

# Compare String

## Introduction

The `strcmp` function is part of the standard C library and is used to compare two strings. This function determines the lexicographical order of the given strings.

## Syntax

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

```
int strcmp(char str1[], char str2[]);
```

- **Parameters:**

- `str1`: Array representing the first string.
- `str2`: Array representing the second string.

- **Return Value:**

- Returns a value less than 0 if `str1` is less than `str2`.
- Returns 0 if `str1` is equal to `str2`.
- Returns a value greater than 0 if `str1` is greater than `str2`.

## Explanation

The `strcmp` function compares two strings character by character in a lexicographical manner. The comparison stops when a difference is found or when the end of one of the strings is reached.

## Example Usage

Below is a simple C program demonstrating the use of `strcmp`:

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

int main() {
    char str1[] = "Geeks";
    char str2[] = "Geeks";
    char str3[] = "GEEKS";

    // Comparing str1 and str2
    int result = strcmp(str1, str2);
    if (result == 0) {
        printf("str1 and str2 are equal.\n");
    } else {
        printf("str1 and str2 are different.\n");
    }

    // Comparing str1 and str3
    result = strcmp(str1, str3);
    if (result == 0) {
        printf("str1 and str3 are equal.\n");
    } else {
        printf("str1 and str3 are different.\n");
    }

    return 0;
}
```

## Expected Output:

```
str1 and str2 are equal.
str1 and str3 are different.
```

In the example above, `strcmp(str1, str2)` returns 0 because both strings are identical. However, `strcmp(str1, str3)` returns a nonzero value because of the difference in letter casing.

## Using `strcmp` in Searching

The `strcmp` function can also be used in searching within an array of strings. Below is an example demonstrating how to search for a string in an array using `strcmp`:

```
#include <stdio.h>
#include <string.h>
#define SIZE 5

int main() {
    char names[SIZE][20] = {"Alice", "Bob", "Charlie", "David", "Eve"};
    char search[20];
    int found = 0;

    printf("Enter a name to search: ");
    scanf("%s", search);

    for (int i = 0; i < SIZE; i++) {
        if (strcmp(names[i], search) == 0) {
            printf("%s found at position %d\n", search, i + 1);
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("%s not found in the list.\n", search);
    }

    return 0;
}
```

## Expected Output:

```
Enter a name to search: Bob
Bob found at position 2
```

This program takes user input for a name and searches for it in the predefined list using `strcmp`.

## Important Notes

- The `strcmp` function is **case-sensitive**. To perform a case-insensitive comparison, you can use `strcasecmp` (available on some platforms) or convert both strings to lowercase or uppercase before comparing.
  - Ensure both string arrays are valid and contain null-terminated strings (`'\0'`) to avoid undefined behavior.
- 

Revision #1

Created 22 February 2025 02:23:00 by YP

Updated 22 February 2025 02:29:09 by YP