**Strings**

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

"c string"

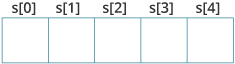
When compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character \0 at the end.

**How to declare a string?**

Before you can work with strings, you need to declare them first. Since string is an array of characters. You declare strings in a similar way like you do with arrays.

Here's how you declare a string:

char s[5];



**How to initialize strings?**

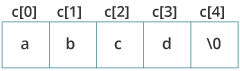
You can initialize strings in a number of ways.

char c[] = "abcd";

char c[50] = "abcd";

char c[] = {'a', 'b', 'c', 'd', '\0'};

char c[5] = {'a', 'b', 'c', 'd', '\0'};



**Read String from the user**

You can use the scanf() function to read a string.

The scanf() function reads the sequence of characters until it encounters a [whitespace](https://stackoverflow.com/questions/30033582/what-is-the-symbol-for-whitespace-in-c)(space, newline, tab etc.).

**Example 1: scanf() to read a string**

#include <stdio.h>

int main()

{

char name[20];

printf("Enter name: ");

scanf("%s", name);

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

return 0;

}

**Output**

Enter name: Dennis Ritchie

Your name is Dennis.

Even though Dennis Ritchie was entered in the above program, only "Ritchie" was stored in the name string. It's because there was a space after Ritche.

**How to read a line of text?**

You can use gets() function to read a line of string. And, you can use puts() to display the string.

**Example 2: gets() and puts()**

#include <stdio.h>

int main()

{

char name[30];

printf("Enter name: ");

gets(name); // read string

printf("Name: ");

puts(name); // display string

return 0;

}

When you run the program, the output will be:

Enter name: Tom Hanks

Name: Tom Hanks

**Passing Strings to Function**

Strings can be passed to a function in a similar way as arrays. Learn more about [passing array to a function](https://www.programiz.com/c-programming/c-arrays-functions).

**Example 3: Passing string to a Function**

#include <stdio.h>

void displayString(char str[]);

int main()

{

char str[50];

printf("Enter string: ");

gets(str);

displayString(str); // Passing string to a function.

return 0;

}

void displayString(char str[])

{

printf("String Output: ");

puts(str);

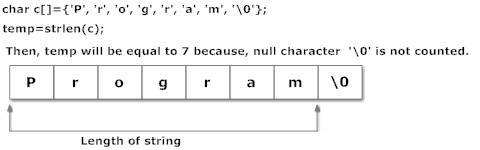
}

C strlen() Prototype

int strlen(string);

The function takes a single argument, i.e, the string variable whose length is to be found, and returns the length of the string passed.

The strlen() function is defined in [<string.h>](https://www.programiz.com/c-programming/library-function/string.h) header file.



Example: C strlen() function

#include <stdio.h>

#include <string.h>

int main()

{

char a[20]="Program";

char b[20]={'P','r','o','g','r','a','m','\0'};

char c[20];

printf("Enter string: ");

gets(c);

printf("Length of string a = %d \n",strlen(a));

//calculates the length of string before null charcter.

printf("Length of string b = %d \n",strlen(b));

printf("Length of string c = %d \n",strlen(c));

return 0;

}

**Output**

Enter string: String

Length of string a = 7

Length of string b = 7

Length of string c = 6

**C strcpy()**

**The strcpy() function copies the string to the another character array.**

The strcpy() function is defined in string.h header file.

**Example: C strcpy()**

#include <stdio.h>

#include <string.h>

int main()

{

char str1[10]= "awesome";

char str2[10];

char str3[10];

strcpy(str2, str1);

strcpy(str3, "well");

puts(str2);

puts(str3);

return 0;

}

**C strcmp()**

**The strcmp() function compares two strings and returns 0 if both strings are identical.**

The strcmp() function takes two strings and return an integer.

The strcmp() compares two strings character by character. If the first character of two strings are equal, next character of two strings are compared. This continues until the corresponding characters of two strings are different or a null character '\0' is reached.

It is defined in string.h header file.

Return Value from strcmp()

| Return Value | Remarks |
| --- | --- |
| 0 | if both strings are identical (equal) |
| negative | if the ASCII value of first unmatched character is less than second. |
| positive integer | if the ASCII value of first unmatched character is greater than second. |

Example: C strcmp() function

#include <stdio.h>

#include <string.h>

int main()

{

char str1[] = "abcd", str2[] = "abCd", str3[] = "abcd";

int result;

// comparing strings str1 and str2

result = strcmp(str1, str2);

printf("strcmp(str1, str2) = %d\n", result);

// comparing strings str1 and str3

result = strcmp(str1, str3);

printf("strcmp(str1, str3) = %d\n", result);

return 0;

}

**Output**

strcmp(str1, str2) = 32

strcmp(str1, str3) = 0

The first unmatched character between string str1 and str2 is third character. The ASCII value of 'c' is 99 and the ASCII value of 'C' is 67. Hence, when strings str1 and str2 are compared, the return value is 32.

When strings str1 and str3 are compared, the result is 0 because both strings are identical.

**C strcat()**

**The function strcat() concatenates two strings.**

In C programming, strcat() concatenates (joins) two strings.

The strcat() function is defined in [<string.h>](https://www.programiz.com/c-programming/library-function/string.h) header file.

C strcat() Prototype

strcat(string1, string2)

It takes two arguments, i.e, two strings or character arrays, and stores the resultant concatenated string in the first string specified in the argument.

Example: C strcat() function

#include <stdio.h>

#include <string.h>

int main()

{

char str1[] = "This is ", str2[] = "programiz.com";

//concatenates str1 and str2 and resultant string is stored in str1.

strcat(str1,str2);

puts(str1);

puts(str2);

return 0;

}

**Output**

This is programiz.com

programiz.com

Example: Find the Frequency of Characters

#include <stdio.h>

int main()

{

char str[1000], ch;

int i, frequency = 0;

printf("Enter a string: ");

gets(str);

printf("Enter a character to find the frequency: ");

scanf("%c",&ch);

for(i = 0; str[i] != '\0'; ++i)

{

if(ch == str[i])

++frequency;

}

printf("Frequency of %c = %d", ch, frequency);

return 0;

}

**Output**

Enter a string: This website is awesome.

Enter a character to find the frequency: e

Frequency of e = 4

Example: Program to count vowels, consonants etc.

#include <stdio.h>

int main()

{

char line[150];

int i, vowels, consonants, digits, spaces;

vowels = consonants = digits = spaces = 0;

printf("Enter a line of string: ");

scanf("%[^\n]", line);

for(i=0; line[i]!='\0'; ++i)

{

if(line[i]=='a' || line[i]=='e' || line[i]=='i' ||

line[i]=='o' || line[i]=='u' || line[i]=='A' ||

line[i]=='E' || line[i]=='I' || line[i]=='O' ||

line[i]=='U')

{

++vowels;

}

else if((line[i]>='a'&& line[i]<='z') || (line[i]>='A'&& line[i]<='Z'))

{

++consonants;

}

else if(line[i]>='0' && line[i]<='9')

{

++digits;

}

else if (line[i]==' ')

{

++spaces;

}

}

printf("Vowels: %d",vowels);

printf("\nConsonants: %d",consonants);

printf("\nDigits: %d",digits);

printf("\nWhite spaces: %d", spaces);

return 0;

}

**Output**

Enter a line of string: adfslkj34 34lkj343 34lk

Vowels: 1

Consonants: 11

Digits: 9

White spaces: 2

Example: Remove Characters in String Except Alphabets

#include<stdio.h>

int main()

{

char line[150];

int i, j;

printf("Enter a string: ");

gets(line);

for(i = 0; line[i] != '\0'; ++i)

{

while (!( (line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A' && line[i] <= 'Z') || line[i] == '\0') )

{

for(j = i; line[j] != '\0'; ++j)

{

line[j] = line[j+1];

}

line[j] = '\0';

}

}

printf("Output String: ");

puts(line);

return 0;

}

**Output**

Enter a string: p2'r-o@gram84iz./

Output String: programiz

Example: Copy String Manually Without Using strcpy()

#include <stdio.h>

int main()

{

char s1[100], s2[100], i;

printf("Enter string s1: ");

scanf("%s",s1);

for(i = 0; s1[i] != '\0'; ++i)

{

s2[i] = s1[i];

}

s2[i] = '\0';

printf("String s2: %s", s2);

return 0;

}

**Output**

Enter String s1: programiz

String s2: programiz

Example: Calculate Length of String without Using strlen() Function

#include <stdio.h>

int main()

{

char s[1000];

int i;

printf("Enter a string: ");

scanf("%s", s);

for(i = 0; s[i] != '\0'; ++i);

printf("Length of string: %d", i);

return 0;

}

**Output**

Enter a string: Programiz

Length of string: 9

Example: Program to Sort Strings in Dictionary Order

#include<stdio.h>

#include <string.h>

int main()

{

int i, j;

char str[10][50], temp[50];

printf("Enter 10 words:\n");

for(i=0; i<10; ++i)

scanf("%s[^\n]",str[i]);

for(i=0; i<9; ++i)

for(j=i+1; j<10 ; ++j)

{

if(strcmp(str[i], str[j])>0)

{

strcpy(temp, str[i]);

strcpy(str[i], str[j]);

strcpy(str[j], temp);

}

}

printf("\nIn lexicographical order: \n");

for(i=0; i<10; ++i)

{

puts(str[i]);

}

return 0;

}

**Output**

Enter 10 words:

C

C++

Java

PHP

Python

Perl

Ruby

R

JavaScript

PHP

In lexicographical order:

C

C++

Java

JavaScript

PHP

PHP

Perl

Python

R

Ruby

Example: Concatenate Two Strings Without Using strcat()

#include <stdio.h>

int main()

{

char s1[100], s2[100], i, j;

printf("Enter first string: ");

scanf("%s", s1);

printf("Enter second string: ");

scanf("%s", s2);

// calculate the length of string s1

// and store it in i

for(i = 0; s1[i] != '\0'; ++i);

for(j = 0; s2[j] != '\0'; ++j, ++i)

{

s1[i] = s2[j];

}

s1[i] = '\0';

printf("After concatenation: %s", s1);

return 0;

}

**Output**

Enter first string: lol

Enter second string: :)

After concatenation: lol:)