

# Strings in C

Strings are actually one-dimensional array of characters terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the rule of array initialization then you can write the above statement as follows –

```
char greeting[] = "Hello";
```

Following is the memory presentation of the above defined string in C/C++ –

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

Actually, you do not place the null character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array. Let us try to print the above mentioned string –

```
#include <stdio.h>

int main () {

    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
    printf("Greeting message: %s\n", greeting );
    return 0;
}
```

When the above code is compiled and executed, it produces the following result –

**Greeting message: Hello**

C supports a wide range of functions that manipulate null-terminated strings –

Sr.No.	Function & Purpose
1	<b>strcpy(s1, s2);</b> Copies string s2 into string s1.
2	<b>strcat(s1, s2);</b> Concatenates string s2 onto the end of string s1.
3	<b>strlen(s1);</b> Returns the length of string s1.
4	<b>strcmp(s1, s2);</b> Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	<b>strchr(s1, ch);</b> Returns a pointer to the first occurrence of character ch in string s1.
6	<b>strstr(s1, s2);</b> Returns a pointer to the first occurrence of string s2 in string s1.

### Sample Programs based on String Header File(**string.h**)

#### 1. Write a program to accept and display a string.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char a[100];
    clrscr();
    printf("Enter a string\n");
    gets(a);
    printf("The entered string is: %s",a);
    getch();
}
```

**Output:**

Enter a String  
How are you?  
The entered string is: How are you?

**2. Write a program to accept a string and display its length (strlen)**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int l;
    char a[100];
    clrscr();
    printf("Enter a string\n");
    gets(a);
    l=strlen(a);
    printf("The length of the entered string is:%d",l);
    getch();
}
```

**Output:**

Enter a string  
Hello  
The length of the entered string is:5

**3. Write a program to accept a string, copy it into another string and display this new string (strcpy)**

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

void main()
{
    char a[100],b[100];
    clrscr();
    printf("Enter a string\n");
    gets(a);
    strcpy(b,a);
    printf("The new string is:%s",b);
    getch();
}
```

**Output:**

Enter a string  
Hello, how are you?  
The new string is Hello. How are you?

4. **Write a program to accept two strings, compare them and display if they are equal or not. If they are not equal display the one which is greater.**  
**(strcmp)**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char a[100],b[100];
clrscr();
printf("Enter two Strings:\n");
gets(a);
gets(b);
if(strcmp(a,b)==0)
printf("The Strings are equal");
else if(strcmp(a,b)>0)
printf("%s string is greater",a);
else
printf("%s string is greater",b);
getch();
}
```

**Output:**

```
Enter two strings:
Hello
Hi
Hi string is greater
```

5. **Write a program to accept two strings, compare them and display if they are equal or not. If they are not equal display the one which is greater.**  
**(strcat)**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char a[100],b[100];
clrscr();
printf("Enter two strings:\n");
gets(a);
gets(b);
strcat(a,b);
printf("The concatenated string is %s",a);
getch();
}
```

## Programs based on Strings

6. Write a program to accept a string and find its length without using the string header file.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[100];
    int len=0;
    clrscr();
    printf("Enter a string:\n");
    gets(a);
    while(a[len]!='\0');
    {
        len++;
    }
    printf("The length of this string is %d characters.",len);
    getch();
}
```

**Output:**

Enter a string:

Hello

The length of the string is 5 characters

7. Write a program to accept a string and find the number of vowels in it. (**Don't use the string header file**)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[100];
    int i,len=0, count=0;
    clrscr();
    printf("Enter a string:\n");
    gets(a);

    while(a[len]!=0)
    {
        len++;
    }

    for(i=0;i<=len;i++)
    {
        If(a[i]=='a' || a[i]=='e' || a[i]=='i' || a[i]=='o' || a[i]=='u' || a[i]=='A' || a[i]=='E' || a[i]=='I' ||
a[i]=='O' || a[i]=='U' )
        count++;
    }

    printf("The total number of vowels are : %d", count);
    getch();
}
```

8. Write a program to **reverse** a user entered string (**strrev**)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int l;
    char a[100];
    clrscr();
    printf("Enter a String:");
    gets(a);
    strrev(a);
    printf("The reversed string is: %s",a);
    getch();
}
```

9. Write a program to check whether the entered string is a palindrome. (**Do not use the string header file**)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n=0,i;
    char a[100],rev[100];
    clrscr();
    printf("Enter a string:");
    gets(a);

    while(a[n]!='\0')
    {
        n++;
    }

    for(i=0;i<=(n-1);i++)
    {
        rev[n-i-1]=a[i];
    }

    for(i=0;i<=n;i++)
    {
        if(a[i]!=rev[i])
            break;
    }

    if(i==n)
        printf("The string is a palindrome");
    else
        printf("The string is not a palindrome");
    getch();
}
```

10. Write a program to count blank **spaces**, digits, **vowels** and consonants in the string.

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```

char a[100];
int i,len=0,vowels=0,spaces=0,digits=0,consonants=0;
clrscr();
printf("Enter a string:\n");
gets(a);
while(a[len]!=0)
{
len++;
}

for(i=0;i<=len-1;i++)
{
if(a[i]=='a' || a[i]=='e' || a[i]=='i' || a[i]=='o' || a[i]=='u' || a[i]=='A' || a[i]=='E' || a[i]=='I' ||
a[i]=='O' || a[i]=='U')
vowels++;
else
{
If((a[i]>='a' && a[i]<='z')||(a[i]>='A' && a[i]<='Z'))
consonants++;
else
{
if(a[i]>='0' && a[i]<='9')
digits++;
else
{
if(a[i]==' ')
spaces++;
}
}
}
}

printf("The total number of vowels are: %d\nThe total number of spaces are:%d\nThe total
number of digits are:%d\nThe total number of consonants are:%d",vowels, spaces, digits,
consonants);

getch();
}

```