

## String functions in C with examples

What are strings?

In simple language STRING'S are nothing but the character array. The declaration of string (character array) is much similar to normal array declaration. Each string is terminated by '\0' as indication of string termination. So obviously you will require an extra byte of memory for storing '\0'. '\0' is single character whose ASCII ([American Standard Code for Information Interchange](#)) value is 0.

How to initialize strings (char array)?

```
char site[]={'g','o','o','g','l','e'};
char Site[]="Programming spark";
```

How to take Input as string and print Output that string?

way 1:

```
void main()
{
    char name[30];
    printf("Enter your name");
    scanf("%s", name);    //format specifier
    printf("%s", name);
}
```

way 2:

```
void main()
{
    char name[30];
    printf("Enter your name");
    gets(name);    //in-built function
    puts(name);
}
```

Difference between both ways is that 1st way use's format specifier and second way uses in-built function.

In 1st way if you Enter something like john karter it will take only john as string and anything after (space) will be discarded. In second way if you enter any space it will be accepted. Output of second way would be john karter.

Similarity between both way is that you don't have to add '\0' character.

scanf(%s ) converts 32(space) or '\n' character into '\0' and gets() converts newline character '\n' into '\0'.

List of (most frequently used) string library functions in c.

strlen strcpy strcat strrev strcmp strcmpistrupr strlwr strncmpi  
strncat strncmp strncmpi strncmpi.

String functions examples

1) int strlen(char array): This function accepts string as parameter and return integer i.e the length of String passed to it.

Example

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char string[]="spark";
    int len;
    len=strlen(string);
    printf("length of %s is %d\t", string, len);
}
```

Output: :length of spark is 5.

Did you notice that strlen() does not include '\n' in string length or else length would be 6.

2) strcpy (Destination string,source string): This function accepts 2 strings as parameter, 1st one is destination string and 2nd is source string. This function copies source string to destination string.

Example

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char src[]="spark",dest[15];
    strcpy(dest,src);
    printf("%s is copied to dest string\t",dest);
}
```

Output: spark is copied to dest string.

3) strcat (Destination string,source string): This function accepts two strings source string is appended to the destination string.

Example

```
#include <stdio.h>
#include <string.h>
```

```

void main(void)
{
    char src[]="spark",dest[]="programming";
    strcat(dest,src);
    printf("concatenated string is %s",dest);
}

```

Output: concatenated string is programmingspark

4) `strrev (string)`:This function accepts single string as parameter and reverse that string.

Example

```

#include <stdio.h>
#include <string.h>
void main(void)
{
    char string[]="spark";
    strrev(string);
    printf("reverse string is %s",string);
}

```

Output: reverse string is krapS.

5) `int strcmp (string 1, string2)`:This function compares two strings passed as parameters and returns either +ve number,0,-ve number.

+ve value indicates  $string1 > string2$ .  
 0 indicates  $string1$  and  $string2$  are equal  
 -ve value indicates  $string1 < string2$ .

Example

```

#include <stdio.h>
#include <string.h>
void main(void)
{
    char string1[]="spark",string2[]="programming";
    int cmp;
    cmp=strcmp(string1,string2);
    if(cmp>0)
        printf("%s > %s",string1,string2);
    else
    {
        if(cmp<0)
            printf("%s < %s",string1,string2);
        else
            printf("%s = %s",string1,string2);
    }
}

```

Output: spark > programming.

.this is because alphabetically p comes first then s.so the string compare function returns difference between ascii of s and p which would be +ve.

6) strcmpi (string 1, string2):This function is similar to strcmp().The only difference is that it ignores the case.example SPARK and spark both are same.

Example

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char string1 []="spark",string2 []="SPArk";
    int cmp;
    cmp=strcmpi(string1,string2);
    if(cmp>0)
        printf("%s > %s",string1,string2);
    else
    {
        if(cmp<0)
            printf("%s < %s",string1,string2);
        else
            printf("%s = %s",string1,string2);
    }
}
```

Output: spark = SPArk.

7) strlwr (string)::This function accepts single string that can be in any case(lower or upper).Itconverts the string in lower case.

Example

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char string1 []="SPArk";
    strlwr(string1);
    printf("%s is in lower case",string1);
}
```

Output: spark is in lower case.

8) `strupr (string)`:: This function accepts single string that can be in any case (lower or upper). It converts the string in upper case.

Example

```
#include <stdio.h>
#include <string.h>
void main(void)
{
    char string1[]="SPArk";
    strupr(string1);
    printf("%s is in upper case",string1);
}
```

Output: SPARK is in upper case.