In C programming, a string is a sequence of characters terminated with a null character \0. For example:

char c[] = "c string";

When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character \0 at the end by default.

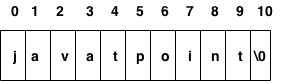
Memory diagram of strings in C programmingMemory Diagram

**How to declare a string?**

Here's how you can declare strings:

There are two ways to declare a string in c language.

1. By char array
2. By string literal
3. **char** ch[10]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};

As we know, array index starts from 0, so it will be represented as in the figure given below. 

4.8M

122

Epic Games Receives $2 Billion in Funding for Metaverse Goals

While declaring string, size is not mandatory. So we can write the above code as given below: **char** ch[]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};

We can also define the **string by the string literal** in C language. For example:

**char** ch[]="javatpoint";

In such case, '\0' will be appended at the end of the string by the compiler.

  Ex. 1

#include <stdio.h>

int main()

{

char name[20];

printf("Enter name: ");

scanf("%s", name);

printf("Your name is %s.", name);

return 0;

}

Ex. 2

#include<stdio.h>

#include <string.h>

**int** main(){

**char** name[50];

printf("Enter your name: ");

gets(name); //reads string from user

printf("Your name is: ");

puts(name);  //displays string

**return** 0;

}

# **C String Length: strlen() function**

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
5. printf("Length of string is: %d",strlen(ch));
6. **return** 0;
7. }

Using the null character

Let's see the same example of counting the number of vowels by using the null character.

#include<stdio.h>

**void** main ()

{

**char** s[11] = "javatpoint";

**int** i = 0;

**int** count = 0;

**while**(s[i] !=’\0’)

   {

**if**(s[i]=='a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'u' || s[i] == 'o')

        {

            count ++;

        }

        i++;

    }

    printf("The number of vowels %d",count);

}

**strcpy(s1, s2);**

Copies string s2 into string s1.

**strcat(s1, s2);**

Concatenates string s2 onto the end of string s1.

**strlen(s1);**

Returns the length of string s1.

**strcmp(s1, s2);**

Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.

**strchr(s1, ch);**

Returns a pointer to the first occurrence of character ch in string s1.

**strstr(s1, s2);**

Returns a pointer to the first occurrence of string s2 in string s1.

# C String Concatenation: strcat()

The strcat(first\_string, second\_string) function concatenates two strings and result is returned to first\_string.

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** ch[10]={'h', 'e', 'l', 'l', 'o', '\0'};
5. **char** ch2[10]={'c', '\0'};
6. strcat(ch,ch2);
7. printf("Value of first string is: %s",ch);
8. **return** 0;
9. }

# C String Length: strlen() function

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

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
5. printf("Length of string is: %d",strlen(ch));
6. **return** 0;
7. }

Output:

# C Reverse String: strrev()

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

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** str[20];
5. printf("Enter string: ");
6. gets(str);//reads string from console
7. printf("String is: %s",str);
8. printf("\nReverse String is: %s",strrev(str));
9. **return** 0;
10. }

# C String Lowercase: strlwr()

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

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** str[20];
5. printf("Enter string: ");
6. gets(str);//reads string from console
7. printf("String is: %s",str);
8. printf("\nLower String is: %s",strlwr(str));
9. **return** 0;
10. }

# 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.

1. #include<stdio.h>
2. #include <string.h>
3. **int** main(){
4. **char** str1[20],str2[20];
5. printf("Enter 1st string: ");
6. gets(str1);//reads string from console
7. printf("Enter 2nd string: ");
8. gets(str2);
9. **if**(strcmp(str1,str2)==0)
10. printf("Strings are equal");
11. **else**
12. printf("Strings are not equal");
13. **return** 0;
14. }