# APPLICATION OF RECURSIVE FUNCTION CALL

## RESEARCH :

1. Strings in the C programming language are stored as arrays of characters and are always terminated by a null character (‘\0’) Although built-in functions like sterlen() and strcat() are commonly used for string operations, implementing these functions manually helps to understand how strings work internally in memory. This research explains how to calculate the length of a string and how to concatenate two strings without using any built-in functions.

The objective of this research is to develop a C program that:

1. Calculates the length of a string manually.

2. Concatenates two strings manually.
   Both tasks must be done without using built-in functions like strlen() or strcat()

**String Representation in C**

In C, a string is stored as a character array and always ends with a null character (‘\0’). This null character tells the compiler where the string stops.

Example:
The string “Ved” is stored as:
Ved \0

**Finding String Length Without strlen()**

To calculate the length of a string manually, we count each character one by one in a loop until we reach the null terminator (‘\0’). The number of characters counted is the length of the string.

**Concatenating Two Strings Without strcat()**

To concatenate manually, we first move to the end of the first string (where we find ('\0'). Then, we copy the characters of the second string one by one from this position. After copying, we add '\0' to indicate the end of the new combined string.

The null character ('\0') plays a crucial role in both string length calculation and string concatenation. In string length, it helps detect where the string ends. In concatenation, it helps to properly terminate the final combined string. Without using built-in functions, these operations are performed using simple loops and array indexing.

By manually implementing string length and concatenation functions, we gain a deeper understanding of how strings are stored and managed in C. It improves knowledge of arrays, loops, memory handling, and character-based processing. These concepts are useful in building custom string libraries, text editors, and system-level applications.

SOURCE :

1. https://www.geeksforgeeks.org/c/length-of-string-without-using-the-strlen-function-in-c/
2. https://www.w3schools.in/c-programming/examples/concatenate-two-strings-without-using-strcat

**ANALYSE :**

The program uses following steps :

1. It is used to calculate the length in a particular word without using the inbuilt functions as string length or strlen()
2. It is also used to combine the two words or many strings / words without using the inbuilt functions as string concatenation or strcat()

**ALGORITHM :**

For string length :

1. Start
2. Declare a character array called str
3. Set a counter variable i equal to 0
4. Repeat the following steps while str at position i is not equal to the null character
   a. Increase i by 1
5. When the loop stops, i will store the length of the string
6. Display the value of i
7. Stop

For string concatenation :

1. Start
2. Declare two character arrays called str1 and str2
3. Set i equal to 0
4. Move to the end of str1 by repeating the following while str1 at position i is not equal to the null character
   a. Increase i by 1
5. Set j equal to 0
6. Repeat the following steps while str2 at position j is not equal to the null character
   a. Set str1 at position i equal to str2 at position j
   b. Increase i and j by 1

7. Place a null character at str1 position i to end the string
8. Display the final string
9. Stop

**BUILD :**

String Length :

```
#include<stdio.h>



int count (const char *s){
    if(*s == '\0'){
        return 0;
    }
    else {
        return 1 + count(s+1);
    }
}

int main (){
    const char *a = "Ved";
```

```
    int length = count(a);
    printf("The length of \"%s\" ; %d\n", a,length);

    return 0;
}
```

String Concatenation :

```
#include <stdio.h>

void concatenate(char str1[], char str2[]) {
    int i = 0, j = 0;
    while (str1[i] != '\0') {
        i++;
    }
    while (str2[j] != '\0') {
        str1[i] = str2[j];
        i++;
        j++;
    }
    str1[i] = '\0';
}

int main() {
    char str1[200] = "Ved ";
```

```
    char str2[100] = "Muley";

    concatenate(str1, str2);

    printf("Concatenated String: %s\n", str1);

    return 0;
}
```

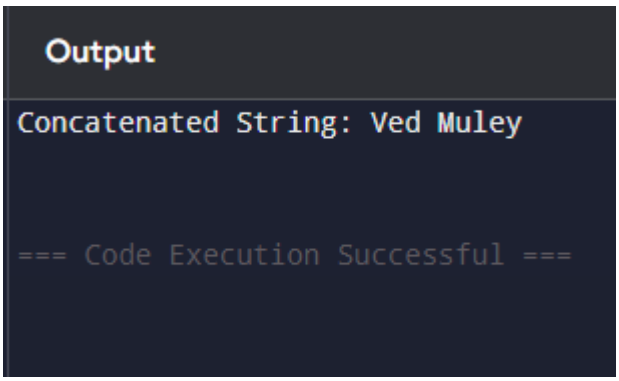
**OUTPUT :**

1. String length

```
Output
The length of "Ved" ; 3


=== Code Execution Successful ===
```

2. String Concatenation :




**IMPLEMENTATION :**

https://github.com/vedmuley536/Ved-Muley