A Micro Project Report

on

Problem Solving using C Language

Submitted by
KAGITALA.BHAVYA (23471A05FW)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-1

NIRF rank in the band of 201-300 and is an ISO 9001:2015 certified Approved by AICTE, New Delhi, Permanently affiliated to JNTU Kakinada, Approved by AICTE, Accredited by NBA and accredited 'A+' grade by NAAC Narasaraopet-522601, Palnadu(Dt.), Andhra Pradesh, India

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that K . Bhavya , Roll No: 23471A05FW, a Second Year Student of the Department of Computer Science and Engineering, has completed the Micro Project Satisfactorily in "Problem Solving using C Language" for the Academic Year 2024-2025..

Project Co-Ordinator
Mr. M. Venkata Rao, M.Tech.
Asst. Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor

INDEX

String Palindrome

1.C program to check String palindrome without using String handling functions

Program:

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

int main()
{
  char str[100];
  int i, len, flag;
  flag = 0;

  printf("\n Please Enter any String :  ");
  gets(str);

  len = strlen(str);

  for(i = 0; i < len; i++)
  {
     if(str[i] != str[len - i - 1])
```

```
        {
                flag = 1;
                break;
        }
    }
    if(flag == 0)
    {
        printf("\n %s is a Palindrome String", str);
    }
    else
    {
        printf("\n %s is Not a Palindrome String", str);
    }

    return 0;
}
```

OUTPUT:

```
 Please Enter any String :  siri

 siri  is Not a Palindrome String

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

Shortest word

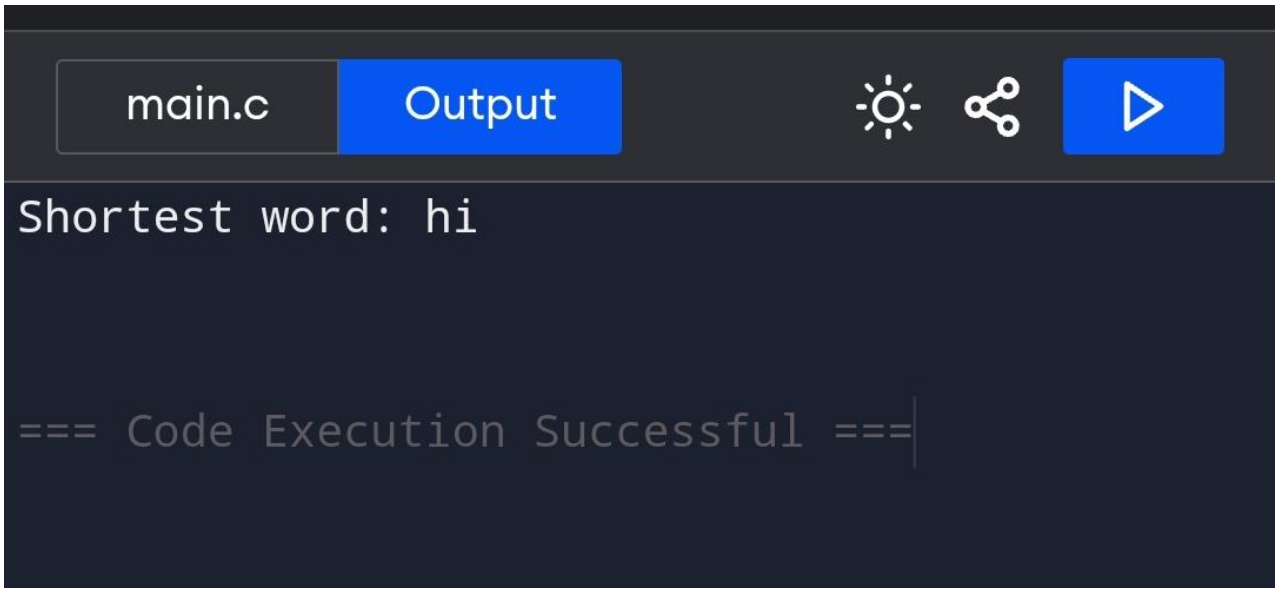
2.Find shortest word from given sentence "hi","hello","Thinava" using c program.

Program:

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

int main() {
   char words[3][10] = {"hi", "hello", "thinava"};
   char shortest[10] = "";
   int min_len = 100; // Initialize with a large value

   for (int i = 0; i < 3; i++) {
      int len = strlen(words[i]);
      if (len < min_len) {
         min_len = len;
         strcpy(shortest, words[i]);
      }
   }
```

```
    printf("Shortest word: %s\n", shortest);

    return 0;
}
```

Output :



String strcpy

3.String copy Without Strcpy()

Program:

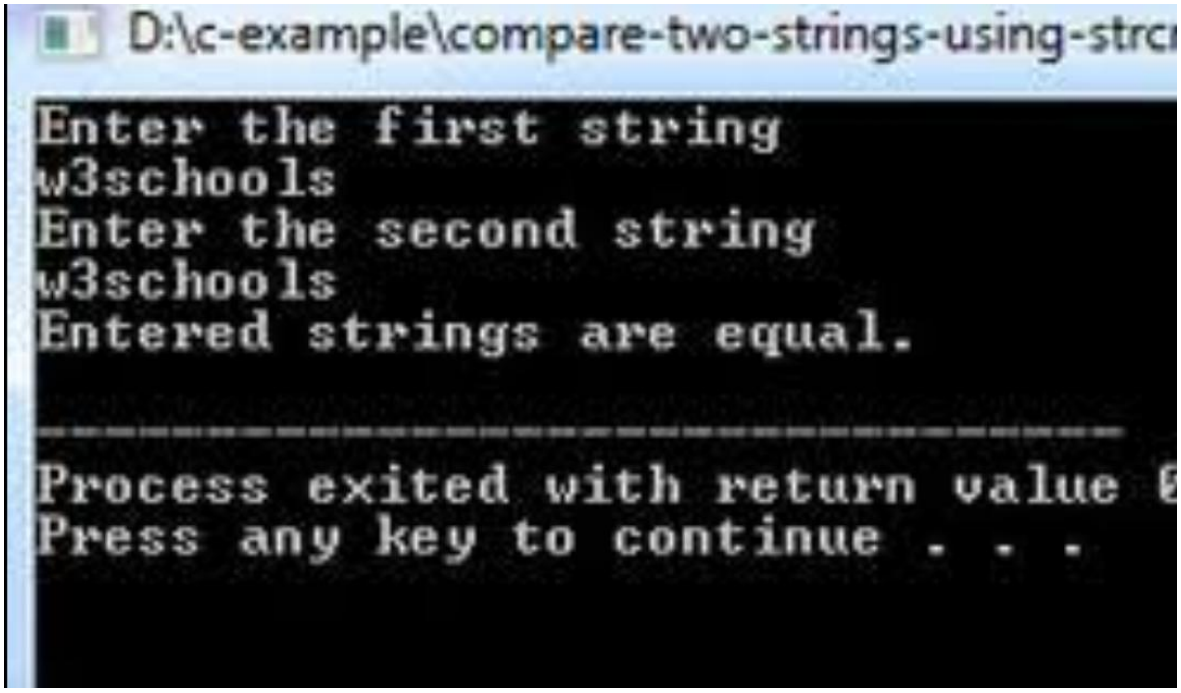
```
#include <stdio.h>
#include <string.h>

int main()
{
	char Str[100], CopyStr[100];
	int i;

	printf("\n Please Enter any String :  ");
	gets(Str);

	for (i = 0; Str[i]!='\0'; i++)
	{
		CopyStr[i] = Str[i];
	}
	CopyStr[i] = '\0';

	printf("\n String that we coped into CopyStr = %s",
CopyStr);
```

```
  printf("\n Total Number of Characters that we copied = %d\n",
  i);

        return 0;
}
```

OUTPUT:

```
Please Enter any String :  prasanna

String that we coped into CopyStr = prasanna
Total Number of Characters that we copied =
    8


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

OUTPUT:


D:\c-example\compare-two-strings-using-strc
Enter the first string
w3schools
Enter the second string
w3schools
Entered strings are equal.
Process exited with return value
Press any key to continue . . .

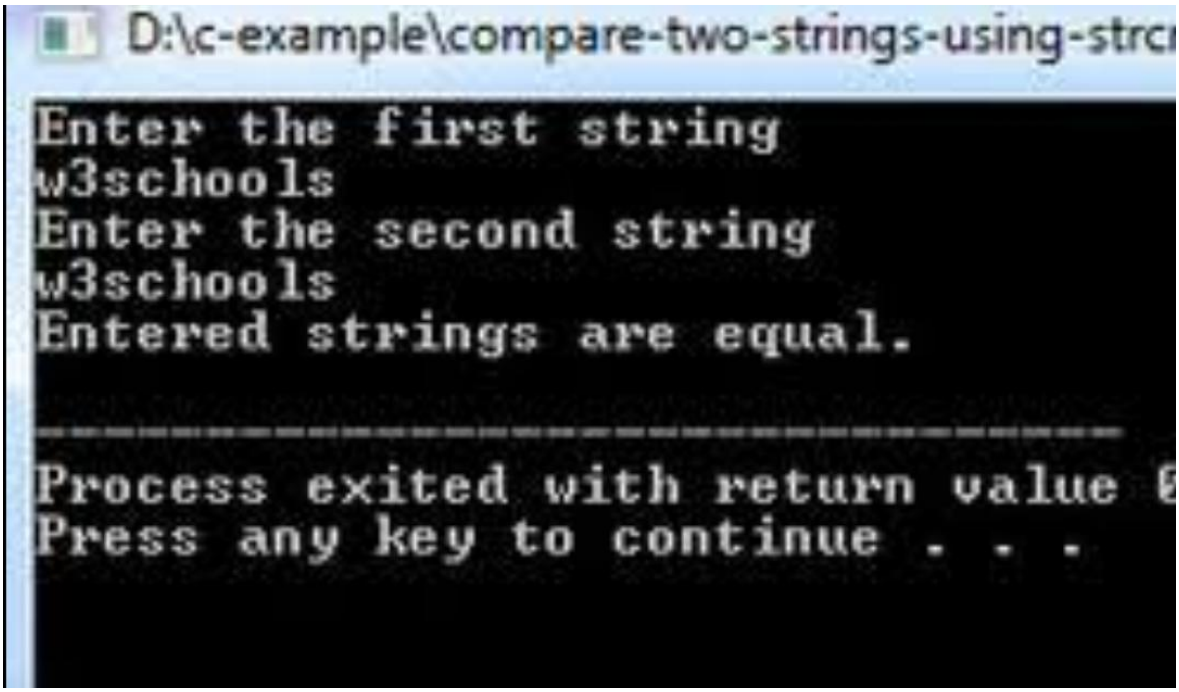
String strcmp

4.Compare String without Strcmp()

Program:

```
int compareStrings(char *str1, char *str2) {
 while (*str1 && (*str1 == *str2)) {
    str1++;
    str2++;
  }
  return *str1 - *str2;
}
```

OUTPUT:


D:\c-example\compare-two-strings-using-strcr
Enter the first string
w3schools
Enter the second string
w3schools
Entered strings are equal.
Process exited with return value 0
Press any key to continue . . .

String strcmp

5.Write a program that uses an array of pointers to strings str[].Recieve two strings str1 and str2 and check if str1 is embedded in any of the strings in str[],if str1 is found,then replace it with string 2
Char*Str[]={"we will teach you how to .....",
"Move a mountain ",Level a building",
"Erase the past",Make a million",
....all through!"}

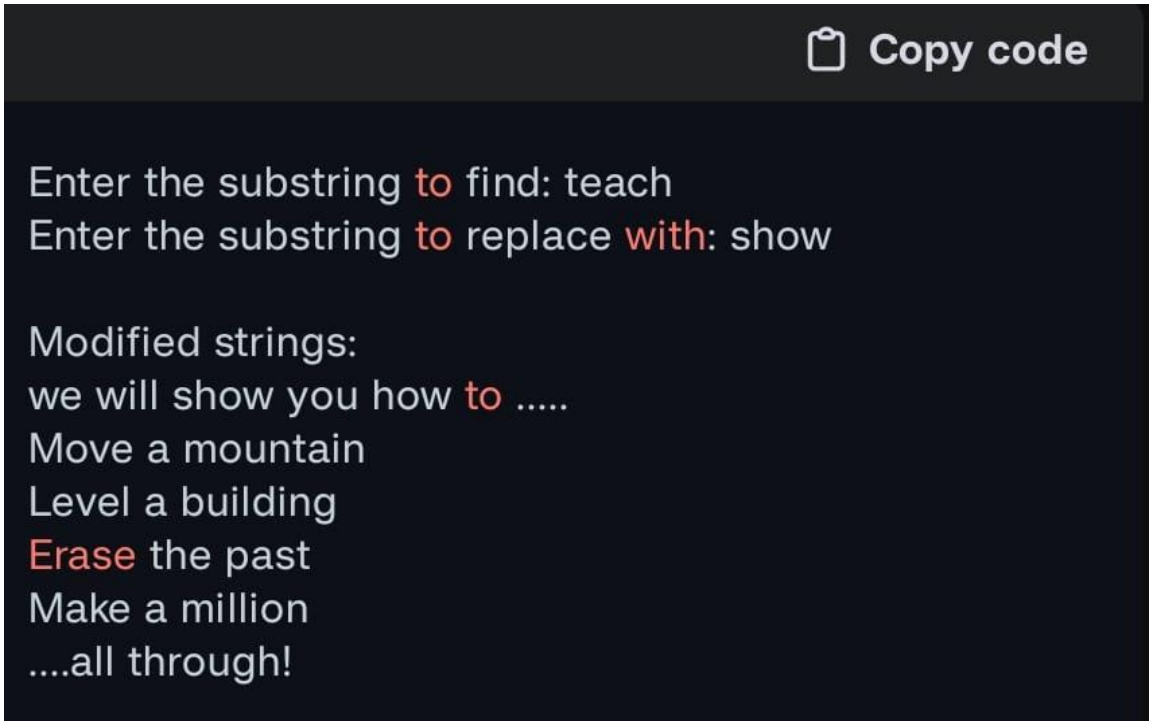
Program:

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

#define MAX_STRINGS 10
#define MAX_LENGTH 100

void replace_string(char* str[], char* str1, char* str2) {
    for (int i = 0; i < MAX_STRINGS; i++) {
        if (str[i] != NULL) {
            char* ptr = strstr(str[i], str1);
```

```
            if (ptr != NULL) {
                printf("Found '%s' in '%s'\n", str1, str[i]);
                // Replace str1 with str2
                char temp[MAX_LENGTH];
                strcpy(temp, str[i]);
                printf("Replacing '%s' with '%s' in '%s'\n", str1, str2,
str[i]);
                char* token = strtok(temp, str1);
                char new_string[MAX_LENGTH];
                sprintf(new_string, "%s%s%s", token, str2, ptr +
strlen(str1));
                strcpy(str[i], new_string);
                printf("Modified string: %s\n", str[i]);
            }
        }
    }
}

int main() {
    char* str[MAX_STRINGS] = {
        "We will teach you how to Move a mountain.",
        "Level a building.",
        "Erase the past.",
        "Make a million.",
        "All through!",
        NULL, NULL, NULL, NULL, NULL
    };

    char str1[] = "a ";
```

```
    char str2[] = "the ";

    replace_string(str, str1, str2);

    return 0;
}
```

Output:


Copy code
Enter the substring to find: teach
Enter the substring to replace with: show
Modified strings:
we will show you how to .....
Move a mountain
Level a building
Erase the past
Make a million
....all through!