

A string constant is a set of character that enclosed within the double quotes and is also called a literal. Whenever a string constant is written anywhere in a program it is stored somewhere in a memory as an array of characters terminated by a NULL character ('\0').

Example – “m”

“Tajmahal”

“My age is %d and height is %f\n”

The string constant itself becomes a pointer to the first character in array.

Example-char crr[20]=”Taj mahal”;

1000	1001	1002	1003	1004	1005	1006	1007	1008	1009
T	a	j		M	A	H	a	l	\0

↓  
It is called base address.

### Lecture Note: 13

#### **String library function**

There are several string library functions used to manipulate string and the prototypes for these functions are in header file “string.h”. Several string functions are

#### **strlen()**

This function return the length of the string. i.e. the number of characters in the string excluding the terminating NULL character.

It accepts a single argument which is pointer to the first character of the string.

For example-

```
strlen("suresh");
```

It return the value 6.

**In array version to calculate length:-**

```
int str(char str[])
{
    int i=0;
    while(str[i]!='\0')
    {
        i++;
    }
    return i;
}
```

Example:-

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str[50];
    print("Enter a string:");
```

```
    gets(str);  
    printf("Length of the string is %d\n",strlen(str));  
}
```

Output:

Enter a string: C in Depth

Length of the string is 8

### **strcmp()**

This function is used to compare two strings. If the two strings match, `strcmp()` returns a value 0 otherwise it returns a non-zero value. It compares the strings character by character and the comparison stops when the end of the string is reached or the corresponding characters in the two strings are not the same.

```
strcmp(s1,s2)
```

return a value:

<0 when  $s1 < s2$

=0 when  $s1 = s2$

>0 when  $s1 > s2$

The exact value returned in case of dissimilar strings is not defined. We only know that if  $s1 < s2$  then a negative value will be returned and if  $s1 > s2$  then a positive value will be returned.

For example:

```

/*String comparison.....*/
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[10],str2[10];
    printf("Enter two strings:");
    gets(str1);
    gets(str2);
    if(strcmp(str1,str2)==0)
    {
        printf("String are same\n");
    }
    else
    {
        printf("String are not same\n");
    }
}

```

**strcpy()**

This function is used to copying one string to another string. The function `strcpy(str1,str2)` copies `str2` to `str1` including the NULL character. Here `str2` is the source string and `str1` is the destination string.

The old content of the destination string `str1` are lost. The function returns a pointer to destination string `str1`.

Example:-

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[10],str2[10];
    printf("Enter a string:");
    scanf("%s",str2);
    strcpy(str1,str2);
    printf("First string:%s\t\tSecond string:%s\n",str1,str2);
    strcpy(str,"Delhi");
    strcpy(str2,"Bangalore");
    printf("First string :%s\t\tSecond string:%s",str1,str2);
```

**strcat()**

This function is used to append a copy of a string at the end of the other string. If the first string is ""Purva" and second string is "Belmont" then after using this function the string becomes "PusvaBelmont". The NULL character from str1 is moved and str2 is added at the end of str1. The 2<sup>nd</sup> string str2 remains unaffected. A pointer to the first string str1 is returned by the function.

Example:-

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[20],str[20];
    printf("Enter two strings:");
    gets(str1);
    gets(str2);
    strcat(str1,str2);
    printf("First string:%s\t second string:%s\n",str1,str2);
    strcat(str1,"-one");
    printf("Now first string is %s\n",str1);
}
```

Output

Enter two strings: data

Base

First string: database second string: database

Now first string is: database-one

### Lecture Note: 14

## FUNCTION

A function is a self contained block of codes or sub programs with a set of statements that perform some specific task or coherent task when it is called.

It is something like to hiring a person to do some specific task like, every six months servicing a bike and hand over to it.

Any 'C' program contain at least one function i.e main().

There are basically two types of function those are

### **1. Library function**

### **2. User defined function**

The user defined functions defined by the user according to its requirement

System defined function can't be modified, it can only read and can be used.

These function are supplied with every C compiler

Source of these library function are pre compiled and only object code get used by the user by linking to the code by linker

**Here in system defined function description:**

**Function definition** : predefined, precompiled, stored in the library