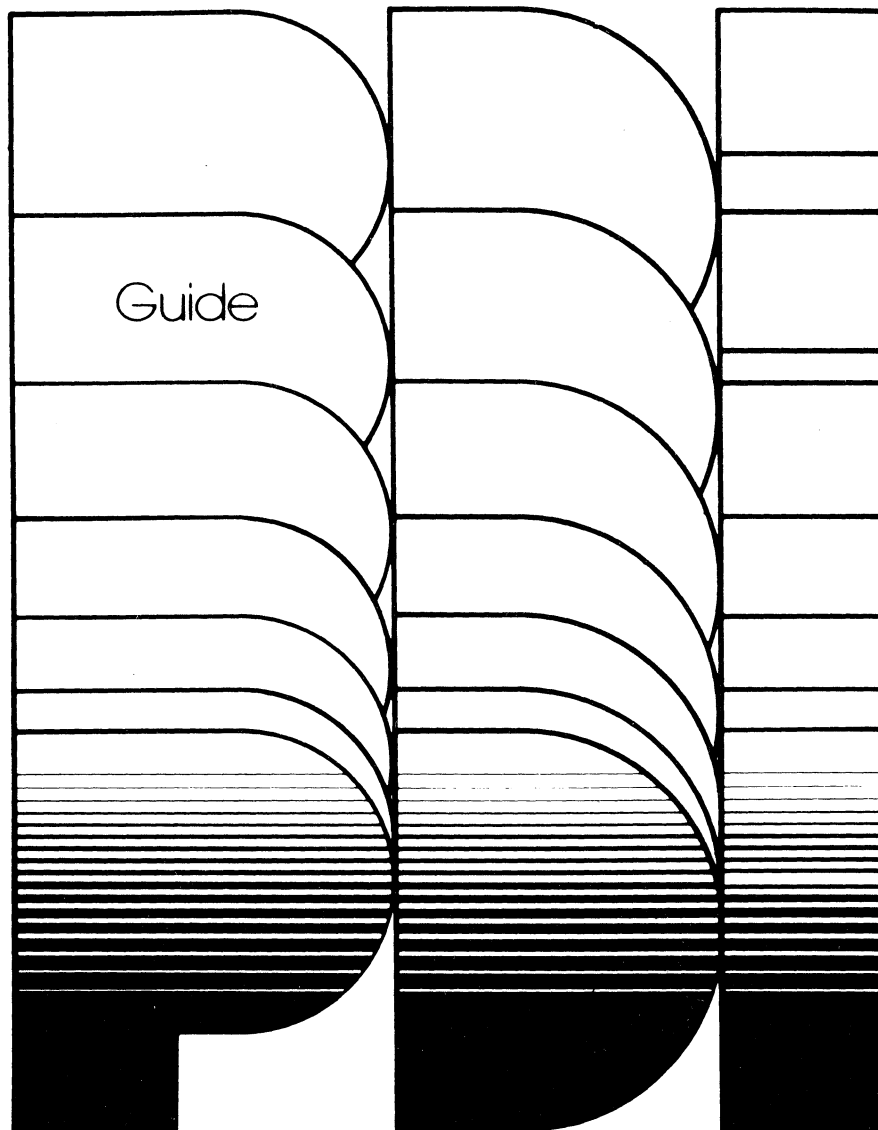


# APPLESOFT 2

# STEP BY STEP



Program Design, Inc. 11 Idar Court Greenwich, Conn. 06830

AUTHORS: CAROL KLITZNER  
JOHN VICTOR

Copyright © 1978 Program  
Design, Inc. All rights  
reserved. No part of this  
material may be reproduced  
in any form without the  
written permission of the  
publisher. Printed in the  
U.S.A.

STEP BY STEP is a course that will teach you to program in BASIC. It is designed for people with no programming experience. The course took over a year to develop using five different types of small computers with junior high school students, homemakers and other beginning programmers as test subjects.

Computer professionals tend to think of BASIC as a simple programming language that anyone can learn in a few hours of practice. The fact is that although BASIC may be easier to learn than assembly language or FORTRAN, it is not all that easy to learn. A person must learn to think in programming terms. Many of the skills a person needs to program in more difficult languages they also need for BASIC.

STEP BY STEP teaches programming skills that are common to all small BASIC-speaking computers. It is not designed to teach the specifics of any one computer although some special features are covered. We feel that a person should learn the basics first and then go on to learn the special features of his or her computer. These features include special graphics, text editing and string manipulations.

#### HOW TO USE STEP BY STEP

The lessons in the course should be completed in sequence. No lessons should be skipped even if you feel that you already know the material being covered. You can do as little or as much as you wish in a single sitting. But, it is best not to let too much time elapse between sessions. Remember, too, that practice sessions are extremely important to effective learning.

Read your loading instructions before loading the tapes. Since the lessons are taped in consecutive order, you may not want to rewind the tapes after each session. The tapes will then be ready for your next session. However, if the tapes are not rewound, they can be easily damaged, so you should handle them carefully and always put them back in the holders. The following list will help you select the right cassette:

STEP BY STEP 1

Side A	Side B
Lesson 1, Part 1	Lesson 3, Part 1
Lesson 1, Part 2	Lesson 3, Part 2
Quiz 1	Quiz 3
Lesson 2, Part 1	Lesson 4, Part 1
Lesson 2, Part 2	Lesson 4, Part 2
Quiz 2	Quiz 4

STEP BY STEP 2

Side A	Side B
Lesson 5, Part 1	Lesson 7, Part 1
Lesson 5, Part 2	Lesson 7, Part 2
Quiz 5	Quiz 7
Lesson 6, Part 1	Lesson 8, Part 1
Lesson 6, Part 2	Lesson 8, Part 2
Quiz 6	Quiz 8

STEP BY STEP 3

Side A	Side B
Lesson 9, Part 1	Quiz for Lessons 1-5
Lesson 9, Part 2	Quiz for Lessons 6-10
Quiz 9	Note: Written questions for quiz are found at the back of this manual.
Lesson 10, Part 1	
Lesson 10, Part 2	
Quiz 10	

## SPECIAL PROBLEMS LOADING CASSETTE TAPES

While it is possible that the tapes we have sent you are defective, most loading problems are caused by faulty equipment or by the way the tapes are being loaded. If you have problems:

1. After the first failure to load a tape, change the volume setting and try again. Do this at several different settings.
2. If you still can't load, clean and demagnetize the heads on the tape recorder. Dirty heads are a common problem.
3. If the problem still persists, turn off the system and try again in a few minutes. This often helps.
4. A discouragingly-high percentage of cassette players are defective when they leave the factory. The most common defect is misalignment of the playing head. A recorder with a misaligned head will often load tapes recorded on it without a problem. But, tapes from other machines or commercial tapes (like PDI's) will not load on these machines. Check to see that your machine will load tapes that have been recorded on other machines.

APPLE II OWNERS: Frequently the computer produces a load error at the beginning of the tape when it encounters the 10-second tone leader on the tape. You can circumvent this problem by listening to the tape until you hear the tone. Load the tape after you hear the tone.

PET OWNERS: Load by typing LOAD rather than by using the RUN key.

TRS-80, LEVEL II OWNERS: Volume is critical with this system. If you are even slightly off the tapes will not load. You may need to try at several volume settings.

If everything fails, the  
tapes may well be defective.  
Send them back and we will  
replace them.

You are now ready to begin. Load Cassette 1, Side A. Your computer will take over as your teacher.

Use the Progress Chart at the back of this manual to keep a record of where you stop each day. You can also record your quiz scores.

Good luck.

## LESSON 1, PART 1

### ●Frame 1

Take a look at the computer terminal's keyboard. When you type something into the computer, it is not official until you press the RETURN or ENTER key at the right side of the keyboard.

When you type zero, do not use the letter O. The computer zero has a slash through it like this: Ø.

When you type a 1, don't use the letter L. The computer has a number "1" key.

If you make a mistake and wish to change your answer, press the backspace key--the one with the arrow pointing to the left.

### ●Frame 2

These numbers are what you'd find in your computer's memory. NOTE the letters are numbers to a computer!

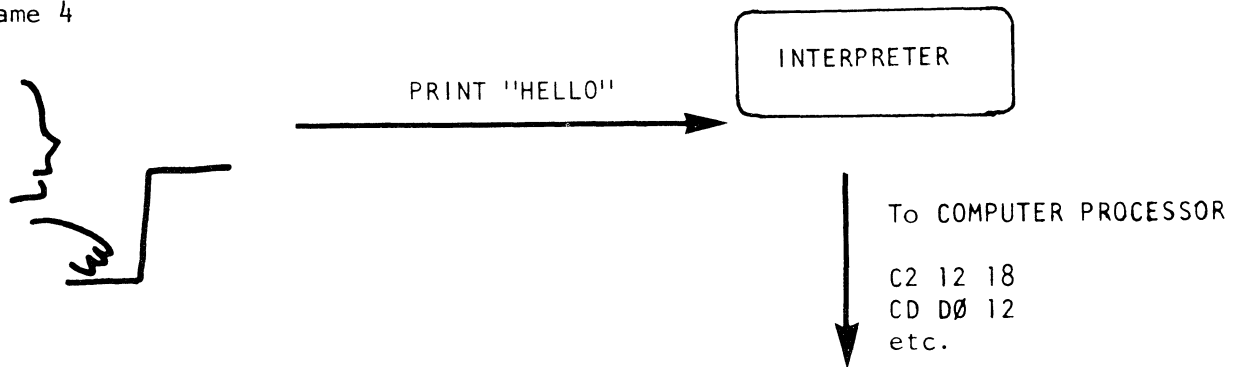
```
ØFØØ 21 ØØ 45 4D 11 ØØ ØE CD 1F ØF B9 CA Ø8 ØF 77 ØØ
ØF1Ø 8Ø 47 23 1B 7A B3 CA 29 ØF CD 1F ØF C3 ØF ØF DB
ØF2Ø ØØ E6 4Ø CA 1F ØF DB Ø1 C9 AF B8 C2 2F ØF 76 FB
ØF3Ø C3 3Ø ØF ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

### ●Frame 3

#### MACHINE LANGUAGE PROGRAM

```
FFØØ D8
FFØ1 58
FFØ2 AØ 7F
FFØ4 8C 12 DØ
FFØ7 A9 A7
FFØ9 8D 11 DØ
FFØC 8D 13 DØ
FFØF C9 DF
FF11 FØ 13
FF13 C9 9B
FF15 FØ Ø3
FF17 C8
FF18 1Ø ØF
FF1A A9 DC
```

● Frame 4



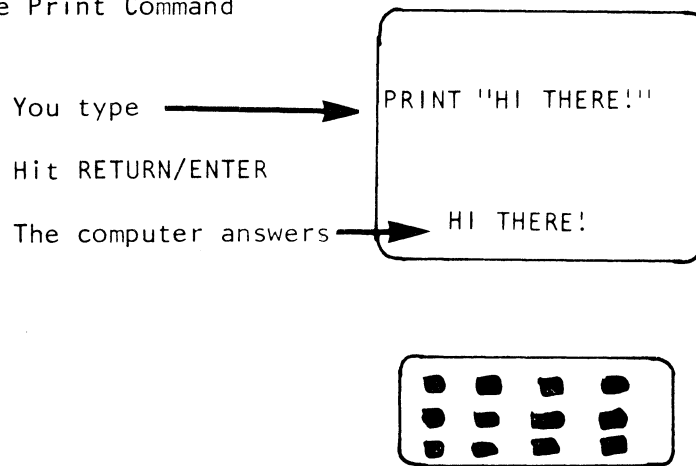
Frame 5

Some Common Basic Commands and Statements

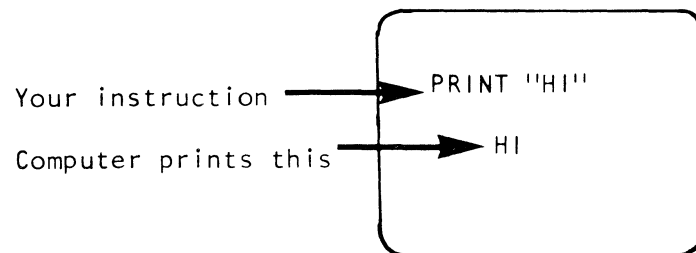
PRINT   STOP   RUN   NEW   SAVE   LOAD   INPUT   GOTO  
READ   DATA   LET   FOR   NEXT   RETURN   RESTORE   IF...THEN

● Frame 6

Use of the Print Command



● Frame 7





●Frame 8

Here are some things to practice with your computer. See what happens when you type the following commands into the computer. NOTE that the quotation marks are over the 2. Push the shift key to type " marks. Press the RETURN/ENTER key after each command.

```
PRINT "HELLO" ← Press RETURN/ENTER after each line
PRINT 'HELLO'
PRINT HELLO
PRNT "HELLO
PRINT "HELLO ← Note: On some computers this will
                not produce an error message. But
                be sure to type the second " or you
                will have trouble with more complicated
                commands.

PRINT "!!??#"
```

●Frame 9

If you make a mistake in typing a line, you can correct it by using the ← key. Press the ← key to move back on the line as many spaces as necessary. Then type in the correct letters and numbers. Try it now:

```
Type:      PRINT "HELO THERE
Next press the ← key 7times
Now finish typing the line so it reads:

      PRINT "HELLO  THERE"
```

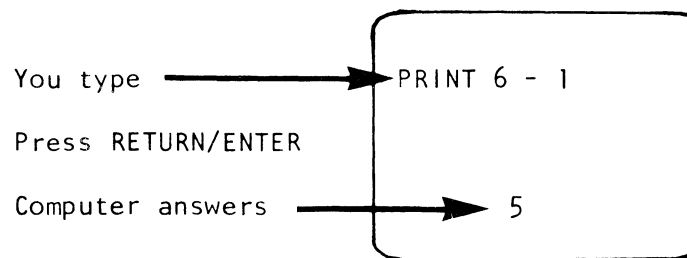
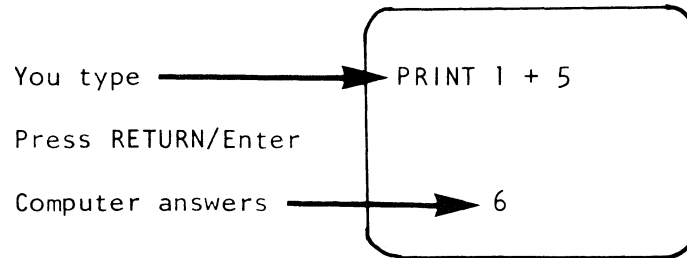
Try these commands, using the ← key to correct any errors you make.

```
PRINT "YOUR NAME"
PRINT "'YOUR NAME'"
PRINT "MY NAME IS 'BOB'"
PRINT "MY NAME IS 'BOB'"
```

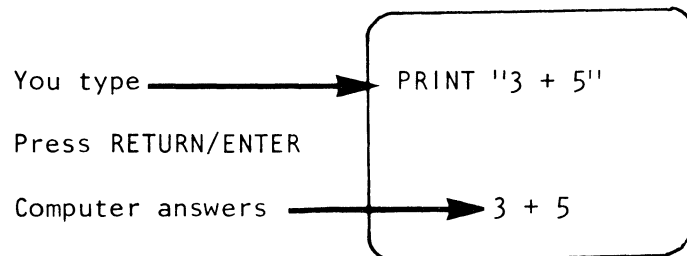
Now practice, on your own, using the PRINT command. Do not use numbers at this time. When you are ready, load Part 2 of Lesson 1 into the computer.

LESSON 1, PART 2

● Frame 10



● Frame 11



● Frame 12

* = multiplication	$2 * 5 = 10$
/ = division	$10 / 2 = 5$
+ = addition	$3 + 3 = 6$
- = subtraction	$10 - 6 = 4$

● Frame 13

\* and / are done first.

Problem:  $3+8*2/4$

$$\left. \begin{array}{l} 3+16/4 \\ 3+4 \end{array} \right\} \quad * \text{ and } / \text{ done}$$

7  $\longleftarrow$  Then + done. Computer prints 7.

Problem:  $10-50/2+8*3$

$10-25+24$

39  $\longleftarrow$  Computer prints.

● Frame 14

Problem:  $(25+8-32)/(9-8)$

1/1  $\longleftarrow$  Work inside () done first.

1  $\longleftarrow$  Computer prints.

Problem:  $16*(4+3-2)/16/(8-7)$

$16*(5)/16/1$

$80/16/1$

5  $\longleftarrow$  Computer prints.

●Frame 15

Here are the steps the computer follows:

- 1) multiplication and division inside ()
- 2) addition and subtraction inside ()
- 3) multiplication and division outside ()
- 4) addition and subtraction outside ()

Here's an example:

Problem:         $(3*4+1) + 2 * (3+3)$   
                  $(12+1) + 2 * (3+3)$  ← \* and / inside ()  
                  $13 + 2 * 6$  ← + and - inside ()  
                  $13 + 12$  ← \* and /  
                  $25$  ← + and -

●Frame 16

Let's practice typing commands into the computer. Type each of these commands and press the RETURN/ENTER key. See what the computer responds.

PRINT 2+6

PRINT 3\*9 ← NOTE: The \* key is usually next to the Ø key.  
                 Press the shift key and hold it while  
                 you type \*.

PRINT 48/6

9ØØØ/2

PRINT 5+6-3+7+19

PRINT "3+3"

Try some others. Then go to Frame 17.

● Frame 17

Look at each example and try to solve the problem using the computer's rules. Then type the command into the computer, press the RETURN/ENTER key, and see how the computer solves it.

If you feel you understand the rules very well, try to answer the last few examples. If you can do them, you can skip the rest.

PRINT 4/2+6

PRINT 3\*2+1

PRINT 10/5+4-5

PRINT 8\*2-6\*2

PRINT (3+6)/(3+3)

PRINT (10\*4-4)/(2\*8-4)

PRINT 2\*(6\*2)-20

PRINT 25+(5\*5)/5

PRINT (6\*4+4)/(3+4)-4

PRINT 100/(10+10)\*(2\*4)-1

● LESSON 2, PART 1

Frame 18

This is a program. Each statement begins with a line number. Here the numbers are 1, 2, and 3.

```
1 PRINT "BEGINNING"  
2 PRINT  
3 PRINT "END"
```

● Frame 19

The commands you've used before are executed as soon as you press RETURN/ENTER. A statement in a program is executed only when the whole program is executed.

● Frame 20

```
1 PRINT "2+2="
2 PRINT 2+2
```

RUN ← You type and hit RETURN/ENTER

2+2=  
4 ← The computer answers with these two lines.

● Frame 21

Look at this program.

```
3 PRINT "HELLO"
4 PRINT
5 PRINT "GOODBYE"
```

● Frame 22

```
12 PRINT "THIS IS ACTUALLY THE MIDDLE"
35 PRINT "THIS IS ACTUALLY THE END"
5 PRINT "THIS IS THE BEGINNING"
```

● Frame 23

```
5 PRINT
10 PRINT "8 + 5 ="
15 PRINT 8 + 5
```

RUN ← first run

8 + 5 = {  
13                      COMPUTER PRINTS

RUN ← second run

8 + 5 = {  
13                      COMPUTER PRINTS

● Frame 24

```
15 PRINT 10*2
5 PRINT
10 PRINT "10 x 2 ="
```

LIST ← LIST Command

5 PRINT  
10 PRINT "10 x 2 =" }  
15 PRINT 10\*2                      COMPUTER PRINTS

● Frame 25

```
10 PRINT "START"  
30 PRINT  
20 PRINT "STOP"
```

You should know a command that will execute this program and a command that will let you see the program typed in the correct order.

LESSON 2, PART 2

● Frame 26

An "error message" is printed by the computer to tell you that the computer found an error and can't continue to run the program with the error in it.

```
?SYNTAX ERROR IN 10
```

Syntax error means "wrong word" or indicates a misspelled command.

● Frame 27

There is a typing error in line 30 of this program. The computer will print an error message.

```
10 PRINT "BEGINNING"  
20 PRINT  
30 PIRNT  
40 PRINT "END"
```

RUN

BEGINNING

?SYNTAX ERROR IN 30

} Computer prints

● Frame 28

Here is one way to correct an error:

Type line number and press RETURN/ENTER.

```
10 PRINT "BEGINNING"  
20 PRINT  
30 PRINT  
40 PRINT "END"
```

30 ← you type and press RETURN/ENTER

LIST ← this command makes the computer print the program again.

```
10 PRINT "BEGINNING"  
20 PRINT  
40 PRINT "END" }
```

Computer Prints (line is gone)

● Frame 29

Another way to correct an error: Retype the line.

```
10 PRINT "BEGINNING"  
20 PRINT  
30 PRINT  
40 PRINT "END"
```

30 PRINT ← You retype the line and enter it.

LIST

```
10 PRINT "BEGINNING"  
20 PRINT  
30 PRINT  
40 PRINT "END"
```

New line

Computer Prints  
(error is corrected)

RUN

```
BEGINNING  
END }
```

The program run

● Frame 30

Look at this program:

```
10 PRINT "START"  
20 PRINT "2+2="  
30 PRINT 2+2
```



● Frame 31

All programs have to end somewhere. To tell the computer where the end is, you type a line number and END. Look at this example.

```
10 PRINT "MATH PROBLEMS"
20 PRINT "2 + 2 ="
30 PRINT 2+2
40 END
```

● Frame 32

Another statement that will stop the program is STOP. Look at the results using STOP and the results using END.

```
10 PRINT "1"
20 PRINT "2"
30 STOP
```

RUN

```
1
2
BREAK AT 30 }
```

COMPUTER PRINTS

```
10 PRINT "1"
20 PRINT "2"
30 END
```

RUN

```
1
2
READY }
```

●Frame 33

Enter these programs and RUN each one to see what happens.

```
10 PRINT "START"
20 PRINT "MIDDLE"
30 PRINT "STOP"
40 END
```

RUN this program.

```
10 PRINT "8 x 5 ="
20 PRINT 8 * 5
30 STOP
```

RUN it.

```
5 PRINT "MATH BY 2'S"
10 PRINT "2 + 2 ="
20 PRINT 2 + 2
30 PRINT "2 - 2 ="
40 PRINT 2 - 2
50 PRINT "2 x 2 ="
60 PRINT 2 * 2
70 PRINT "2/2 ="
80 PRINT 2/2
90 PRINT "THAT'S IT"
100 STOP
```

RUN it.

Add the following to the last program without retyping in the above statements.

```
8 PRINT "ADDING"
25 PRINT "SUBTRACTING"
45 PRINT "MULTIPLYING"
65 PRINT "DIVIDING"
```

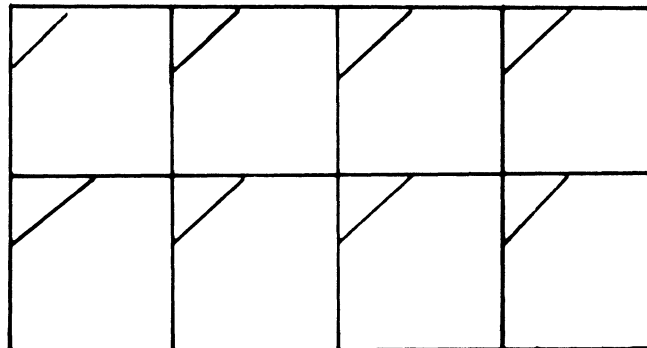
RUN it.

NOTE: Line numbers are typed in 10's to allow room to add more statements between the ones already typed in.

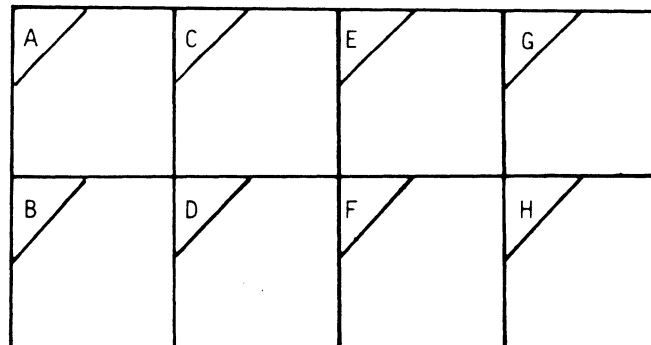
LESSON 3, PART 1

● Frame 34

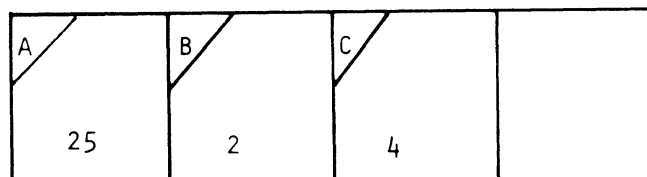
POST OFFICE BOXES



● Frame 35



● Frame 36



● Frame 37

Here are some examples of ways to use the "memory boxes" in different programs:

A is average age of group: 14.2	A is area: 15.73
<u>K</u> is distance in kilometers: 200	<u>L</u> is length: 2
<u>D</u> is number of errors: 6	

The only limitation on using the box is that you have to be able to put a number in the box.

● Frame 38

```
10 LET X=6
20 PRINT X ← Statement
30 END
```

RUN

6 ← COMPUTER PRINTS

● Frame 39

```
10 LET Y=12
20 END
PRINT Y ← Command (no line number)
```

12 ← COMPUTER PRINTS

● Frame 40

```
10 LET A=25
20 PRINT A
30 END
```

● Frame 41

```
10 LET N= -8
20 PRINT N
30 END
```

● Frame 42

```
10 LET Z= .5
20 PRINT Z
30 END
```

●Frame 43

```
10 LET A=10
20 PRINT A + 5
30 END
```

●Frame 44

```
10 LET A = 5
20 LET B = 8
30 PRINT A + B
40 END
```

●Frame 45

```
10 LET P = 10
20 LET S = P
30 PRINT S*S
40 END
```

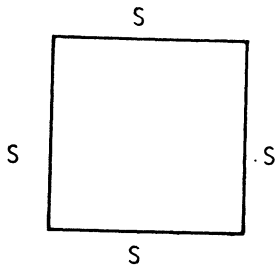
● Frame 46

Frame 46 to 49 contain simple programs using variables. Enter each program and RUN it. Make the suggested changes and RUN again. Type NEW before you go on to the next program.

Do as many of these frames as you like until you feel comfortable with variables.

Program to find the perimeter (distance around) a square.

Type NEW before you begin.



S = SIDE

P = PERIMETER

PERIMETER EQUALS SIDE + SIDE + SIDE + SIDE

```
10 LET S = 5
20 LET P = S + S + S + S
70 PRINT P
80 END
```

Add the statements with arrows to make an expanded program with callouts:

```
10 LET S = 5
20 LET P = S + S + S + S
▶30 PRINT "PERIMETER OF A SQUARE"
▶40 PRINT "SIDE ="
▶50 PRINT S
60 PRINT "PERIMETER ="
70 PRINT P
80 END
```

In the above program, change line 10 and find the perimeter of a square with sides of 12; of 15; of some other number.

● Frame 47

Program to find the area of a square.



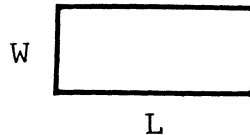
$$\text{AREA} = \text{SIDE} \times \text{SIDE}$$

```
10 LET S = 14
20 LET A = S*S
30 PRINT "AREA OF SQUARE"
40 PRINT "SIDE ="
50 PRINT S
60 PRINT "AREA ="
70 PRINT A
80 END
```

Change line 10 and find the areas of other squares.

● Frame 48

Program to find area of a rectangle.



```
10 LET L = 3
20 LET W = 2
30 LET A = L*W
40 PRINT "AREA OF A RECTANGLE"
50 PRINT
60 PRINT "LENGTH ="
70 PRINT L
80 PRINT "WIDTH ="
90 PRINT W
100 PRINT "AREA ="
110 PRINT A
120 END
```

Change lines 10 and 20 and find other areas.

● Frame 49

Program to convert kilometers to miles.

$$\text{KILOMETER} = .621 \text{ MILES}$$

```
10 LET K = 100
20 LET M = .621*K
30 PRINT "KILOMETERS ="
40 PRINT K
50 PRINT "MILES ="
60 PRINT M
70 END
```

Change line 10 to convert 60 kilometers to miles.

● LESSON 3, PART 2

Frame 50

```
10 PRINT "PROGRAM TO FIND THE AREA OF A SQUARE"
20 LET S = 10
30 LET A = S*S
40 PRINT A
50 END
```

● Frame 51

```
10 PRINT "PROGRAM TO FIND THE AREA OF A SQUARE"
20 INPUT S
30 LET A = S*S
40 PRINT A
50 END
```

← Calls for you to type  
in something.

RUN

```
PROGRAM TO FIND THE AREA OF A SQUARE
? 12 ← You type in 12. S = 12
144 ← Computer prints S*S for A.
```

● Frame 52

```
10 PRINT "PROGRAM TO FIND INTEREST ON PRINCIPLE"
20 INPUT P
30 LET I = .05
40 PRINT I*P
50 END
```

RUN

```
PROGRAM TO FIND INTEREST ON PRINCIPLE } COMPUTER PRINTS
? }
```

● Frame 53

Program 1

```
10 INPUT S
20 LET A = S*S
30 PRINT A
40 END
```

Program 2

```
10 LET S = 15.8
20 LET A = S*S
30 PRINT A
40 END
```

● Frame 54

```
10 INPUT A,B
20 PRINT A + B
30 END
```



● Frame 55

```
10 INPUT A,B
20 PRINT A + B
30 END
```

RUN

```
? 5,3  ← Type in two numbers separated by a comma
8      hit RETURN/ENTER
```

● Frame 56

```
5 PRINT "PROGRAM TO FIND THE AREA OF A SQUARE"
8 PRINT "INPUT THE LENGTH OF SIDE"
10 INPUT S
20 LET A = S*S
25 PRINT "AREA ="
30 PRINT A
40 END
```

● Frame 57

Frames 57 to 59 contain programs to try out. Enter these programs to see how the input statement works. When you think you understand it, take the Quiz for Lesson 3.

```
5 PRINT
10 PRINT "PROGRAM TO ADD ANY TWO NUMBERS"
20 PRINT "INPUT TWO NUMBERS"
30 INPUT A,B
40 PRINT "THE ANSWER IS"
50 PRINT A + B
60 END
```

As practice, make your own programs to do the following:

subtract two numbers

multiply two numbers

divide two numbers

add three numbers

multiply three numbers

● Frame 58

```
10 PRINT "PROGRAM TO CONVERT GALLONS TO LITERS"
20 PRINT "INPUT NUMBER OF GALLONS"
30 INPUT G
40 LET L = 16.907*G
50 PRINT "NUMBER IN LITERS:"
60 PRINT L
70 END
```

● Frame 59

```
10 PRINT
20 PRINT "PROGRAM TO CONVERT FEET TO METERS"
30 PRINT "HOW MANY FEET?"
40 INPUT F
50 LET M = .3048*F
60 PRINT "CONVERTS TO THE FOLLOWING IN METERS:"
70 PRINT M
80 END
```

LESSON 4, PART 1

● Frame 60

```
10 PRINT "THIS IS THE BEGINNING"
20 PRINT "THIS IS THE MIDDLE"
30 PRINT "THIS IS THE END"
40 END
```

GOTO 30 ← GOTO command

THIS IS THE END ← COMPUTER PRINTS

● Frame 61

```
10 LET A = 30
20 GOTO 50
30 PRINT "THE RESULTS ARE"
40 PRINT A*A
50 END
```

● Frame 62

The GOTO statement can make a program loop around and around.

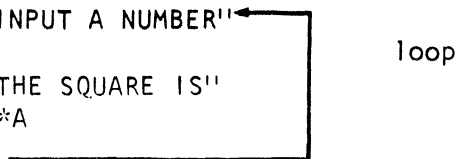
```
10 LET A = 5
20 PRINT A
30 GOTO 10
```

loop

● Frame 63

The purpose of the GOTO loop is to make a program repeat itself. This program keeps squaring any number which is input.

```
10 PRINT "SAMPLE PROGRAM"
20 PRINT "INPUT A NUMBER"
30 INPUT A
40 PRINT "THE SQUARE IS"
50 PRINT A*A
60 GOTO 20
```



loop

● Frame 64

IF-THEN STATEMENT HAS COMPUTER MAKE A DECISION

IF A = 30 THEN STOP

condition

What happens or where computer goes if condition occurs.

● Frame 65

```
10 INPUT A
20 PRINT A
30 IF A = 0 THEN STOP
40 GOTO 10
```

RUN

```
?3
3
? 8
8
? 3000
3000
? 0 ← 0 INPUT
BREAK AT 30
```

When these numbers are input, the computer prints them

● Frame 66

The IF-THEN statement will stop a repeating program. This program will stop when we input -1.

```
10 PRINT "PROGRAM TO ADD 2 NUMBERS"
20 PRINT "INPUT 2 NUMBERS"
30 INPUT A, B
40 PRINT A+B
50 IF A = -1 THEN END
60 GOTO 20
```

● Frame 67

We can use other statements with the IF-THEN statement.

```
10 INPUT A
20 IF A = 1 THEN PRINT "YES"
30 IF A = 2 THEN PRINT "NO"
40 IF A = 3 THEN PRINT "MAYBE"
50 END
```

● Frame 68

In this program, if the input is anything except 1, the computer will go on to 30 and not print anything.

```
10 INPUT X
20 IF X = 1 THEN PRINT "HELLO"
30 END
```

● Frame 69

Run the following program and then try to stop it. You can always turn off the computer if you forget how to break a loop.

```
10 LET A = 10
20 PRINT A
30 GOTO 10
40 END
```

See what happens when this program is run.

```
10 GOTO 20
20 GOTO 10
30 END
```

Try this program .

```
5 PRINT "TYPE A NUMBER"
10 INPUT A
20 IF A = 1 THEN PRINT "YES"
30 IF A = 2 THEN PRINT "NO"
40 IF A = 3 THEN PRINT "MAYBE"
50 IF A = 4 THEN STOP
60 GOTO 5
```

LESSON 4, PART 2

● Frame 70

```
IF A = 3 THEN GOTO 30
```

● Frame 71

```
10 INPUT X
20 IF X = 1 THEN GOTO 50
30 IF X = 2 THEN GOTO 60
40 STOP
50 PRINT X
60 PRINT "END"
70 END
```

● Frame 72

```
10 PRINT "INPUT 1 FOR THE FIRST ROUTINE"
20 PRINT "INPUT 2 FOR THE SECOND ROUTINE"
30 INPUT A
40 IF A = 1 THEN 100
50 IF A = 2 THEN 200
60
70...
80...
90...
100...
110...
120...
200...
210...
220...
```

● Frame 73

```
10 INPUT A
20 IF A = 1 THEN 200
30 IF A = 2 THEN 400
40 IF A = 3 THEN 600
50 IF A = 4 THEN 900
```

```
10 INPUT A
20 ON A GOTO 200, 400, 600, 900
```

● Frame 74

When A is 1 the computer goes to the first number--100. When A is 2, it goes to 200. When A is 3 it goes to 300.

```
10 INPUT A
20 ON A GOTO 100, 200, 300
100 PRINT "HERE"
200 PRINT "WE"
210 GOTO 10
300 PRINT "GO"
310 GOTO 10
```

RUN

```
? 1
HERE }
?3   }
GO   }
etc. }
```

Computer Prints

● Frame 75

```
10 INPUT R
20 ON R GOTO 100, 220, 300, 450, 600
```

● Frame 76

So far we have been using just equal sign (=). Here are some other signs we can use.

$X > Y$  means that number X is GREATER than number Y (the open end is toward the larger number).

$X < Y$  means that number X is LESS THAN number Y (Point is toward smaller number).

● Frame 77

```
10 INPUT N
20 INPUT X
30 IF X<>N THEN STOP
40 GOTO 10
```

● Frame 78

We can mix  $<$ ,  $=$ , and  $>$  signs to mean different things.

$X \leq Y$	X is less than or equal to Y
$X \geq Y$	X is greater than or equal to Y
$X \neq Y$	X does not equal Y

● Frame 79

```
10 INPUT N
20 IF N <= 10 THEN STOP
30 GOTO 10
```

●Frame 80

Enter and RUN these programs trying out different inputs.

```
5 PRINT
10 PRINT "WHERE DOES THIS PROGRAM GO? ENTER A NUMBER"
20 INPUT A
30 ON A GOTO 100, 200, 300, 400
40 PRINT "THE NUMBER WAS TOO BIG"
50 GOTO 5
100 PRINT "LINE 100"
110 GOTO 5
200 PRINT "LINE 200"
210 GOTO 5
300 PRINT "LINE 300"
310 GOTO 5
400 PRINT "LINE 400"
410 GOTO 5
500 END
```

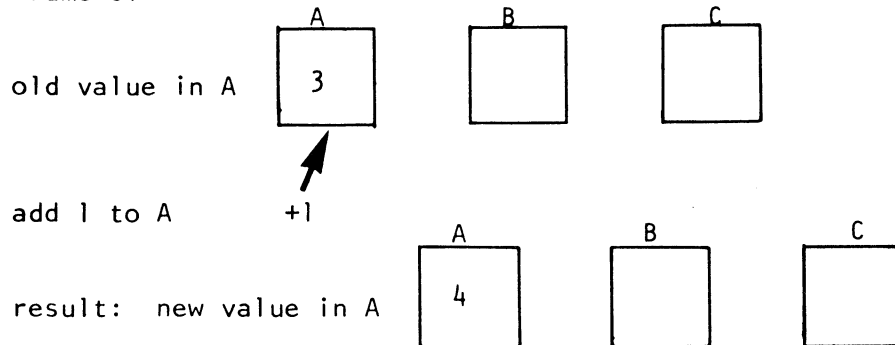
Delete lines 40 and 50 and see what happens.  
Then delete 110, 210, and 310.

```
5 LET N=5
10 PRINT "THIS IS A NUMBER GUESSING GAME"
20 PRINT "TRY TO GUESS THE NUMBER BETWEEN 1 and 10"
30 PRINT "YOUR GUESS?"
40 INPUT G
50 IF G = N THEN 100
60 IF G <> N THEN PRINT "NO, GUESS AGAIN"
70 GOTO 20
100 PRINT "THAT'S IT!"
110 END
```

```
10 PRINT "THIS IS A NUMBER GUESSING GAME FOR TWO"
20 PRINT "PEOPLE. ONE PERSON INPUTS A SECRET"
30 PRINT "NUMBER BETWEEN 1 and 100 AND CLEARS THE"
40 PRINT "SCREEN. THE OTHER PERSON TRIES TO GUESS"
50 PRINT "THE NUMBER. THE PROGRAM SUPPLIES HINTS"
60 PRINT "INPUT THE SECRET NUMBER"
65 INPUT N
70 PRINT "START GUESSING..."
80 INPUT G
90 IF G = N THEN 200
100 IF G > N THEN PRINT "TOO HIGH, GUESS LOWER"
110 IF G < N THEN PRINT "TOO LOW, GUESS HIGHER"
120 GOTO 80
200 PRINT "THAT'S IT!"
210 END
```

## LESSON 5, PART 1

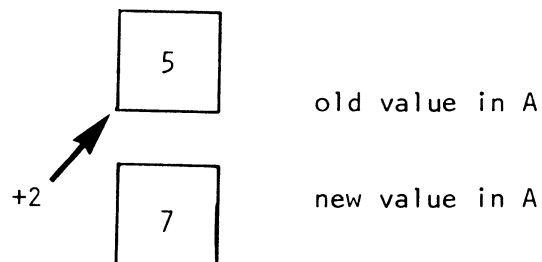
Frame 81



Frame 82

In this example, A was equal to 5. Then we added 2.

Command LET A = A + 2



### ● Frame 83

```
10 LET A = 4
20 LET A = A + 2
30 PRINT A
40 END
```

### ● Frame 84

```
10 LET A = 4
20 LET A = (A + 3) / 7
30 PRINT A
40 END
```

### ● Frame 85

```
5 LET A = 0
20 LET A = A + 1
25 PRINT A
30 GOTO 20
```



● Frame 86

When the program begins, A is set equal to 0. Then the program adds 1 to A. Line 25 checks to see whether A equals 5. Since it doesn't, the program loops back to line 10 and adds 1 to A again.

It continues in this manner until A = 5.

```
5 LET A = 0
10 LET A = A + 1 ← Counter
20 PRINT A
25 IF A = 5 THEN END ← New line
30 GOTO 10
```

● Frame 87

```
10 LET C = C + 1
20 PRINT C*C
30 IF C > 19 THEN END
40 GOTO 10
```

● Frame 88

Each mistake the user makes increases E by 1.

```
10 PRINT "GUESS A NUMBER FROM 1 TO 10."
20 LET N = 5
30 PRINT "START GUESSING"
35 INPUT G
40 IF G <> N THEN LET E = E + 1
45 IF G <> N THEN PRINT "GUESS AGAIN"
50 IF G <> N THEN GOTO 35
60 IF G = N THEN PRINT "YOU GOT IT!"
70 PRINT "YOU MADE THIS NUMBER OF ERRORS"
80 PRINT E
90 END
```

● Frame 89

Line 10 is a REMARK or REM statement.

```
Tells BASIC interpreter to ignore this line. 10 REM THIS IS A SIMPLE PROGRAM ← label for program
20 PRINT 2 + 2
30 END
RUN
4 ← COMPUTER PRINTS
```

● Frame 90

```
10 REM THIS IS A REM STATEMENT
20 REM THE COMPUTER WILL IGNORE IT
30 PRINT "THE COMPUTER WILL NOT IGNORE THIS STATEMENT! "
40 END
```

RUN

THE COMPUTER WILL NOT IGNORE THIS STATEMENT! ←COMPUTER PRINTS

LIST

```
10 REM THIS IS A REM STATEMENT
20 REM THE COMPUTER WILL IGNORE IT
30 PRINT "THE COMPUTER WILL NOT IGNORE THIS STATEMENT! "
40 END
```

} COMPUTER PRINTS

● Frame 91

```
5 REM THE FOLLOWING IS A COUNTER
8 LET A = 0
10 LET A = A + 1
20 PRINT A
30 IF A = 10 THEN END
40 GOTO 10
```

● Frame 92

```
5 REM PROGRAM TO PRINT 1 TO 10
10 B = 0
20 B = B + 1
30 IF B = 10 THEN END
40 PRINT B
50 GOTO 10
```

● Frame 93

What happens when this program is run?

```
10 LET A = A + 1
20 PRINT A
30 GOTO 10
```

This program will stop itself:

```
10 LET A = A + 1
20 PRINT A
30 IF A = 20 THEN PRINT "WE'RE HALF WAY"
40 IF A = 40 THEN PRINT "THAT'S IT"
50 IF A = 40 THEN END
60 GOTO 10
```

●Frame 94

```
5 REM NUMBER GUESSING GAME WITH ERROR COUNTER
6 LET E = 0
10 PRINT "THIS IS A NUMBER GUESSING GAME FOR TWO"
20 PRINT "ONE PERSON INPUTS A SECRET NUMBER BETWEEN"
30 PRINT " 1 AND 100 AND CLEARS THE SCREEN. THE"
40 PRINT "OTHER PERSON TRIES TO GUESS IT. THE"
50 PRINT "PROGRAM SUPPLIES HINTS AND COUNTS ERRORS"
60 PRINT "INPUT SECRET NUMBER AND CLEAR SCREEN"
70 INPUT N
80 PRINT "START GUESSING"
85 INPUT G
90 REM RIGHT GUESS
100 IF G = N THEN 200
110 REM WRONG GUESS
120 IF G > N THEN PRINT "TOO HIGH" GUESS LOWER"
130 IF G < N THEN PRINT "TOO LOW" GUESS HIGHER"
140 REM ERROR COUNTER
150 IF G <> N THEN LET E = E + 1
160 GOTO 85
200 REM CORRECT GUESS
210 PRINT "CORRECT!"
220 PRINT "NUMBER OF ERRORS:"
230 PRINT E
235 LET E = 0
240 PRINT "DO YOU WANT TO GO AGAIN? (1 = YES, 0 = NO)"
250 INPUT A
260 IF A = 1 THEN 60
270 END
```

●Frame 95

```
10 REM PROGRAM TO COMPUTE BANK INTEREST
20 PRINT
30 PRINT "THIS IS A PROGRAM TO COMPUTE COMPOUND"
40 PRINT "INTEREST. YOU INPUT YOUR STARTING"
50 PRINT "PRINCIPLE, INTERST RATE AND NO. OF"
60 PRINT "YEARS THE PRINCIPLE IS IN THE ACCOUNT"
70 PRINT "WE WILL GIVE YOU YOUR BANK BALANCE FOR"
80 PRINT "THE END OF THE PERIOD"
90 PRINT "INPUT 3 NUMBERS SEPARATED BY COMMAS"

95 REM P IS PRINCIPLE, R IS INTEREST RATE
96 REM AND Y IS NO. OF YEARS
100 INPUT P,R,Y
110 REM PRINCIPLE WITH INTEREST ADDED TO FIND NEW PRINCIPLE
120 LET P = P + P*R
130 REM COUNTER FOR NUMBER OF YEARS
135 LET A = A+1
140 IF A = Y THEN 200
150 GOTO 120
200 PRINT "YOUR PRINCIPLE PLUS INTEREST FOR THAT PERIOD OF"
210 PRINT "TIME IS:"
220 PRINT P
230 END
```

## LESSON 5, PART 2

### ● Frame 96

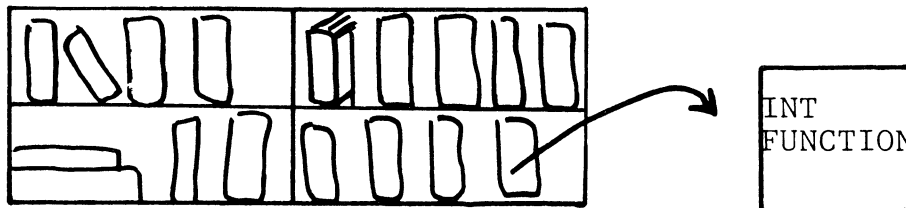
INT: INTeger function. Converts any number into a whole number.

ABS: ABSolute function. Converts any number into a positive number.

RND: RaNDom function. Picks a random number "out of a hat."

### ● Frame 97

These instructions are sometimes called library functions. These functions could all be done by a program you write. Instead of writing a program, the BASIC interpreter's library supplies it for you.



### ● Frame 98

The INT function converts a number to an integer. An integer is a whole number.

INT (3.7) → 3 ← (3.7 becomes a whole number and the computer prints it.)

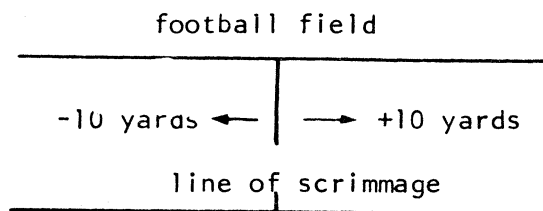
### ● Frame 99

INT will convert a number such as 4.56 to a whole number by chopping off the .56 and leaving 4.

PRINT INT (4.56)

4 ← Computer prints the number without the .56.

● Frame 100



One arrow shows a 10 yard gain: +10. The other shows a 10 yard loss: -10.

The sign of a number shows which way it's going. Sometimes you just want to know the size of a number without the sign. That's the absolute value of the number.

The absolute value of -10 or +10 is 10.

● Frame 101

The ABS command or statement gets rid of the sign.

PRINT ABS (-7) = 7 absolute value

number inside ( )

● Frame 102

```
10 LET A = ABS (-10)
20 PRINT A
30 END
```

● Frame 103

```
10 LET A = ABS (-2.58)
20 PRINT A
30 END
```

● Frame 104

RND stands for RaNDom number. Random numbers are numbers picked by flipping a coin, shuffling a deck of cards, picking a number out of a hat, etc. They are picked completely by chance.

● Frame 105

On different BASIC interpreters, the random number generator works differently. Here are two different methods used to print a random number between 0 and 99.

Commodore PET and Applesoft BASIC

```
10 LET N = RND (1)
20 LET N = 100*N
30 PRINT INT (N)
```

Apple Integer BASIC

```
10 LET N = RND (100)
20 PRINT N
```

● Frame 106

Steps in making a random number between 0 and 99:

1. LET N = RND (1) get decimal random number: .7630058
2. LET N = 100\*N make number between 0 and 99: 76.30058
3. PRINT INT (N) get rid of decimal: 76

● Frame 107

```
10 REM HEADS OR TAILS PROGRAM
20 LET H = 0
30 LET T = 0
40 PRINT "THE COMPUTER WILL FLIP A COIN 10 TIMES."
50 LET N = RND (1)
55 LET N = INT (2*N)
60 IF N = 0 THEN PRINT "HEADS"
70 IF N = 0 THEN H = H + 1
80 IF N = 1 THEN PRINT "TAILS"
90 IF N = 1 THEN T = T + 1
100 IF H + T = 10 THEN 120
110 GOTO 50
120 PRINT "TOTAL HEADS"
130 PRINT H
140 PRINT "TOTAL TAILS"
150 PRINT T
160 END
```

●Frame 108

```
10 REM NUMBER GUESSING GAME WITH RANDOM NUMBERS
20 PRINT "THE COMPUTER WILL THINK UP A NUMBER BETWEEN"
30 PRINT "0 AND 99. YOU TRY TO GUESS IT."
40 REM COMPUTER THINKS UP NUMBER
50 LET N = RND (1) }
60 LET N = INT (100*N) }
70 PRINT "OK, TAKE A GUESS."
80 INPUT G
90 REM RIGHT GUESS
100 IF G = N THEN 200
120 IF G > N THEN PRINT "TOO HIGH, GUESS LOWER"
130 IF G < N THEN PRINT "TOO LOW, GUESS HIGHER"
140 REM ERROR COUNTER
150 LET E = E + 1
160 GOTO 70
200 REM CORRECT GUESS
210 PRINT "CORRECT!"
220 PRINT "NUMBER OF ERRORS:"
230 PRINT E
235 LET E = 0
240 PRINT "DO YOU WANT TO GO AGAIN? (1 = YES 0 = NO)"
250 INPUT A
260 IF A = 1 THEN 50
270 END
```

LESSON 6, PART 1

●Frame 109

```
10 PRINT "HELLO"
20 PRINT
30 PRINT "GOODBYE"
40 END
```

●Frame 110

```
10 PRINT "HOW OLD ARE YOU?"
20 INPUT A
30 PRINT "TELL THE TRUTH"
40 PRINT
50 PRINT
60 PRINT
70 GOTO 10
80 END
```

● Frame 111

When the BASIC interpreter sees a comma in a PRINT statement, it places the material after the comma in the next 'column' on the screen. (Of course, the comma can't be inside quotes. Then the computer would print the comma.)

```
10 A = 1
20 B = 2
30 C = 3
40 PRINT A, B, C
50 END
```

RUN

1    2    3 ← computer prints

● Frame 112

```
10 A = 1
20 B = 2
30 PRINT "A"; "B" ← prints the letters
40 PRINT A;B ← prints the values of A and B
50 END
```

RUN

AB  
12

● Frame 113

```
10 PRINT "BACK"
20 PRINT "UP"
30 END
```

RUN

BACK  
UP ← computer prints

```
10 PRINT "BACK"; ← semi-colon
20 PRINT "UP"
30 END
```

RUN

BACKUP ← computer prints



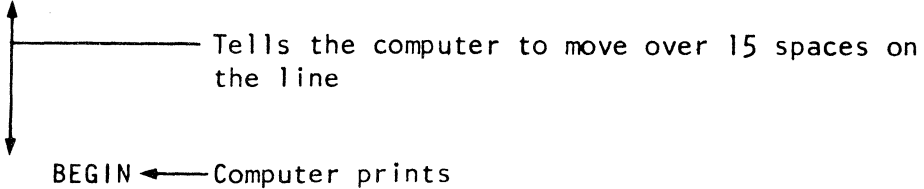
● Frame 114

```
10 PRINT 6,  
20 PRINT 2  
30 END
```

● Frame 115

The "TAB" command or statement works the way a TABulator key works on a typewriter. It tells the computer how many blank spaces to leave on a line.

```
PRINT TAB (15); "BEGIN"
```



Tells the computer to move over 15 spaces on the line

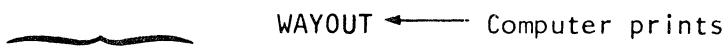
BEGIN ← Computer prints

● Frame 116

TAB is used with a semicolon.

```
10 PRINT TAB (20); "WAYOUT"  
20 END
```

RUN



WAYOUT ← Computer prints

space left by TAB

● Frame 117

RUN these PRINT commands and see what happens.

```
PRINT 5, 6, 7  
PRINT 5; 6; 7  
PRINT TAB (20); B  
PRINT TAB (20); "8+5="   
PRINT TAB (20); 8+5  
PRINT "FIRST"; TAB (20); "LAST".
```

● Frame 118

Sample Programs

```
10 PRINT "LINE";
20 PRINT "SPACE"
30 PRINT
40 PRINT "LINE",
50 PRINT "SPACE"
60 PRINT
70 PRINT "LINE"
80 PRINT "SPACE"
90 END

10 LET A=5
20 LET B=8
30 LET C=A+B
40 PRINT A;"+";B;"=";C
50 END
```

● Frame 119

```
10 REM ADDITION DRILL PROGRAM
20 PRINT "DO THE FOLLOWING ADDITION PROBLEMS"
30 PRINT
40 LET A=INT (10*RND (1))
50 LET B=INT (10*RND (1))

60 PRINT A;"+";B;"=";
70 INPUT C
80 IF C=A+B THEN PRINT "RIGHT!"
90 IF C<>A+B THEN PRINT "WRONG"

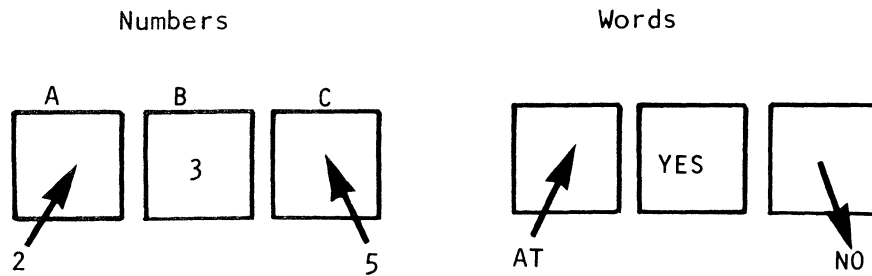
110 PRINT "TYPE 0 TO GO AGAIN."
120 INPUT D
130 IF D=0 THEN 40
140 END

10 REM FANCIER ADDITION PROGRAM
20 PRINT "DO THE FOLLOWING ADDITION PROBLEMS:"
30 PRINT
40 LET A=INT (10*RND (1))
50 LET B=INT (10*RND (1))

60 PRINT A;"+";B;"=";
70 INPUT C
80 IF C=A+B THEN 100
90 PRINT "WRONG TRY AGAIN"
95 LET E=E+1
98 GOTO 60
100 PRINT "CORRECT!"
110 LET R=R+1
120 REM CHECK IF STUDENT HAS DONE 10 PROBLEMS
130 IF E+R=10 THEN 200
140 GOTO 40
200 PRINT "YOU SCORED";R;"RIGHT OUT OF 10."
210 END
```

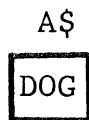
LESSON 6, PART 2

● Frame 120



● Frame 121

A\$ holds strings of characters, so it is called a string variable.



● Frame 122

```
10 LET A$="DOG"
20 PRINT A$
30 END
```

● Frame 123

```
10 LET A$="DO NOT PASS GO."
20 PRINT A$
30 END
```

RUN

DO NOT PASS GO. ← computer prints

● Frame 124

```
10 INPUT A$
20 PRINT A$
30 END
```

RUN

? UP AND DOWN ← input from user

UP AND DOWN ← computer prints

●Frame 125

```
10 LET A$="RIGHT"
20 LET B$="WRONG"
30 PRINT "WHAT IS 3X3?"
40 INPUT N
50 IF N=9 THEN PRINT A$
60 IF N<>9 THEN PRINT B$
70 END
```

●Frame 126

These statements compare two string variables letter by letter to see whether they're the same. In the first statement, the computer prints "RIGHT" if they are.

In the second, the computer goes to 100 if they're not.

In the third, if A\$ is YES the computer prints "RIGHT."

```
1) IF A$=W$ THEN PRINT "RIGHT"
2) IF C$<>T$ GOTO 100
3) IF A$="YES" PRINT "RIGHT"
```

●Frame 127

```
10 LET A$="DOG"
20 PRINT "WHAT'S YOUR FAVORITE ANIMAL?"
30 INPUT F$
40 IF A$=F$ PRINT "MINE TOO!"
50 IF A$<>F$ PRINT "HOW COME?"
60 END
```

●Frame 128

In the following program experiment with things that can be typed into the string variable from the keyboard.

```
10 INPUT A$
20 PRINT
30 PRINT "THIS IS WHAT YOU TYPED INTO THE STRING VARIABLE."
40 PRINT A$
50 END
```

● Frame 129

Try these two programs.

```
10 REM SILLY ANSWERS TO INPUTS PROGRAM
20 PRINT "WHAT IS YOUR NAME?"
30 INPUT A$

40 PRINT
50 PRINT A$;"!  WHAT KIND OF NAME IS THAT?"
55 PRINT
60 PRINT "WHERE DO YOU LIVE?"
70 INPUT B$
80 PRINT B$;"?  I HAVE A FRIEND WHO LIVES IN ";B$;"."
90 PRINT "OK, ";A$;", GO BACK TO WHATEVER YOU WERE DOING"
95 PRINT "IN";B$;"."
100 END

10 REM MADLIB
20 PRINT "TYPE IN THE NAME OF A PLACE OR THING:"
30 INPUT A$
40 PRINT "TYPE IN THE NAME OF A BODY PART:"
50 INPUT B$
60 CLS
70 PRINT "HERE IS YOUR MADLIB"
80 PRINT
90 PRINT "JACK AND JILL"
100 PRINT "WENT UP THE ";A$
110 PRINT "TO FETCH A PAIL OF WATER."
120 PRINT "JACK FELL DOWN"
130 PRINT "AND BROKE HIS ";B$
140 PRINT "AND JILL CAME TUMBLING AFTER!"
150 END
```

← This useful command clears  
the screen for TRS-80.

## LESSON 7, PART 1

### ●Frame 130

You will notice we use a routine similar to this in Step By Step to stop the program while you read something.

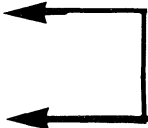
```
1000 PRINT
1010 PRINT "TYPE 0 AND PRESS THE RETURN KEY."
1020 INPUT A
1030 PRINT
```

### ●Frame 131

In this program, we use the same routine several times, so we moved it to the end instead. Now it's called a subroutine.

```
10 ...
20 ...
30 IF A=1 THEN PRINT "YES"
40 ...
50 ...
60 IF A=1 THEN PRINT "YES"
70 ...
80 END

1000 IF A=1 THEN PRINT "YES"
```



Routines taken out of program.

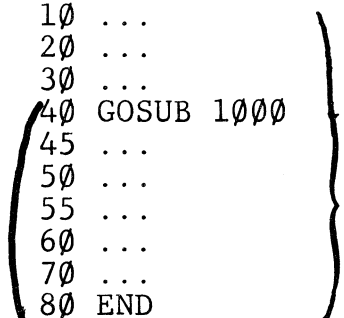
Subroutine written just once.

### ●Frame 132

When the computer reaches the GOSUB statement in line 40, it goes to line 1000.

```
10 ...
20 ...
30 ...
40 GOSUB 1000
45 ...
50 ...
55 ...
60 ...
70 ...
80 END

1000 PRINT
1010 PRINT "TYPE 0 AND PRESS THE RETURN KEY"
1020 INPUT A
1030 PRINT
```



Regular program

Subroutine

Frame 133

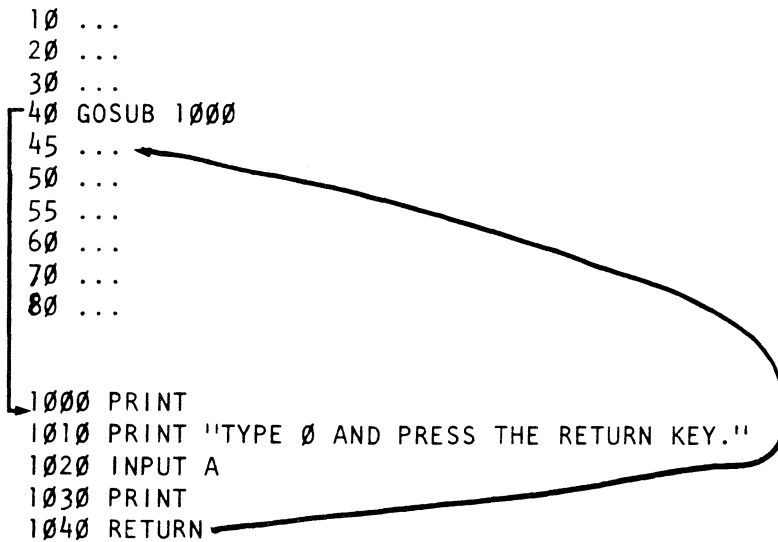
Of course, once the computer gets into the subroutine, you need a way to get it back into the program to pick up and continue where you left off.

A RETURN statement returns the computer to the statement right after the GOSUB statement.

Program	Subroutine
10 ...	1000 PRINT
20 ...	1010 PRINT "TYPE Ø AND PRESS THE RETURN KEY."
30 ...	1020 INPUT A
40 GOSUB 1000	1030 PRINT
45 ...	1040 RETURN ← sends computer back to main program
50 ...	
55 ...	
60 ...	
70 ...	
80 ...	

Frame 134

Here is how the RETURN statement works.



Frame 135

- Look at this program to answer the next few questions.

```
10 REM SIMPLE SUBROUTINE PROGRAM
20 PRINT "FIRST TIME THRU SUBROUTINE:"
30 GOSUB 1000
40 PRINT "SECOND TIME THRU:"
50 GOSUB 1000
60 PRINT "THIRD TIME THRU:"
70 GOSUB 1000
80 STOP
1000 PRINT
1010 PRINT 1;2;3;4;5
1020 PRINT
1030 RETURN
1040 END
```

● Frame 136

Here is the end of an addition drill program.

```
110 PRINT "ADD 2 + 2"
120 INPUT S
130 LET A = 4
140 GOSUB 180
150 PRINT "ADD 3 + 3"
160 INPUT S
170 LET A = 6
175 GOSUB 180
180 IF S = A THEN PRINT "RIGHT"
190 IF S <> A THEN PRINT "WRONG"
200 RETURN
210 PRINT "GO ON TO THE NEXT LESSON"
```

GOSUB statements

} subroutine

● Frame 137

```
10 REM GOSUB DEMONSTRATION
20 PRINT "GO TO SUBROUTINE"
30 GOSUB 1000
40 PRINT "AGAIN"
50 GOSUB 1000
60 PRINT "THIRD TIME"
70 GOSUB 1000
80 STOP

1000 PRINT "HERE'S THE ROUTINE"
1010 PRINT "*****"
1020 PRINT
1030 RETURN
```

RUN this. Then take out line 80 and RUN it.



● Frame 138

```
10 REM ARITHMETIC PROGRAM WITH SUBROUTINE
20 PRINT "WHAT'S YOUR NAME?"
30 INPUT A$
40 PRINT
50 PRINT A$;" , DO THE FOLLOWING MULTIPLICATION PROBLEMS."
60 LET A = INT (10*RND (1))
70 LET B = INT (10*RND (1))
80 PRINT A; "X"; B; "=";
90 INPUT C
100 IF C = A*B THEN 140
110 PRINT "WRONG. TRY AGAIN"
120 LET E = E + 1
130 GOTO 80
140 GOSUB 1000
150 REM CHECK TO SEE IF STUDENT HAS DONE 10 PROBLEMS
160 IF E = R = 10 THEN 200
170 GOTO 60
200 PRINT A$;" , YOU SCORED ";R;" RIGHT OUT OF 10."
210 END
1000 REM SUBROUTINE
1010 LET Q = INT (3*RND (1))
1020 IF Q = 0 THEN PRINT "CORRECT"
1030 IF Q = 1 THEN PRINT "YOU'VE GOT IT, ";A$
1040 IF Q = 2 THEN PRINT "EXCELLENT"
1050 LET R = R + 1
1060 PRINT
1070 RETURN
```

LESSON 7, PART 2

● Frame 139

This program has a READ statement and a DATA statement.

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,2,3,4,5,6
```

● Frame 140

Here's what happens: The computer takes the first number in the DATA statement and puts it in the variable in the READ statement.

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,2,3,4,5,6
```

RUN

1 ← first time through, this is X

● Frame 141

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 8,5,7,10
```

● Frame 142

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,2,3,4,5,6
```

RUN

```
1
2  }
3  }
4  } computer prints X
5  }
6  }
```

OUT OF DATA ← computer prints error message for your machine

● Frame 143

```
10 READ A
20 PRINT A*A
30 GOTO 10
40 DATA 3,2,8
```

● Frame 144

Here RESTORE makes the computer use the data again.

```
10 READ Y
20 PRINT Y
30 IF Y=-1 THEN RESTORE
40 GOTO 10
50 DATA 1,2,3,-1
```

RUN

```
1
2
3
-1 ← data restored here
1
2
3
etc.
```

} computer prints

● Frame 145

```
5 DATA 8,3,6,4,0
10 READ X
20 PRINT X
30
40 GOTO 10
```

● Frame 146

The numbers in DATA must be separated by commas.

```
10 DATA 6,8,3
20 READ A
30 IF A=3 THEN RESTORE
```

● Frame 147

Here the computer reads a number for X and the next one for Y each time.

```
10 READ X,Y
20 PRINT X,Y
30 GOTO 10
40 DATA 1,2,3,4,5,6
```

RUN

1	2
3	4
5	6

} Computer prints X,Y each time.

OUT OF DATA ERROR ← Computer is out of data.

● Frame 148

```
5 REM ADDITION EXAMPLES
10 DATA 1,3,2,3,2,1,2,3,1
20 READ A,B,C
30 PRINT A;"+";B;"+";C;"=";
40 PRINT A+B+C
50 GOTO 20
```

● Frame 149

```
10 REM READ...DATA PRACTICE
20 READ A
30 PRINT A
40 GOTO 10
50 DATA 5,6,7,8
```

RUN the above program and then add the following line to it:

```
35 IF A=8 THEN RESTORE
```

Add the following lines:

```
5 LET B=0
36 LET B=B+1
37 IF B=8 THEN STOP
```

Add the following line:

```
6 DATA 1,2,3,4
```

Add the following lines:

```
20 READ X,Y
30 PRINT X,Y
and delete 35, 36, and 37
```

● Frame 150

```
10 REM DECIMALS DRILL PROGRAM
20 PRINT "DO THE FOLLOWING DIVISION TO 2 DECIMAL PLACES:"
30 PRINT
40 READ A,B,C
50 PRINT A;" / ";B;" = ";
60 INPUT D
70 IF D<>C THEN PRINT "NO, THE CORRECT ANSWER IS";C
80 DATA .1,.2,.5,.8,.2,4,.9,1.2,.75
90 DATA .7,.2,3.5
100 IF D=C THEN PRINT "CORRECT!"
105 IF C=3.5 THEN END
110 GOTO 40
120 END
```

LESSON 8 PART 1

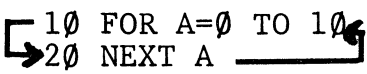
● Frame 151

```
10 REM COUNTING LOOP
15 LET A=0
20 PRINT A
30 LET A=A+1
40 IF A=11 THEN END
50 GOTO 20
```

● Frame 152

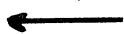
```
10 FOR A=0 TO 10
```

● Frame 153

give A a value and go on.  go back to  
get next A

● Frame 154

This program has a PRINT statement placed between the FOR and NEXT statements. Each time the program loops, the PRINT statement prints the new A .

```
10 FOR A=0 TO 10
20 PRINT A;  (; makes the printing all on one line.)
30 NEXT A
```

● Frame 155

```
10 FOR X=1 TO 7
20 PRINT "*";
30 NEXT X
40 END
```

● Frame 156

```
10 FOR Y=10 TO 50 STEP 10
20 PRINT Y
30 NEXT Y
40 END
```

RUN

```
10 }
20 } — steps of 10 }
30 }
40 }
50 }
```

computer prints

● Frame 157

```
10 FOR Z=2 TO 12 STEP 2
20 PRINT Z;
30 NEXT Z
40 END
```

● Frame 158

```
10 FOR A=10 TO 1 STEP -1
20 PRINT A
30 NEXT A
40 END
```

● Frame 159

```
10 REM SIMPLE FOR...NEXT PROGRAM
20 FOR A=1 TO 10
30 PRINT A
40 NEXT A
50 END
```

Add this line to the above program:

```
35 PRINT A*A
```

Add this line:

```
45 PRINT "THE END"
```

● Frame 160

```
10 REM COUNTDOWN PROGRAM
20 FOR X=10 TO 0 STEP -1
30 PRINT X
40 NEXT X
50 PRINT "*****BOOM*****"
```

note: add a screen clear command

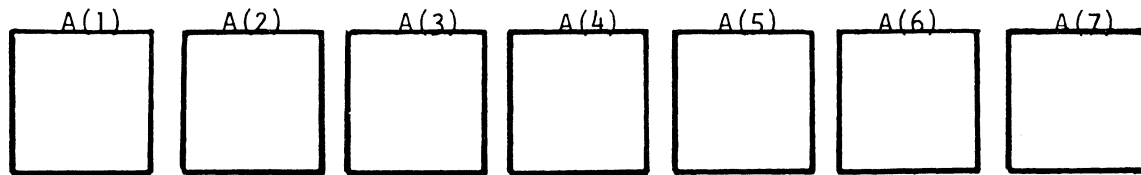
Add this to the above program:

```
35 FOR Y=1 TO 500 }
36 NEXT Y          timing loop
```

LESSON 8, PART 2

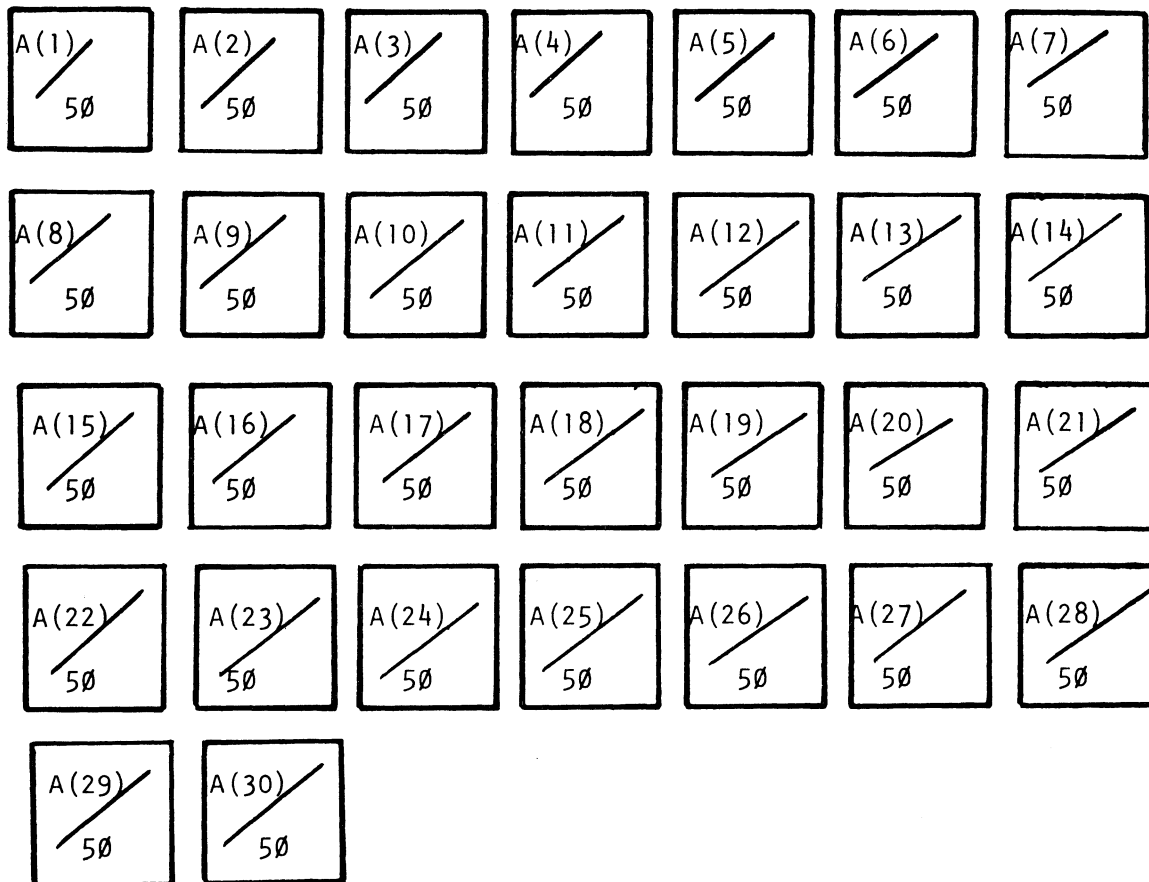
● Frame 161

AN ARRAY

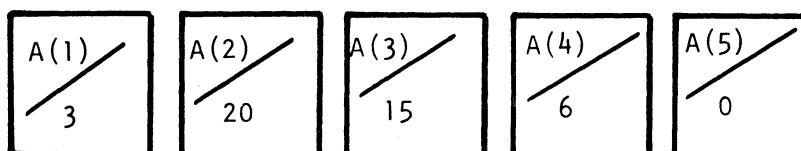


● Frame 162

Here we have set up an array for 30 days. We put 50 into each variable to stand for 50 tickets.



● Frame 163



...etc.



● Frame 164

```
5 LET X=0
10 LET X=X+1
20 LET A(X)=5
30 IF X=10 THEN STOP
40 GOTO 10
```

● Frame 165

```
10 FOR X=1 TO 30
20 LET Z(X)=50
30 NEXT X
40 END
```

● Frame 166

5 DIM A(30) (Put this in if you are using LEVEL II BASIC.)

```
10 FOR X=1 TO 30 }
20 LET A(X)=50    } Sets up an array with 50 in each variable.
30 NEXT X         }
40 PRINT "WHAT DAY DO YOU WANT A TICKET FOR?"
50 INPUT X
60 IF A(X)=0 PRINT "SORRY. NO MORE TICKETS. TRY ANOTHER DAY."
70 IF A(X)<>0 GOTO 90 { Lines 60 and 70
80 GOTO 40           { check whether tic-
90 PRINT "THERE ARE ;A(X);TICKETS LEFT." { kets are left.
100 PRINT "SELL A TICKET FOR THAT DAY."
110 LET A(X)=A(X)-1
120 GOTO 40
```

● Frame 167

Sample Program

```
10 PRINT "THIS IS THE RESERVATION SYSTEM FOR"
20 PRINT "LOWER SLOBOVIA AIRWAYS. OUR BIPLANE HAS"
30 PRINT "4 SEATS ON IT. YOU CAN RESERVE FLIGHTS 1 TO"
40 PRINT "7 (SORRY, ALL FLIGHTS SECOND-CLASS)."
```

50 REM SET UP SEATS ON PLANE

```
60 FOR X=1 TO 7
70 LET A(X)=4
80 NEXT X
90 PRINT "WHAT FLIGHT (1 THRU 7) DO YOU WANT?"
100 INPUT X
110 IF A(X)=0 THEN PRINT "SORRY, FLIGHT CLOSED. TRY ANOTHER..."
120 IF A(X)<>0 THEN 140
130 GOTO 90
140 REM REMOVE SEAT FROM AVAILABILITY
150 LET A(X)=A(X)-1
160 PRINT "OK TO ISSUE TICKET FOR THAT FLIGHT."
170 GOTO 90
```

LESSON 9, PART 1

● Frame 168

```
10 FOR A=1 TO 3
20 PRINT "*****"
30 NEXT A
40 END
```

● Frame 169

```
10 FOR A=1 TO 3
20 PRINT "1*****"
30 PRINT "2*****"
40 PRINT "3*****"
50 PRINT "4*****"
60 PRINT "5*****"
70 NEXT A
80 END
```

RUN

```
1*****
2*****
3*****
4*****
5*****
```

} On each loop, this is what  
the computer prints.

● Frame 170

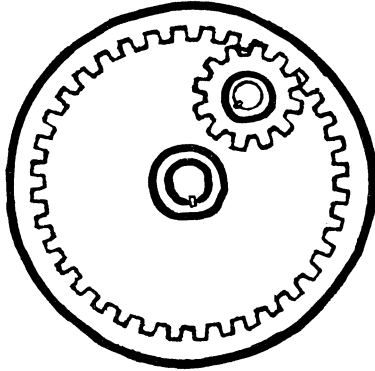
```
10 FOR A=1 TO 3
20 FOR Z=1 TO 5
30 PRINT Z;"*****"
40 NEXT Z
50 NEXT A
60 END
```

RUN

```
1****
2****
3****
4****
5****
```

} Computer prints this 3 times.

● Frame 171



● Frame 172

```
10 FOR A=1 TO 100 ← Outside loop
20 FOR B=1 TO 5   ← Inside loop
30 NEXT B
40 NEXT A
```

● Frame 173

There's an error in the Z loop here.

```
10 FOR A=1 TO 100
20 FOR Z=1 TO 20
30 PRINT Z
40 NEXT A
50 NEXT Z
```

RUN

1  
1  
1  
1

...

WHAT?

50 NEXT Z?

} Computer prints the first Z value 100 times.  
Then you get an error message.

●Frame 174

Program 1

```
10 FOR X=1 TO 10
20 FOR Y=3 TO 1 STEP -1
30 PRINT Y
40 NEXT Y
50 NEXT X
```

Program 2

```
10 FOR A=1 TO 10
20 FOR B=1 TO 3
30 PRINT B
40 NEXT A
50 NEXT B
```

●Frame 175

```
10 FOR P=1 TO 10
20 PRINT "PROBLEM";
30 FOR N=1 TO 2
40 PRINT "GIVE ME A NUMBER"
50 INPUT A(N)
60 NEXT N
70 PRINT A(1);"X";A(2);"=";A(1)*A(2)
80 NEXT P
90 END
```

●Frame 176

First enter and RUN the program you just looked at in Frame 175.  
Then try this one.

```
10 CLS
20 PRINT "HOW MANY PEOPLE IN YOUR FAMILY?"
30 INPUT N
40 FOR C= 1 TO N
50 PRINT "HOW OLD IS PERSON";C
60 INPUT A(C)
70 NEXT C
80 PRINT "HERE'S A GRAPH OF YOUR AGES."
90 FOR C=1 TO N
100 PRINT "PERSON";C
110 FOR X=1 TO A(C)
120 PRINT "*";
130 NEXT X
140 PRINT
150 NEXT C
```

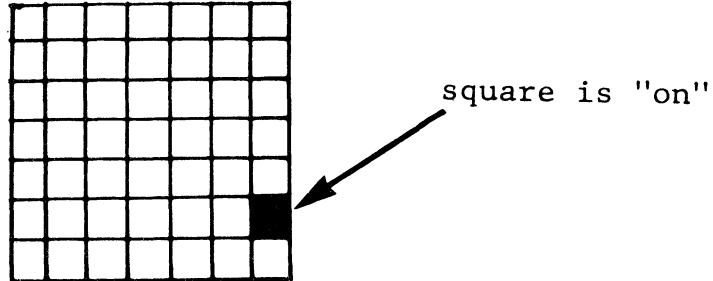
Add lines 160 and 170 and RUN again.

```
160 FOR X=1 TO 5000 }      Timing loop holds display
170 NEXT X
```

## LESSON 9, PART 2

### ● Frame 177

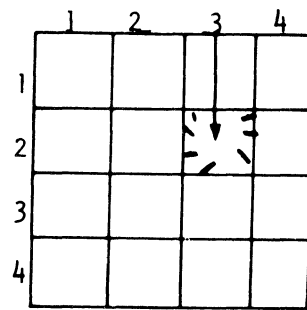
The screen of a computer is divided into squares like this section of graph paper:



Each square can be turned on or off. Turning squares on or off makes patterns we call computer graphics.

### ● Frame 178

Here's a location 3 across and 2 down:



### ● Frame 179

These two programs will both turn on the same location:

```
1 GR
5 COLOR = 6
10 X = 10
20 Y = 20
30 PLOT X,Y
```

```
1 GR
5 COLOR = 6
10 PLOT 10, 20
```

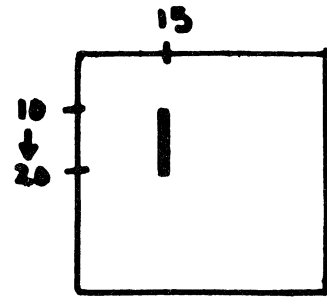
↑ Can be used with different values for X and Y

### ● Frame 180

```
1 GR
5 FOR C = 0 TO 15
10 COLOR = C
20 PLOT 20, 20
30 NEXT C
```

● FRAME 181

The VLIN command draws a vertical line.  
The first 2 numbers show the end points  
of the line. The AT shows where the line  
is positioned across the screen.

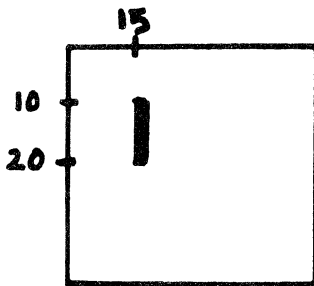


VLIN 10,20 AT 15

● FRAME 182

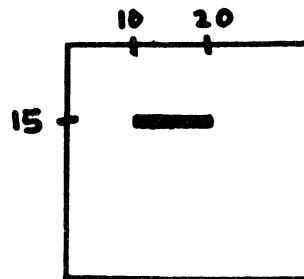
Vertical Line

VLIN 10,20 AT 15



Horizontal Line

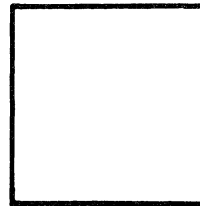
HLIN 10,20 AT 15



● FRAME 183

```
10 GR
15 COLOR=8
20 FOR X=10 TO 20
30 VLIN 5,25 AT X
40 NEXT X
```

This program draws vertical  
lines to make a solid block.



● FRAME 184

Program 1

```
1 GR
5 COLOR=8
10 FOR X=0 TO 39
20 VLIN 0,39 AT X
30 NEXT X
40 FOR T=1 TO 1000
50 NEXT T
60 TEXT
70 HOME
80 GOTO 1
```

Program 2

```
1 GR
5 COLOR=8
10 FOR X=0 TO 39
20 HLIN 0,39 AT X
30 NEXT X
40 FOR T=1 TO 1000
50 NEXT T
60 TEXT
70 HOME
80 GOTO 1
```

timing loop

these two statements  
get rid of graphics  
and clear screen

●FRAME 185

Enter this program and try various values of X and Y to see where the locations are.

```
5 GR
10 COLOR=15
15 PRINT "TYPE VALUES FOR X & Y WITH A COMMA IN BETWEEN:"
20 INPUT X,Y
30 PLOT X,Y
40 PRINT "PRESS RETURN TO CONTINUE."
50 INPUT A$
60 GOTO 15
```

●Frame 186

Enter this program and run it to see what happens:

```
5 GR
10 COLOR=15
20 X=INT (40*RND (1))
30 Y=INT (40*RND (1))
40 PLOT X,Y
50 GOTO 20
```

Change line 10 to the following and try to predict what will happen:

```
10 COLOR=INT (16*RND (1))
```

●Frame 187

```
1 REM PROGRAM TO DRAW HORIZONTAL LINES
5 GR
10 FOR C=1 TO 15
20 COLOR=C
30 FOR X=0 TO 39 STEP 3
40 HLIN 0,39 AT X
50 NEXT X
60 NEXT C
70 END
```

Change the above program to draw vertical lines.

Then change the program to draw the lines and erase them before drawing new ones. (Hint: COLOR=0 turns a location "off".)

If you add the following line to the above program, you will make a grid:

```
45 VLIN 0,39 AT X
```

● Frames 188-189

```
1 REM PROGRAM TO DRAW RECTANGLE
10 GR
15 COLOR=13
20 FOR X=0 TO 39
30 HLIN 0,39 AT X
35 NEXT X
40 REM NOW WE DRAW THE FRAME
45 COLOR=2
50 HLIN 0,39 AT 0
55 HLIN 0,39 AT 39
60 VLIN 0,39 AT 0
65 VLIN 0,39 AT 39
```

Run the above program. Then add the following.

```
70 REM MAKES MOVING DOT BY TURNING LOCATION
71 REM ON AND OFF
75 FOR X=0 TO 39
80 COLOR=0
90 PLOT X,20
100 REM TIMING LOOP
105 FOR D=1 TO 50
110 NEXT D
120 COLOR=13
130 PLOT X,20
140 NEXT X
150 REM 80 AND 90 TURN DOT OFF
151 REM 120 AND 130 TURN IT BACK ON
```

Run the above, and then add the following.

```
150 REM DOT BOUNCES BACK
155 FOR X=39 TO 0 STEP -1
160 COLOR=0
170 PLOT X,0
180 FOR D=1 TO 50
190 NEXT D
200 COLOR=13
210 PLOT X,20
220 NEXT X
```



● LESSON 10, PART 1

Frame 190

In the second program here, we put 3 statements on one line.  
Both of these programs produce the same result.

Program 1

```
10 FOR A=1 TO 3
20 PRINT "*"
30 NEXT A
```

Program 2

```
10 FOR A=1 TO 3: PRINT "*": NEXT A
```

● Frame 191

```
10 PRINT "ARE YOU 18 OR OVER? 1=YES, 0=NO"
20 INPUT A: IF A=1 THEN 30: IF A=0 THEN 40
30 PRINT "YOU'RE OLD ENOUGH TO VOTE." GOTO 50
40 PRINT "YOU CAN'T VOTE."
50 END
```

RUN

```
ARE YOU 18 OR OVER? 1=YES, 0=NO
?0 ← (input)
YOU'RE OLD ENOUGH TO VOTE.
```

} computer prints

● Frame 192

In this program, the arrow shows the path the computer will follow if the test A=1 is not met. If A does equal 1, then the computer executes the rest of the line.

```
105 REM ANSWER CHECK ROUTINE
110 INPUT A: IF A=1 PRINT "CORRECT":GOTO 130
120 PRINT "WRONG"
130 END
```

● Frame 193

Program 1

```
10 READ A,B: DATA 1,2,3,4
20 PRINT A*B: IF B=4 THEN STOP
30 GOTO 10
```

Program 2

```
10 PRINT "LIST 5 NAMES."
20 C=0: INPUT A: PRINT A
30 IF C=5 GOTO 50
40 GOTO 20
50 END
```

● Frame 194

The computer will accept ? as an abbreviation for PRINT. When the program is listed or run, the ? will be converted into the word PRINT.

```
10 ? "HELLO"
```

```
RUN
```

```
HELLO
```

```
LIST
```

```
10 PRINT "HELLO"
```

● Frame 195

The computer has a command that will tell you how much memory you have left to load in and run your program. The command is: FRE (0). It is used with a PRINT command.

```
PRINT FRE (0)
```

```
3500
```

The 3500 that the computer replies with from the above command means that you have enough room for about 3500 more characters. If you have only a few units of memory left after entering your program, the computer may not have enough room to actually run the program.

● Frames 196 and 197

Enter the following program "as is", and then check how much memory you have left using the FRE (0) command. Then see how much shorter you can make it using multiple statement lines.

```
5 PRINT "NUMBERS AND SQUARES"
10 GOSUB 1000
20 READ Y
30 PRINT Y
40 IF Y=10 THEN 200
50 REM COMPUTE SQUARES
60 GOSUB 1000
70 PRINT "SQUARES TO:";
80 PRINT Y*Y
90 GOTO 20
200 PRINT "THAT'S IT."
210 END
1000 PRINT
1010 PRINT "*****"
1020 RETURN
2000 DATA 1,2,3,4,5,6,7,8,9,10
```

LESSON 10, PART 2

●Frame 198

Here's how we translate a number from scientific notation to regular notation.

3E+07	←	(3 is really 3.)
300000000.	←	(We move the decimal point 7 places and add zeros.)
30,000,000	←	(We can also add commas.)

●Frame 199

1.72E+06	(Here there are some numbers after the decimal point already.)
▼	
172	(Here we've moved 2 places, there are 4 to go.)
▼	
1720000.	
▼	
1,720,000	

●Frame 200

3E-04	(Again, 3 is really 3.)
▼	
.3	(Here we've moved 1 place to the left. There are 3 more to go.)
▼	
.0003	(We moved 3 more places and added zeros.)

●Frame 201

Enter this program. Then try the numbers below as input and see how the computer responds.

```
10 PRINT "ENTER A NUMBER.": INPUT A
20 PRINT A
30 GOTO 10
```

Sample input:

1000000	.1234567
1,000,000	.01234567
2111000	.001234567
.01	1.00345
.003	333333

Try other numbers.

Then change the program so you input 2 numbers and multiply them. See when you get scientific notation for numbers.

Then divide the two numbers. Try to get answers in scientific notation.

WRITTEN QUESTIONS FOR LESSONS 1-5 QUIZ

23. Write a program that will print from 1 to 10 and end the program when it hits 10.
  
  
  
  
  
  
  
  
  
  
24. Write a statement that will produce a random number from 1 to 100.
  
  
  
  
  
  
  
  
  
  
25. Write a statement that will put -7.3 in a box called A.

(answers on reverse)

## ANSWERS FOR LESSONS 1-5 QUIZ

23. Here is the most elementary way to answer the question.

```
10 REM PROGRAM TO COUNT TO 10 AND STOP
20 LET X = 0
30 LET X = X + 1
40 PRINT X
50 IF X = 10 THEN STOP
60 GOTO 30
```

24. 10 PRINT RND(100) or 10 PRINT INT(RND(0)\*100) + 1

25. 10 LET A = -7.3 or 10 A = -7.3

WRITTEN QUESTIONS FOR LESSON 6-10 QUIZ

11. Write a statement that will put the following string in B\$:

YOU WERE THERE

12. Write a program that will input your name and then print it out as follows:

? JOHN  
HELLO JOHN!

13. Write a program using a FOR-NEXT loop that will count backwards from 10 to 1 and print out the results.

14. Write a program that will count from 1 to 5 using READ...DATA statements.

15. Write a program that will assign the value of 1 to each variable in an array from A(1) to A(5).

## ANSWERS TO LESSONS 6-10 QUIZ

11. 10 LET B\$ = "YOU WERE THERE"

or

10 BS = "YOU WERE THERE"

12. 10 INPUT N\$

20 PRINT "HELLO" ;N\$;"'"

13. 10 FOR X = 10 TO 1 STEP -1

20 PRINT X

30 NEXT X

40 END

14. There are several ways to do this--here is the way we did it:

10 DATA 1, 2, 3, 4, 5

20 READ C

30 PRINT C

40 IF C = 5 THEN STOP

50 GOTO 20

15. 10 FOR X = 1 TO 5

20 LET A(X) = 1

30 NEXT X



## PROGRESS CHART

[illegible]

