

C String Functions

There are many important string functions defined in "string.h" library.

No.	Function	Description
1)	<u>strlen(string_name)</u>	returns the length of string name.
2)	<u>strcpy(destination, source)</u>	copies the contents of source string to destination string.
3)	<u>strcat(first_string, second_string)</u>	concat or joins first string with second string. The result of the string is stored in first string.
4)	<u>strcmp(first_string, second_string)</u>	compares the first string with second string. If both strings are same, it returns 0.
5)	<u>strrev(string)</u>	returns reverse string.
6)	<u>strlwr(string)</u>	returns string characters in lowercase.
7)	<u>strupr(string)</u>	returns string characters in uppercase.

C String Length: strlen() function

The strlen() function returns the length of the given string. It doesn't count null character '\0'.

```
#include<stdio.h>
#include <string.h>
int main(){
char ch[20]={'H', 'e', 'l', 'l', 'o', '\0'};
    printf("Length of string is: %d",strlen(ch));
    return 0;
}
```

Output:

```
Length of string is: 5
```

C Copy String: strcpy()

The strcpy(destination, source) function copies the source string in destination.

```
#include<stdio.h>
#include <string.h>
int main(){
    char ch[20]={'H', 'e', 'l', 'l', 'o', '\0'};
    char ch2[20];
    strcpy(ch2,ch);
    printf("Value of second string is: %s",ch2);
    return 0;
}
```

Output:

Value of second string is: Hello

C String Concatenation: strcat()

The strcat(first_string, second_string) function concatenates two strings and result is returned to first_string.

```
#include<stdio.h>
#include <string.h>
int main(){
    char ch[10]={'h', 'e', 'l', 'l', 'o', '\0'};
    char ch2[10]={'c', '\0'};
    strcat(ch,ch2);
    printf("Value of first string is: %s",ch);
    return 0;
}
```

Output:

Value of first string is: helloc

C Compare String: strcmp()

The strcmp(first_string, second_string) function compares two string and returns 0 if both strings are equal.

Here, we are using *gets()* function which reads string from the console.

```
#include<stdio.h>
#include <string.h>
int main(){
    char str1[20],str2[20];
    printf("Enter 1st string: ");
    gets(str1);//reads string from console
    printf("Enter 2nd string: ");
    gets(str2);
    if(strcmp(str1,str2)==0)
        printf("Strings are equal");
    else
        printf("Strings are not equal");
    return 0;
}
```

Output:

Enter 1st string: Hello

Enter 2nd string: Hello

Strings are equal

Another Output:

Enter 1st string: Hello

Enter 2nd string: Hii

Strings are not equal

C Reverse String: strrev()

The *strrev(string)* function returns reverse of the given string. Let's see a simple example of *strrev()* function.

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

```
int main(){
    char str[20];
    printf("Enter string: ");
    gets(str);//reads string from console
    printf("String is: %s",str);
    printf("\nReverse String is: %s",strrev(str));
    return 0;
}
```

Output:

```
Enter string: Hello
String is: Hello
Reverse String is: olleH
```

C String Lowercase: strlwr()

The strlwr(string) function returns string characters in lowercase. Let's see a simple example of strlwr() function.

```
#include<stdio.h>

#include <string.h>

int main(){
    char str[20];
    printf("Enter string: ");
    gets(str);//reads string from console
    printf("String is: %s",str);
    printf("\nLower String is: %s",strlwr(str));
    return 0;
}
```

Output:

```
Enter string: I Love C Programming
String is: I Love C Programming
Lower String is: i love c programming
```

C String Uppercase: strupr()

The strupr(string) function returns string characters in uppercase. Let's see a simple example of strupr() function.

```
#include<stdio.h>

#include <string.h>

int main(){

    char str[20];

    printf("Enter string: ");

    gets(str);//reads string from console

    printf("String is: %s",str);

    printf("\nUpper String is: %s",strupr(str));

    return 0;

}
```

Output:

```
Enter string: Hello World
String is: Hello World
Upper String is: HELLO WORLD
```