

STRINGS IN C PROGRAMMING

A string is a group of characters(*char* type of data) terminated by the NULL character. ('\0'). Whenever a program in C does an operation using a string like comparing it, printing it or copying it to another string, it does it until it encounters a NULL which is nothing but the ASCII code zero ('\0').

An array is a series of homogenous data that are of the same type (as discussed in Chapter 6). It can be an array of integers, or it can be of complex data types like structures or pointers. The string is a simple array of *char* data type. Let us look at a simple example for storing and displaying a string.

```
#include <stdio.h>
main ()
{
    char name[4+1];
    name[0] = 'J';
    name[1] = 'o';
    name[2] = 'h';
    name[3] = 'n';
    name[4] = '\0';
    printf("The name is %s", name);
    printf("Part of the name is %s", &name[1]);
}
```

First we start by declaring the size of the string. We choose a string of size 4 characters. You have to remember that in C the array index starts from 0 and not 1. The size of the array or the string cannot be changed or re-defined by a programmer. In the above program the characters of the string will be numbered from 0 to 3. Then we start initialising the string with the data. Finally we initialize the last statement with zero indicating that it is the end of the string. Alternately we can use a #define statement to define NULL as '\0' and use NULL in the program. Next we print the string using '%s' in our printf() statement. Note that we just have to pass on the name of the array and not the subscript (since, you remember from chapter 6 that the name of the array is also its address). The last statement shows how one can output part of a string, and in that case, observe that the character "&" is necessary in front of the name because we pass the address of the first character.

Note : It is possible for a string to be empty. A string may have no characters in it. An empty string is a character array with the NULL character in the zeroth index position.

Single characters can be replaced in a string. Following are some of the different possible ways in which we can achieve this :

```
#include <stdio.h>
main ( )
```

```
{  
    char str[9+1] = "but";  
  
    char ch = 'i';  
  
    char Surname[9+1] = "pat";
```

//Replacing a single character using a char variable:

```
    Surname[1] = ch;
```

This would result in Surname containing "pit" instead of "pat".

//Replacing a single character using a char literal:

```
    Surname[1] = 'e';
```

This would result in Surname containing "pet".

//Replacing a single character using a character from a string variable:

```
    Surname [1] = str[1];
```

```
//This would result in Surname containing "bet".
```

```
}
```

Source : <http://www.peoi.org/Courses/Coursesen/cprog/frame9.html>