TutorialKart

# How to Compare Strings in C?

## String Comparison in C

In strings strcmp() function is used to compare two strings under a common header file called string.h .This function returns a negative,zero or a positive integer depending on the string pointed to,by str1 to string pointed to by str2.

### Syntax – strcmp()

```
int strcmp(const char *str1,const char *str2);
```

Arguments

- **str1** is first array to compare.
- **str2** is the second array to compare.

Return values

- If str1=str2 function returns 0.
- If str1<str2 function returns a negative integer.
- If str1>str2 function returns a positive integer.
- If string doen’t exists in str2 then it returns NULL.

### How return value of strcmp() is calculated ?

**strcmp**() function cannot compare strings by length. It compares strings by its ASCII values. If s1=”apple” and s2=”banana”, when we compare both the strings like strcmp(s1,s2), then this function returns a negative integer because ASCII value of ‘a’ is less than the ASCII value of ‘b’. If both the first characters are same then it goes for second characters.

### Examples

### Example – Compare Strings using strcmp() function

If two strings are equal, strcmp(s1, s2) should return zero.

### C Program

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[]="apple";
    char s2[]="banana";
    if(strcmp(s1,s2)==0)
        printf("equal");
    if (strcmp(s1,s2)<0)
        printf("s1 less than s2");
    if(strcmp(s1,s2)>0    )
        printf("s1 greater than s2");

    return 0;
}
```

### Output

```
s1 less than s2
```

## Example – Compare Strings using Recursion

In string comparison using recursion we are using three conditions. If one of the pointer is NULL pointer then it returns -2 in order to stop the process. If two strings are identical it returns 0 after comparing every character in one string with another string. We recursively call CompareStrings() function in order to compare both the strings.

### C Program

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

int main() {
    char str[100],s[100];
    printf("enter strings to compare");
    gets(str);
    gets(s);
    printf("the result of comparison is %d\n",CompareStrings(s, str));
    return 0;
}

int CompareStrings(char *s, char *s1) {
    // if one of the pointer is a NULL pointer return directly -2
    // in order to stop the process
    if(s==NULL || s1==NULL)
            return -2

    if(strcmp(s,s1)==0) // the two strings are identical
        return 0;

    if((s[0])==(s1[0]) && (s[0])==((s1+1)[0]))
```

```
        CompareStrings(s, ++s1);
    else if((s[0])==(s1[0]) && (s1[0])==((s+1)[0]))
        CompareStrings(++s, s1);
    else if((s[0])==(s1[0]))
        CompareStrings(++s, ++s1);
    else
        return -1;
}
```

## Example – Compare Strings using Pointers

In this example, we will use string pointers and write a function to compare the two strings.

### C Program

```
#include<stdio.h>
int compstring(char* s1, char* s2);
int main() {
    char s1[100], s2[100];
    int result;

    printf("Input a string1\n");
    gets(s1);

    printf("Input a string2\n");
    gets(s2);

    result = compstring(s1,s2);

    if (result == 0)
        printf("The strings are same.\n");
    else
        printf("The strings are different.\n");

    return 0;
}

int compstring(char *s1, char *s2) {
    while (*s1== *s2) {
        if (*s1 == '\0' || *s2 == '\0')
            break;
        s1++;
        s2++;
    }

    if (*s1 == '\0' && *s2== '\0')
        return 0;
    else
        return -1;
}
```

## Conclusion

In this C Tutorial – **Compare Strings**, we have gone through the syntax of strcmp() inbuilt function and two other ways using recursion and pointers respectively with Example C Programs.

**C Programming**

- ✦ C Tutorial
- ✦ C Data Types
- ✦ C Variables
- ✦ C Constants
- ✦ C if
- ✦ C if else
- ✦ C Ternary Operator
- ✦ C loops
- ✦ C While Loop
- ✦ C For Loop
- ✦ C Structures
- ✦ C Unions
- ✦ C typedef

**C String Operations**

- ✦ C Reverse String
- ✦ C String Length
- ➪ **C Compare Strings**

**C FileOperations**

- ✦ C Write to File
- ✦ C Delete File
- ✦ C Concatenate Files