

8

Further Programming in BASIC



Last year you learnt how to write programs using computer language in BASIC. Let us recollect the statements in BASIC.



What we have already learnt

1. PRINT Command
2. INPUT Command
3. Let Command
4. CLS Command
5. IF... Then usage
6. IF... Then ... Else usage
7. FOR... NEXT Loop usage
8. SCREEN Command
9. LINE Command
10. CIRCLE Command
11. PSET Command
12. DIM Command
13. READ... DATA Command
14. SOUND Command

15. ARRAYS Command

Using these command we have written following types of programs.

1. To draw picture using circles and lines.
2. To find the smallest and largest number in a given set of numbers.
3. To find the sum of given numbers.
4. To sort a given set of numbers (sorting).
5. To create sound as desired.

In this chapter we will discuss some interesting programs in BASIC.

Fortune Teller

It will be interesting to write a program that predicts the fortune of the person, when his date of birth is given.

What will be the input to this program?

The date, month and year in which a person is born are the input to our program. These three values can be given as integers. If it is January – 1, February – 2 and so on, the months can be given as integers.

What should be the output of the program?

If the fortune shown is the same for all given dates of birth, then no one will take your program seriously? So, our program should be able to give different fortunes. For example:

- (i) You will get very good marks for the S.S.L.C Exam.
- (ii) You should work harder.
- (iii) You will have a very important visitor next sunday!
- (iv) You should eat more!

Let these be the output of our program. Select a fortune according to the date of birth and show it on screen.

How can the statement be selected?

As soon as you get the input add up the month, year and date. Divide the resultant with the number four (4) and find out the remainder. When we divide any number with 4 the possible remainders are 0, 1, 2 or 3. If the remainder is '0' print the first statement and so on.

The program is given below.

```
10  CLS
20  PRINT "ENTER THE DAY OF YOUR BIRTH"
30  INPUT D
40  PRINT "ENTER THE MONTH OF YOUR BIRTH"
50  INPUT M
60  PRINT "ENTER THE YEAR OF YOUR BIRTH"
70  INPUT Y
80  LET TOTAL = D + M + Y
90  LET R = TOTAL MOD 4
100 IF R = 0 THEN
110 PRINT "YOU WILL GET VERY GOOD MARKS FOR
      THE S.S.L.C EXAM"
```



To find the remainder value when the number A is divided by B, we use MOD keyword that is $A \text{ MOD } B$.

```

120 END IF
130 IF R = 1 THEN
140 PRINT "YOU SHOULD WORK
      HARDER"
150 END IF
160 IF R = 2 THEN
170 PRINT "YOU WILL HAVE A VERY
      IMPORTANT VISITOR NEXT SUNDAY!"

180 END IF
190 IF R = 3 THEN
200 PRINT "YOU SHOULD EAT MORE!"
210 END IF
220 END

```

In this program we are storing the day value in D, month value in M and year value in Y. We are adding these values and storing the result in the variable TOTAL (that is statement 80). We are dividing the total with 4 (four) storing the remainder value in the variable R (that is statement 90). We are using the MOD operator to do it. From the statement 100 we are selecting the fortune lines according to the R value obtained.

Multiplication Table in BASIC

Last year we did many small programs using loops. Let us see how to write a multiplication program of 5 using loops.

We are going to write the program for printing the first 20 lines of the multiplication table for 5.

1	× 5 =	5
2	× 5 =	10
3	× 5 =	15
4	× 5 =	20
5	× 5 =	25
6	× 5 =	30

If you observe the above table you can see the following facts. The value in the first column shows the number of rows being printed. In the first row it is 1 and in second row it is 2 and so on. If it is the I^{th} row then the value of I will be present in the first column. In the next

column $\times 5 =$ should be printed. What about third column? It is the product of the first column and 5. That is in the first row, the value of 3rd column should be 5, in second row it should be 10 and so on. So in the Ith row the value will be $I \times 5$. By studying these facts and summarising we get a logic to print the Ith line of multiplication table. That is PRINT I; " $\times 5 =$ "; $I * 5$.

By using this print statement in the FOR loop, with the value of I varies from 1 to 20, we will get this required table.

Now let us write the program:

```
10  CLS
20  FOR I = 1 TO 20
30  PRINT I; "X5 = "; I*5
40  NEXT I
50  END
```

Type and run this program. You will get the multiplication table for 5.

The above program will give you only the multiplication table for 5. We are going to write a program to get the multiplication table for any number, that is, if the computer accepts an integer value, then the result should be the multiplication table of that integer.

If we write some multiplication table, we will be able to find a general rule.

1	\times	2	=	2
2	\times	2	=	4
3	\times	2	=	6
4	\times	2	=	8
5	\times	2	=	10

1	\times	5	=	5
2	\times	5	=	10
3	\times	5	=	15
4	\times	5	=	20
5	\times	5	=	25

1	\times	7	=	7
2	\times	7	=	14
3	\times	7	=	21
4	\times	7	=	28
5	\times	7	=	35

Here, the tables are printed in 5 rows. The first column specifies the number of row. In the second column the ' \times ' multiplication symbol is printed for the multiplication table of 2, the third column is number 2. It is 5 in 5th multiplication table. So in n's multiplication table the third column will be n. The fourth column is always '='. In the fifth column the product of the first column and third column is printed.



Now find out the changes, which should be brought in multiplication table for 5 in line 40.

```
PRINT I; " X "; N; " = "; I*N
```

Let us write the program:

```
10 PRINT "ENTER A NUMBER"
20 INPUT N
30 FOR I = 1 TO 20
40 PRINT I; "X", N; "="; I * N
50 NEXT I
60 END
```

Strings

The data that we have to handle are not always integers. For example: names, place names etc are collection of characters. These characters are called BASIC STRINGS. When we refer the value of strings it should be written inside the double quotes ("....."). For example "Hello", "Orissa", "ABCD" etc. The variables which store the strings are called string variables. The variable names for strings end with a (dollar symbol). For example A\$, K\$, My Name\$ etc. Let us see a small program below:

```
10 CLS
20 A$ = "GOOD"
30 B$ = "MORNING"
40 C$ = A$ + B$
50 PRINT C$
60 END
```

The A\$ variable holds the string "GOOD" in the 2nd statement. In the 3rd statement B\$ variable holds the string "MORNING". In the fourth statement we are putting both these strings together and storing it into C\$. Fifth statement displays the value of C\$.

Let us write a program to accept our name from the key board and add "Hello" to it.

```
10 PRINT "ENTER YOUR NAME"
20 INPUT A$
30 B$ = "Hello" + A$
```

```

40 PRINT B$
50 END

```

The first two statements accept the name of the person. Since name is a string it should be stored in a string only. So we are using variable name. The 3rd statement adds up the string 'Hello' before the given name. That is, if the name given is 'RAJU' then after the statement 30 (that is 3rd statement) the B\$ variable will have 'Hello RAJU'. The next statement displays this string on the monitor.

String Functions

To find the length of a string, separate a part of the string from the given string etc. We make use of string functions. Let us see some important functions:

1. LEN ()

Assume a situation to find the longest name of the students studying in your class. We can use LEN () function in such situations. The length of the string, the number of characters and spaces in the string) is found using this function. For example:

```

10 A$ = "BASIC PROGRAM"
20 L = LEN(A$)
30 PRINT L

```

Find out the output of the program by analyzing the above given code. The length of the string stored in the string variable A\$ will be 13.

2. LEFT\$ ()

To break up the left part of a given string this function is used. See the example given below:

```

10 A$ = "BASIC PROGRAM"
20 B$ = LEFT$(A$, 5)
30 PRINT B$

```

Here notice the values given inside the LEFT\$() functions. Firstly the string variable (A\$) from which the string should be extracted is specified after this comma (,), the next is the number of characters from the string. So within the LEFT\$() function the string variable



name and the number of characters to be separated should be specified. The output of the above program will be BASIC.

3. RIGHT\$()

To breakup the right part of a given string we use this function.

```
10  A$ = "BASIC PROGRAM"
20  B$ = RIGHT$(A$, 7)
30  PRINT B$
```

The output of the program will be "PROGRAM."

4. MID\$()

To find out the mid position of the given string, we use this function. See the program give below:

```
10  A$ = "PROGRAM IN BASIC"
20  B$ = MID$(A$, 9, 2)
30  PRINT B$
```

The second statement consists of three values inside the MID\$(). The first value is the strings name, the integer 9 is the starting position from where the string should be separated and the integer 2 is the number of characters to be separated out that is MID\$(A\$, 9, 2) specifies that from string values stored in A\$, 2 characters starting from 9th character of the string be separated. Then what will be the output of the above program.

Concentrate on the program given below.

```
10  A$ = "INDIA IS MY COUNTRY"
20  B$ = LEFT$(A$, 5)
30  C$ = RIGHT$(A$, 11)
40  D$ = C$ + " IS " + B$
50  PRINT D$
```

- In this program we are storing the string "India is my Country" in the variable string.
- In the statement 20, we are breaking 5 characters on the left of A\$. "India", in the string variable B\$.
- In the statement 30 we are storing last 11 characters of A\$, "My country" in the string variable C\$.

- Now you can imagine the value of D\$ if you might have understood the output of the program.

Counting Characters

Let us write a program to find the number of occurrences, of a particular character in a given string. Suppose you are interested in counting the occurrences of character 'E'.

```

10 INPUT A$
20 L = LEN(A$)
30 N = 0
40 FOR I = 1 TO L
50 IF MID$(A$, I, 1) = "E" THEN N = N + 1
60 NEXT I
70 PRINT "NUMBER OF E'S ="; N
80 END

```

In the above program, at statement number 10 we are storing a string given by the user in string variable A\$. In the next statement the length of the given string is stored in the variable 'L'.

In the for loop (given statement number 40 – 60) we are checking characters of the given string one by one. If it happens to be character 'E' value of N is incremented by one. (i.e., N is the count variable). In this way we can find the number of times the character 'E' occurred. The value store in variable 'N' will be the occurrence of the character 'E' in the string A\$.

The number of occurrence of character 'E' is printed in the statement 70.

The above program finds out the occurrence of the character 'E'. What changes have to be made to find the occurrence of the character 'B' in the above program. Make the change and find out whether you are right.



To concatenate strings

To concatenate two different strings we use '+' symbol. That is, if A\$ = "Ammu eats" and B\$ = "Mango" then C\$ = A\$ + B\$ will be "Ammu eats mango."



Music through BASIC

We have learnt to play our national anthem in 8th standard itself. You can create and play music in basic, as you desire. The basic of all the music is Sa, Re, Ga, Ma, Pa, Tha, Ne. In BASIC language these base sounds can be specified by letters C, D, E, F, G, A and B. To get the sound, with these letters, we use PLAY command. That is if the sound Sa has to be played, then the command should be PLAY "C", commands for all other base sounds are given below.

<i>Base Sound</i>	<i>Command</i>
Sa	PLAY "C"
Re	PLAY "D"
Ga	PLAY "E"
Ma	PLAY "F"
Pa	PLAY "G"
Tha	PLAY "A"
Ne	PLAY "B"

Now notice the program given below:

```

10  PLAY "C"
20  PLAY "D"
30  PLAY "E"
40  PLAY "F"
50  PLAY "G"
60  PLAY "A"
70  PLAY "B"

```

When you run the program, the music will be created.

Instead of writing all the statement one below the other, we can also write the commands in one single statement itself, that is PLAY "CDEFGAB". Try out the program in both ways.

Do you feel that while running this program, the music is very fast? If a time break is given to the statements then, we can hear a better music. For this a small alteration can be is done in our program.

```

10  PLAY "L2 CDEFGAB"

```

The command statement is altered by adding a new keyword L2. This is used for time break between each of these base sounds. The time break is specified by the integer which we add to the “L”, in this program we have given L2. Check the program by giving L3, L4, L5, L6, L7, L8 instead of L2. As the integer value increases the duration of the music played will also increase. Type the program given below and run it.

```
10 PLAY "L2 CDE L3 FGAB"
20 PLAY "L5 CD L2 EF L1 GA"
```

In this way try out of the other sounds also on your own.

More activities for you

- Predict the output of the following program.
 - ```
10 A$ = "12"
20 B$ = "34"
30 C$ = A$ + B$
40 PRINT C$
50 END
```
  - ```
10 A$ = "WELCOME"
20 I = LEN (A$)
30 FOR I = 1 TO L
40 B$ = LEFT$ (A$, I)
50 PRINT B$
60 NEXT I
70 END
```
- Write a program to find the factorial of a given number ‘N’ value. (For example if the N value is entered as 5, the output should be calculated as 1 x 2 x 3 x 4 x 5, the output will be 120.)
- Write a program to find the electricity charge used at your home and display the bill on the screen. (The rates are given below.)

Unit	Rate/Unit*
1 - 99	Rs. 1.25
100 - 199	Rs. 1.50
200 - More	Rs. 2.00

* Rate is dummy.

4. Write a program to find out the age of a person on 31st March of 2005. The input of the program should be date of birth (that is date, month and year he was born) of the person. The date, month and year should be stored in different variables. The output of the program should be..... years, Months and days.

A Computer Game

You might have played many games in computers. An interesting game created using the BASIC language is given below. You can type the code, run the program and play the game in your computer. The code is given below and the way to play the game has also been given, most of commands used in this program will be familiar you, See the note for the unfamiliar commands.

```

10 REM ===== PING PONG GAME
   =====
20  REM ==== MAKE SURE THAT THE NUMLOCK IS ON
   =====
30  REM ==USE LEFT ARROW  KEY IN THE NUMBER PAD  FOR
   LEFT MOVEMENT ==
40  REM  USE RIGHT ARROW KEY IN THE NUMBER PAD FOR THE
   RIGHT MOVEMENT
45  REM =====  PRESS ESC KEY TO END THE GAME
   =====
50  SCREEN 2
60  BL = 10: BR = 600:    BT = 10: BB = 185
70  CR = 4:  CX = 300: CY = 20: CXOLD = CX: CYOLD = CY
80  LW = 70: LH = 2: LL = 300 - 50: LT = 190
90  LARROW = 52: RARROW = 54
100 INCR = 15: BINCR = .01
110 XSIGN = 1: YSIGN = 1
120 BSPEED = 5000
130 LINE (LL, LT) - (LL + LW, LT + LH), , BF
140 CIRCLE (CX, CY), CR, 1
150 PAINT (CX, CY), 1, 1
160 LINE (3, BT - CR - 1) - (BR + CR + 4, BT - CR - 1), 1
170 LINE (BL - CR - 4, BT - 5) - (BL - CR - 4, BB), 1
180 LINE (BR + CR + 4, BT - 5) - (BR + CR + 4, BB + 5), 1

```

```

190 X$ = INKEY$
200 IF X$ = "" THEN GOTO 190
205 REM
      =====
210 X$ = INKEY$
220 IF X$ <> "" THEN
230     IF ASC(X$) = 27 THEN SCREEN 0: END
240     LLN = LL
250     IF ASC(X$) = LARROW THEN
260         LLN = LL - INCR
270         IF LLN < BL THEN LLN = BL
280     END IF
290     IF ASC(X$) = RARROW THEN
300         LLN = LL + INCR
310         IF LLN > BR - LW THEN LLN = BR - LW
320     END IF
330     IF LLN <> LL THEN
340         LINE (LL, LT) - (LL + LW, LT + LH), 0, BF
350         LL = LLN
360         LINE (LL, LT) - (LL + LW, LT + LH), 1, BF
370     END IF
380 END IF
390 CXOLD = CX
400 CYOLD = CY
410 CX = CX + XSIGN * BINCR
420 CY = CY + YSIGN * BINCR
430 IF CX < BL OR CX > BR THEN
440     CX = CX - XSIGN * BINCR
450     XSIGN = XSIGN * -1
460     PLAY "L9 C"
470     LINE (3, BT - CR - 1) - (BR + CR + 4, BT - CR - 1), 1
480     LINE (BL - CR - 4, BT - 5) - (BL - CR - 4, BB), 1
490     LINE (BR + CR + 4, BT - 5) - (BR + CR + 4, BB + 5), 1
500 END IF
510 IF CY < BT THEN
520     CY = CY - YSIGN * BINCR
530     YSIGN = YSIGN * -1
540     PLAY "L9 C"

```

```

550     LINE (3,BT-CR-1)-(BR + CR + 4, BT - CR - 1), 1
560     LINE (BL - CR- 4, BT - 5)-(BL - CR - 4, BB), 1
570     LINE (BR + CR+4,BT-5)-(BR + CR + 4, BB + 5), 1
580     END IF
590     IF CY >= LT THEN
600         IF CX >= LL AND CX <= LL + LW THEN
610             CY = BB - BINCR
620             YSIGN = YSIGN * -1
630             PLAY "L9 C"
640         ELSE
650             CLS
660             PLAY "L5 G..."
670     PRINT : PRINT : PRINT : PRINT
        " ...GAME OVER ... "
680         END
690     END IF
700     END IF
710     CIRCLE (CXOLD, CYOLD), 0
720     PAINT (CXOLD, CYOLD), 0, 0
730     CIRCLE (CX, CY), CR, 1
740     PAINT (CX, CY), 1, 1
750     LINE (LL, LT)-(LL + LW, LT + LH), 1, BF
760     CXPREV = CX
770     GOTO 210

```

Instruction

This game consists of a ball and a hitting bar. Using the bar hit the ball before it touches the ground. To move the bar towards left and right use the left arrow key and right arrow key in your key board. Num Lock key should be ON.

Note

REM Command

This command is used to give a view of the program. The statements in the REM command will not be compiled by computer. The instruction about the program is given in this command.



INKEY\$

Do you remember the operation performed by the INPUT command? When we use this command after typing the value, the enter key should be pressed. But when we use INKEY\$ without the enter key, the pressed key value is stored into the variable.



What we have learnt? A self assessment

1. What are the needs of strings in the BASIC program.
2. What are the uses of the string functions given below?
 - LEN ()
 - LEFT\$ ()
 - RIGHT\$ ()
 - MID\$ ()
3. What are the operations that can be performed using the PLAY command in BASIC program?
4. Have you studied an equivalent command to PLAY command in your previous classes?

