

## **Strings**

There are certain types of data (or information) called "strings."

Strings contain a sequence of characters (letters, numbers, and symbols) enclosed in quotation marks.

For example, "Hello World!" is a string.

The following are also strings: "0123456789"

"This is a string"

"abc123"

"1 + 1 = 2"

"!@#\$\$%^&\*()"

Numeric value and string can be interchanged provided the characters are numeric.

**VAL** – It converts a string to a numeric value

SYNTAX: Y = VAL(A\$)

where, A\$ is a string variable with numeric characters

Example,   Let A\$ = "80"  
          Let Y = VAL(A\$)  
          Print Y

Run ↵  
80

**STR\$** - It converts numerical value to a string.

SYNTAX: Y\$ = STR\$(A)

where, A is a numeric variable.

**ASC** – It gives the ASCII code of the first character of the string.

SYNTAX: Y = ASC(A\$)

**CHR\$** = It gives the character represented by the ASCII code N.

SYNTAX: CHR\$(N)

The number of characters present in a string is called its length. It includes blank space as well as special characters. The length of a string can be determined by giving the command, LEN(A\$).

Example, 10 LET A\$ = "I am a teacher"

20 Y = LEN(A\$)

30 PRINT Y

40 END

RUN ↵

Strings can be manipulated to get a new string which is a part of the given string. Commonly used commands are:

**Left\$ (A\$, n)** – This command takes first n characters starting from left side of the string A\$.

**Right\$ (A\$, n)** – This command takes last n characters of a string A\$.

**Mid\$ (A\$, m, n)** – This command takes characters from the middle of the string A\$ starting from the m<sup>th</sup> character up to n<sup>th</sup> character.

### Comparison of Strings

When two strings are to be compared, they are compared by the comparing the ASCII codes of the two strings, one at a time. The string with a greater value of the ASCII code is the greater string. But if the first characters are same, then the program compares the second character of both the strings.

Example, A\$ = "RAM" , B\$ = "RAN"

Then, A\$ < B\$

**Write a program in BASIC to print the given name as initial of the first name and surname. Also print the name in a column.**

```
10 REM TO PRINT NAME AND SURNAME
20 INPUT "NAME , SURNAME"; A$, B$
30 S$ = LEFT$ (A$, 1)
40 PRINT S$ ; " . " ; B$
50 L = LEN(A$) : M = LEN(B$)
60 For I = 1 to L
70 X$ = MID$ (A$ , I , 1)
80 PRINT X$
90 NEXT I
100 FOR K = 1 to M
110 Y$ = MID$ (B$ , K , 1)
120 PRINT Y$
130 NEXT K
140 END
```

**RUN ↵**

NAME , SURNAME ? DIVYA, GUPTA

D . Gupta

D

I  
V  
Y  
A  
G  
U  
P  
T  
A

### **Word Unscrambling**

This is done to rearrange a group of letters to form all the possible words. For example, we are given any four letters, M, O, S, T. We have to find out all the possible four letter words which can be made by rearranging these four letters, like MOST, OSMT and so on.

For this, we will place all the four letters in a list called as P\$ and each letter in this list is represented by P\$(1), P\$(2), .....

Let I1 = subscript of the first letter to be printed, i.e., first position in a word.

I2 = subscript of the second letter to be printed, i.e., second position in a word.

I3 = subscript of the third letter to be printed, i.e., third position in a word and so on.

Example, if I1 = 3, I2 = 2, I3 = 4 and I4 = 1, then the order of printing of the letters is in the order,

P\$(3) P\$(2) P\$(4) P\$(1)

Where, P\$(1), P\$(2), P\$(3), P\$(4) are the letters in a list.

However, printing in this form has a limitation that the same letter should not occupy the two places in the same word. Here, three FOR-TO-NEXT nested loops are there, where outermost loop is for the assignment of value to I1, next loop for I2 (with the restriction  $I2 \neq I1$ ) and the innermost loop will assign value to I3 (with the restriction  $I3 \neq I1$  &  $I3 \neq I2$ ).

For the value of I4, the formula used is  $I4 = 10 - (I1 + I2 + I3)$  (since the sum of first four integers is 10) or make a different loop for I4.

### **Word Unscrambling**

```

10 REM Word Unscrambling

20 PRINT "Type any four letters"

30 PRINT

40 INPUT P$(1), P$(2), P$(3), P$(4)

50 FOR I1 = 1 TO 4

60 FOR I2 = 1 TO 4

70 IF I2 = I1 THEN 180

80 FOR I3 = 1 TO 4

90 IF I3 = I1 THEN 170

100 IF I3 = I2 THEN 170

110 FOR I4 = 1 TO 4

120 IF I4 = I1 THEN 160

130 IF I4 = I2 THEN 160

140 IF I4 = I3 THEN 160

150 PRINT P$(I1); P$(I2); P$(I3); P$(I4)

160 NEXT I4

170 NEXT I3

180 NEXT I2

190 NEXT I1

198 END

```

**RUN ↵**

Type any four letters

? p,o,s,t

psto

ptos

ptso

opst

opts

ospt

ostp

otps

spot

spto

sopt

sotp

stpo

stop

tpos

tpso

tops

tosp

tspo

tsop