----------**CONTENTS**-----------

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.  No. | C PROGRAMMING | Page  NO. | Teacher’s  Signature |
| 1 | Write a C program to find length of a string. |  |  |
| 2 | Write a C program to copy one string to another string. |  |  |
| 3 | Write a C program to concatenate two strings. |  |  |
| 4 | Write a C program to compare two strings. |  |  |
| 5 | Write a C program to convert lowercase string to uppercase. |  |  |
| 6 | Write a C program to find total number of alphabets, digits or special character in a string. |  |  |
| 7 | Write a C program to count total number of words in a string. |  |  |
| 8 | Write a C program to find reverse of a string. |  |  |
| 9 | Write a C program to check whether a string is palindrome or not. |  |  |
| 10 | Write a C program to find first occurrence of a character in a given string. |  |  |
| 11 | Write a C program to count frequency of each character in a string. |  |  |
|  |  |  |  |

1. Write a C program to find length of a string.

C CODE:

#include <stdio.h>

int main()

{

    char str[100], i = 0;

    printf("-----------Describe yourself-----------\n");

    gets(str);

    while (str[i] != '\0')

    {

        i++;

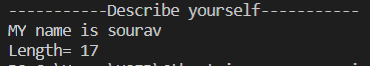
    }

    printf("Length= %d", i);

    return 0;

}

OUTPUT:



2. Write a C program to copy one string to another string.

C CODE:

#include <stdio.h>

int main()

{

  char str1[100], str2[100];

  int i;

  printf("Type here which is copied\n");

  gets(str1);

  i = 0;

  while (str1[i] != '\0')

  {

    str2[i] = str1[i];

    i++;

  }

  str2[i] = '\0';

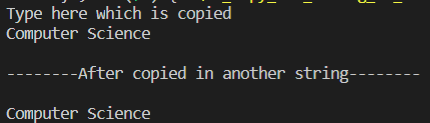
  printf("\n--------After copied in another string--------\n");

  printf("\n%s", str2);

  return 0;

}

OUTPUT:



3. Write a C program to concatenate two strings.

C CODE:

#include <stdio.h>

int main()

{

    char str1[100], str2[100];

    int i = 0, j = 0;

    printf("Enter the first string:\n");

    gets(str1);

    printf("Enter the second string:\n");

    gets(str2);

    while (str2[i] != '\0')

    {

        i++;

    }

    while (str1[j] != '\0')

    {

        str2[i] = str1[j];

        i++;

        j++;

    }

    str2[i] = '\0';

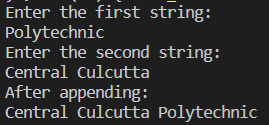
    printf("After appending:\n");

    puts(str2);

    return 0;

}

OUTPUT:



5. Write a C program to convert lowercase string to uppercase.

C CODE:

#include <stdio.h>

int main()

{

    char str[100], upper\_str[100];

    int i;

    printf("Enter word/sentence:\n");

    gets(str);

    for (i = 0; str[i] != '\0'; i++)

    {

        if (str[i] >= 97 && str[i] <= 123)

        {

            upper\_str[i] = str[i] - 32;

        }

        else

        {

            upper\_str[i] = str[i];

        }

    }

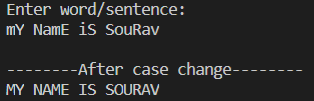
    upper\_str[i] = '\0';

    printf("\n--------After case change--------\n");

    printf("%s", upper\_str);

    return 0;

}

OUTPUT:  


6. Write a C program to find total number of alphabets, digits or special character in a string.

C CODE:

#include <stdio.h>

int main()

{

    char str[100];

    int digit = 0, alphabet = 0, sp\_char = 0;

    printf("Enter string:\n");

    gets(str);

    for (int i = 0; str[i] != '\0'; i++)

    {

        if ((str[i] >= 65 && str[i] <= 91) || (str[i] >= 97 && str[i] <= 123))

        {

            alphabet++;

        }

        else if (str[i] >= 48 && str[i] <= 57)

        {

            digit++;

        }

        else

        {

            sp\_char++;

        }

    }

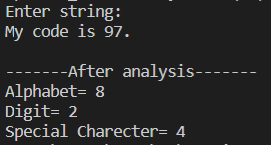
    printf("\n-------After analysis-------\n");

    printf("Alphabet= %d\nDigit= %d\nSpecial Charecter= %d", alphabet, digit, sp\_char);

    return 0;

}

OUTPUT:



7. Write a C program to count total number of words in a string.

C CODE:

#include <stdio.h>

int main()

{

    char str[100];

    int w = 0;

    printf("Enter string:\n");

    gets(str);

    for (int i = 0; str[i] != '\0'; i++)

    {

        if (str[i] == 32)

        {

            w++;

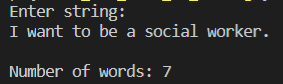
        }

    }

    printf("\nNumber of words: %d", w + 1);

    return 0;

}

OUTPUT:  


9. Write a C program to check whether a string is palindrome or not.

C CODE:

#include <stdio.h>

int main()

{

    char str[50];

    int j = 0, c = 0, i;

    printf("Enter a string:\n");

    gets(str);

    i = 0;

    while (str[i] != '\0')

    {

        j++;

        i++;

    }

    for (int i = 0; i <= j; i++)

    {

        if (str[i] == str[j - i - 1])

        {

            c++;

        }

    }

    if (c == i)

    {

        printf("\npallindrome");

    }

    else

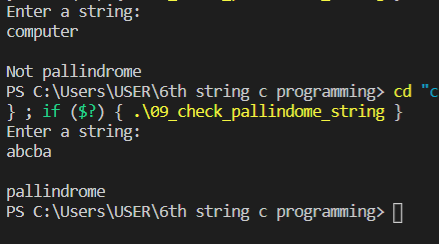
    {

        printf("\nNot pallindrome");

    }

    return 0;

}

OUTPUT:  


10. Write a C program to find first occurrence of a character in a given string.

C CODE:

#include <stdio.h>

#include <string.h>

int main()

{

    char str[50], ch;

    int flag = 0;

    printf("Enter a string:\n");

    gets(str);

    printf("Enter the charecter to be searched:\n");

    scanf("%c", &ch);

    for (int i = 0; i <= strlen(str); i++)

    {

        if (str[i] == ch)

        {

            flag = 1;

            break;

        }

    }

    if (flag == 1)

    {

        printf("Charecter found!");

    }

    else

    {

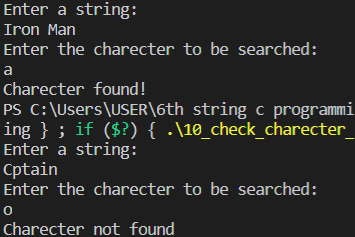
        printf("Charecter not found");

    }

    return 0;

}

OUTPUT:



11. Write a C program to count frequency of each character in a string.

C CODE:

#include <stdio.h>

int main()

{

    char str[50];

    int i, j, count = 0, n = 0;

    printf("Enter a string:\n");

    gets(str);

    for (i = 0; str[i]; i++)

    {

        n++;

    }

    for (i = 0; i < n; i++)

    {

        count = 1;

        if (str[i])

        {

            for (j = i + 1; j < n; j++)

            {

                if (str[j] == str[i])

                {

                    count++;

                    str[j] = '\0';

                }

            }

            printf(" %c = %d\n", str[i], count);

        }

    }

    return 0;

}

OUTPUT:  
