

C Library - strcmp() function

The **C** Library **strcmp()** function is used to compare two strings. It checks each character in the string one by one until it finds a difference or reaches the end of the one string. Additionally, the strings comparison is based on ASCII values.

Syntax

Following is the syntax of the C library **strcmp()** function –

```
strcmp(const char *str_1, const char *str_2)
```

Parameters

This function accepts the following parameters–

- **str_1** – This parameter define the first string to be compared.
- **str_2** – This is the second string to compare with first/previous string.

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Return Value

This function returns the integer value which is –

- Zero, if the string are equal.
- Negative, if the first string is less than second string lexicographically.
- Positive, if the first string is greater than second string lexicographically.

Example 1

In this example, we demonstrate the usage of string characters comparison using **strcmp()** function.



</>

Open Compiler

```

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

int main() {

    char str1[] = "abcd", str2[] = "abcd";
    int res;

    // Compare the strings str1 and str2
    res = strcmp(str1, str2);
    printf("strcmp(str1, str2) = %d\n", res);
    return 0;
}

```

Output

The above code produces the following result–

```
strcmp(str1, str3) = 0
```

Example 2

To check whether the given strings are case-sensitive or not using strcmp() and conditional statement.


[Open Compiler](#)

```

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

int main()
{
    char str1[] = "Tutorials";
    char str2[] = "tutorials";
    // Compare the strings
    int result = strcmp(str1, str2);

    if (result == 0) {
        printf("Strings are equal(case-sensitive)\n");
    }
}

```



```
    else {  
        printf("Strings are not equal(case-insensitive).\n");  
    }  
    return 0;  
}
```

Output

On execution of above code, we get the following result –

```
Strings are not equal(case-insensitive).
```

Example 3

Below the example create the strings of different size and check its comparison using strcmp().

</>

Open Compiler

```
#include <stdio.h>  
#include <string.h>  
  
int main()  
{  
    char str_1[] = "BOOK";  
    char str_2[] = "UNIVERSITY";  
  
    int res = strcmp(str_1, str_2);  
  
    if (res == 0) {  
        printf("The strings are equal.\n");  
    }  
    else if (res < 0) {  
        printf("Str_1 is less than str_2.\n");  
    }  
    else {  
        printf("Str_1 is greater than str_2.\n");  
    }  
    return 0;  
}
```

Output

After executing the above code, we get the following result–

```
Str_1 is less than str_2.
```

Example 4

Here, we use strcmp() to compare the two different string with the help of lexicographical order.

[Open Compiler](#)

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

int main {

    char str1[] = "text";
    char str2[] = "notebook";

    int result = strcmp(str1, str2);
    if (result < 0) {
        printf("%s' comes before '%s' lexicographically.\n", str1, str2);
    }
    else if (result > 0) {
        printf("%s' comes after '%s' lexicographically.\n", str1, str2);
    }
    else {
        printf("Strings are equal.\n");
    }
    return 0;
}
```

Explanation



In this code, str1 ("text") comes before str2 ("Notebook") lexicographically. The strcmp() function returns a negative value.

The above code produces the following result –

'text' comes after 'notebook' lexicographically.

