
 75   INPUT G
 80   IF ABS(I - G) < 5 THEN 95
 85   PRINT "NOT SO GOOD, TRY AGAIN"
 90   GO TO 70
 95   PRINT "VERY GOOD.   NOW WRITE YOUR OWN PROGRAM"
100   DATA 243, 15, 85, 500, -650, 90, 43, 725, 340, 169, 169
105   END
```

APPENDIX D

SUMS OF SERIES

Program No. 1

```
10   LET N = N + 1
20   LET A = 1/N
30   LET B = 1/N ↑ 2
40   LET S1 = S1 + A
50   LET S2 = S2 + B
60   PRINT N, S1, S2
70   IF N < 10 THEN 10
80   END
```

Program No. 2

Here is an alternative program that gives the sum of the series for any number of terms K, where the value of K is put in each time.

```
10   INPUT K
20   FOR N = 1 TO K
25   LET A = 1/N
30   LET B = 1/N ↑ 2
35   LET S1 = S1 + A
40   LET S2 = S2 + B
45   NEXT N
50   PRINT "THE SUMS OF THE FIRST";K;"TERMS EQUAL";S1,S2
60   GO TO 10
70   END
```

DIVIDING A LINE

Program No. 1

```
 1  DIM A(513), B(513)
 2  PRINT "NUMBER OF DIVISIONS OF THE LINE IS";
 3  INPUT K
 4  LET J = 0
 6  LET A(1) = 1
 8  LET A(2) = 1
10  LET J = J + 1
12  LET N = 2 ↑ J
15  LET B(1) = A(1)
17  LET M = 2
20  FOR I = 2 TO N STEP 2
25  LET B(I) = A(M) + A(M-1)
30  LET B(I+1) = A(M)
33  LET M = M + 1
35  NEXT I
40  FOR I = 1 TO N+1
45  LET A(I) = B(I)
50  NEXT I
51  PRINT /
53  FOR I = 1 TO N+1
54  PRINT A(I)
60  IF J < K THEN 10
70  END
```

Program No. 2

This program divides the line and determines the frequency of each number.

```
  1  DIM A(513), B(513)
  2  PRINT "NUMBER OF DIVISIONS OF THE LINE IS";
  3  INPUT K
  4  LET J = 0
  6  LET A(1) = 1
  8  LET A(2) = 1
 10  LET J = J + 1
 12  LET N = 2 ↑ J
 15  LET B(1) = A(1)
 17  LET M = 2
 20  FOR I = 2 TO N STEP 2
 25  LET B(I) = A(M) + A(M-1)
 30  LET B(I+1) = A(M)
 33  LET M = M + 1
 35  NEXT I
 40  FOR I = 1 TO N+1
 45  LET A(I) = B(I)
 50  NEXT I
 60  IF J < 1 THEN 10
 65  LET L = 2
 70  LET C(L) = A(2) + A(1)
 75  FOR I = 2 TO N
 80  LET B(I+1) = A(I+1) + A(I)
 85  IF B(L) >= B(I+1) THEN 100
 90  LET L = L + 1
100  NEXT I
110  PRINT "NUMBER OF DIVISIONS MADE IS";J
112  PRINT "NUMBER     FREQUENCY"
115  FOR A = 1 TO B(L)
120  LET X = 0
125  FOR M = 1 TO N+1
130  IF A(M) <> A THEN 200
150  LET X = X + 1
200  NEXT M
205  IF X = 0 THEN 220
210  PRINT TAB(2);A;TAB(10);X
220  NEXT A
240  IF J < K THEN 10
250  END
```

CHANGING BASES

Program No. 1

```
  1  DIM R(15)
  2  PRINT "THIS PROGRAM IS DESIGNED TO CHANGE A NUMBER"
  4  PRINT "FROM BASE 10 TO A SMALLER BASE"
  6  PRINT
  8  PRINT "ENTER YOUR NUMBER AND BASE";
 10  INPUT N, B
 12  IF N = 0 THEN 120
 13  PRINT
 14  LET A = 0
 18  LET M = N
 20  LET X = M/B
 30  LET A = A + 1
 40  LET R(A) = M - B * INT(X)
 50  IF INT(X) = 0 THEN 70
 60  LET M = INT(X)
 65  GO TO 20
 70  LET S = 0
 73  FOR D = A TO 1 STEP -1
 75  LET S = 10 * S + R(D)
 80  NEXT D
 84  PRINT N; "IN BASE 10 EQUALS";S;"IN BASE";B
105  PRINT
110  GO TO 8
120  END
```

Program No. 2

```
  2   DIM Y(25), Z(25)
  4   PRINT "THIS PROGRAM IS DESIGNED TO CHANGE A NUMBER"
  6   PRINT "FROM A SMALLER BASE TO BASE 10"
  8   PRINT
 10   PRINT "ENTER YOUR NUMBER AND BASE"
 12   INPUT N, B
 13   IF N = 0 THEN 400
 14   PRINT
 16   LET A = 1
 17   LET P = N
 18   LET X = N/10
 20   LET W = INT(X)
 25   LET M = W * 10
 30   LET Y(A) = N - M
 35   IF Y(A) >= B THEN 300
 40   LET N = W
 50   IF N = 0 THEN 58
 52   LET A = A + 1
 55   GO TO 18
 58   LET Z(1) = Y(1)
 60   FOR D = 2 TO A
 65   LET Z(D) =Y(D) * B ↑ (D - 1)
 70   IF D = A THEN 100
 75   NEXT D
100   LET S = 0
102   FOR F = 1 TO D
104   LET S = S + Z(F)
106   IF F = D THEN 200
108   NEXT F
200   PRINT P; "IN BASE";B;"EQUALS";S;"IN BASE 10"
204   PRINT
206   GO TO 10
300   PRINT "THERE IS NO SUCH NUMBER IN BASE";B
302   PRINT
304   GO TO 10
400   END
```