-------CONTENTS------

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.  No. | PROGRAM TITTLE | PAGE  NO. | Teacher’s  Signature |
| 01 | Write a C program to find length of a string. |  |  |
| 02 | Write a C program to copy one string to another string. |  |  |
| 03 | Write a C program to concatenate two strings. |  |  |
| 04 | Write a C program to compare two strings. |  |  |
| 05 | Write a C program to convert lowercase string to uppercase. |  |  |
| 06 | Write a C program to find total number of alphabets, digits or special character in a string |  |  |
| 07 | Write a C program to count total number of words in a string. |  |  |
| 08 | Write a C program to find reverse of a string. |  |  |
| 09 | 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. |  |  |
|  | POINTER |  |  |
| 01 | Write a C program to add two numbers using pointers. |  |  |
| 02 | Write a C program to swap two numbers using pointers. |  |  |
| 03 | Write a C program to input and print array elements using pointer. |  |  |
| 04 | Write a C program to copy one array to another using pointers. |  |  |
| 05 | Write a C program to swap two arrays using pointers. |  |  |
| 06 | Write a C program to reverse an array using pointers. |  |  |
| 07 | Write a C program to search an element in array using pointers. |  |  |
| 08 | Write a C program to access two dimensional array using pointers. |  |  |
| 09 | Write a C program to add two matrix using pointers. |  |  |
| 10 | Write a C program to find length of string using pointers. |  |  |

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

PROGRAM:

#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.

PROGRAM:

#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++;

  }

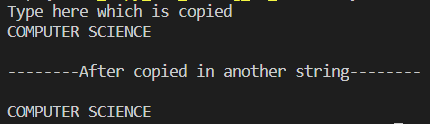
  str2[i] = '\0';

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

  printf("\n%s", str2);

  return 0;

}

OUTPUT: 

3. Write a C program to concatenate two strings.

PROGRAM:

#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';

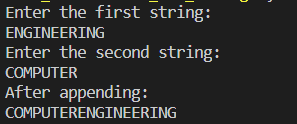
    printf("After appending:\n");

    puts(str2);

    return 0;

}

OUTPUT:



4. Write a C program to compare two strings.

PROGRAM:

#include<stdio.h>

int mystrcmp(char[], char[]);

int main()

{

    char s1[100], s2[100];

    int compare;

    printf("\nEnter 1st string: ");

    gets(s1);

    printf("\nEnter 2nd string: ");

    gets(s2);

    compare = mystrcmp(s1, s2);

    if(compare == 1)

        printf("\nBoth the strings are exactly same.\n");

     else

        printf("\nBoth the strings are different.\n");

}

int mystrcmp(char x[], char y[])  // function definition

{

    int i = 0, flag = 0;

    while(x[i] != '\0' && y[i] != '\0') // until atleast one string ends

    {

        if(x[i] != y[i])

        {

            flag = 1;

            break;

        }

        i++;

    }

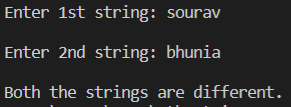
    if(flag == 0 && x[i] == '\0' && y[i] == '\0')

        return 1;

    else

        return 0;

}

OUTPUT:  
 

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

PROGRAM:

#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];

        }

    }

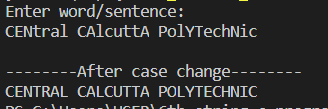
    upper\_str[i] = '\0';

    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.

PROGRAM:

#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++;

        }

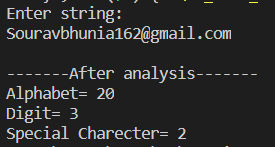
    }

    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.

PROGRAM:

#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++;

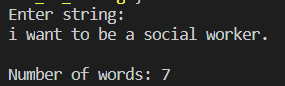
        }

    }

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

    return 0;

}

OUTPUT:  
 

8. Write a C program to find reverse of a string.

PROGRAM:

#include <stdio.h>

int main()

{

    char str[20], rev\_str[20];

    int i, j, max = 0;

    printf("Enter a string:\n");

    gets(str);

    i = 0;

    while (str[i] != '\0')

    {

        i++;

        max = i - 1;

    }

    printf("\n------Reverse String-----\n");

    for (j = 0; j <= i; j++)

    {

        rev\_str[j] = str[max];

        max--;

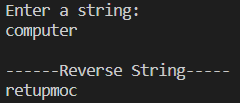
    }

    puts(rev\_str);

    return 0;

}

OUTPUT:



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

PROGRAM:

#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

    {

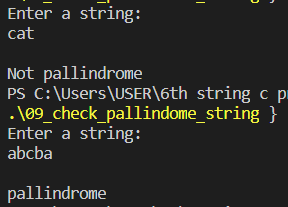
        printf("\nNot pallindrome");

    }

    return 0;

}

OUTPUT:



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

PROGRAM:

#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

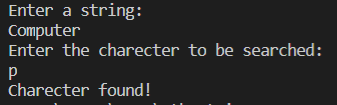
    {

        printf("Charecter not found");

    }

    return 0;

}

OUTPUT:  
 

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

PROGRAM:

#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);

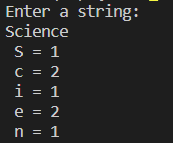
        }

    }

    return 0;

}

OUTPUT:



---------POINTER--------

1. Write a C program to add two numbers using pointers.

PROGRAM:

#include <stdio.h>

int main(){

int a,b;

int \*p,\*q;

p=&a;

q=&b;

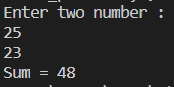
printf("Enter two number :\n");

scanf("%d%d",p,q);

printf("Sum = %d",(\*p+\*q));

return 0;

}

OUTPUT:  
 

1. Write a C program to swap two numbers using pointers.

POINTER:

#include <stdio.h>

void swap\_num(int \*p, int \*q)

{

    int temp;

    printf("\nAfter Swaping:\n");

    temp = \*p;

    \*p = \*q;

    \*q = temp;

    printf("%d, %d", \*p, \*q);

}

int main()

{

    int a, b;

    int \*ptr1, \*ptr2;

    ptr1 = &a;

    ptr2 = &b;

    printf("Enter two number :\n");

    scanf("%d%d", ptr1, ptr2);

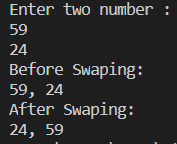
    printf("Before Swaping:\n");

    printf("%d, %d", \*ptr1, \*ptr2);

    swap\_num(ptr1, ptr2);

    return 0;

}

OUTPUT:  
 

3. Write a C program to input and print array elements using pointer.

PROGRAM:

#include <stdio.h>

int main()

{

    int arr[20], n;

    int \*ptr = arr;

    printf("Enter number of element to be stored:\n");

    scanf("%d", &n);

    printf("Enter elements:\n");

    for (int i = 0; i < n; i++)

    {

        scanf("%d", ptr + i);

    }

    printf("Elements are:\n");

    for (int i = 0; i < n; i++)

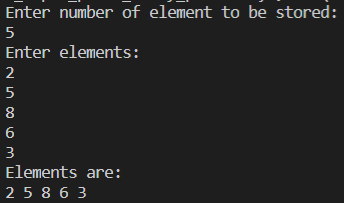
    {

        printf("%d ", \*(ptr + i));

    }

    return 0;

}

OUTPUT:  
 

4. Write a C program to copy one array to another using pointers.

POINTER:

#include <stdio.h>

int main()

{

    int arr[20], arr2[20], n, i;

    int \*ptr1, \*ptr2;

    ptr1 = &arr[0];

    ptr2 = &arr2[0];

    printf("Enter number of elements to be stored:\n");

    scanf("%d", &n);

    printf("Enter elements one by one:\n");

    for (i = 0; i < n; i++)

    {

        scanf("%d", ptr1 + i);

    }

    for (i = 0; i < n; i++)

    {

        arr2[i] = arr[i];

    }

    printf("\n-----In Frist Array----\n");

    for (i = 0; i < n; i++)

    {

        printf("%d ", \*ptr1);

        ptr1++;

    }

    printf("\n-----In Sceond Array----\n");

    for (i = 0; i < n; i++)

    {

        printf("%d ", \*ptr2);

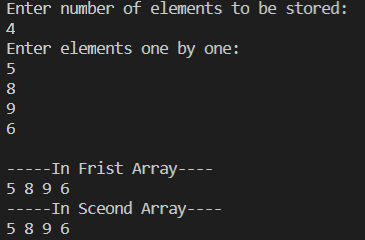
        ptr2++;

    }

    return 0;

}

OUTPUT:



5. Write a C program to swap two arrays using pointers.

PROGRAM:

6. Write a C program to reverse an array using pointers.

PROGRAM:

#include <stdio.h>

int main()

{

    int arr1[20], arr2[20], no, j;

    int \*ptr1, \*ptr2;

    ptr1 = &arr1[0];

    ptr2 = &arr2[0];

    printf("Enter number of elements to be stored:\n");

    scanf("%d", &no);

    printf("Enter elements:\n");

    for (int i = 0; i < no; i++)

    {

        scanf("%d", (ptr1 + i));

    }

    j = 0;

    for (int i = no - 1; i >= 0; i--)

    {

        arr2[j] = arr1[i];

        j++;

    }

    printf("\n----Reverse Array----\n");

    for (int j = 0; j < no; j++)

    {

        printf("%d ", \*ptr2);

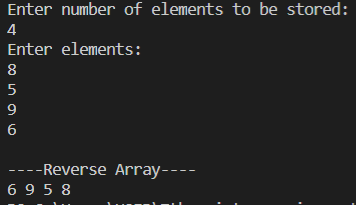
        ptr2++;

    }

    return 0;

}

OUTPUT:



7. Write a C program to search an element in array using pointers.

PROGRAM:

#include <stdio.h>

int main()

{

    int arr[20], num, n, pos = 0, flag = 0;

    int \*ptr;

    ptr = &arr[0];

    printf("Enter number of elments to be stored:\n");

    scanf("%d", &num);

    printf("Enter elments :\n");

    for (int i = 0; i < num; i++)

    {

        scanf("%d", (ptr + i));

    }

    printf("Enter number to be search:\n");

    scanf("%d", &n);

    for (int i = 0; i < n; i++)

    {

        if (\*(ptr + i) == n)

        {

            flag = 1;

            break;

        }

        pos++;

    }

    if (flag == 1)

    {

        printf("Element found sucessfully at position %d", pos + 1);

    }

    else

    {

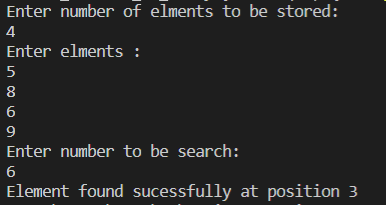
        printf("Element not found");

    }

    return 0;

}

OUTPUT:



8. Write a C program to access two dimensional array using pointers.

PROGRAM:

#include <stdio.h>

int main()

{

    int arr[20][20], i, j, n, m;

    int(\*ptr)[20];

    ptr = arr;

    printf("Enter number of row and column: \n");

    scanf("%d%d", &n, &m);

    printf("Enter element row wise:\n");

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < m; j++)

        {

            scanf("%d", (\*(ptr + i) + j));

        }

    }

    printf("\n-------2D Array-----\n");

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < m; j++)

        {

            printf("%d\t", \*(\*(ptr + i) + j));

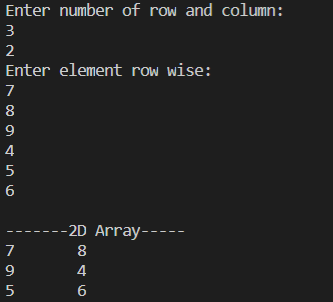
        }

        printf("\n");

    }

    return 0;

}

OUTPUT:  
 

9. Write a C program to add two matrix using pointers.

PROGRAM:

#include <stdio.h>

int main()

{

    int arr1[20][20], arr2[20][20], arr3[20][20], n, r, c, i, j;

    int(\*ptr1)[20], (\*ptr2)[20], (\*ptr3)[20];

    ptr1 = arr1;

    ptr2 = arr2;

    ptr3 = arr3;

    printf("Enter number of row and column respectively:\n");

    scanf("%d%d", &r, &c);

    printf("Enter elements of first matrix in row major method:\n");

    for (i = 0; i < r; i++)

    {

        for (j = 0; j < c; j++)

        {

            scanf("%d", (\*(ptr1 + i) + j));

        }

    }

    printf("Enter elements of sceond matrix in row major method:\n");

    for (i = 0; i < r; i++)

    {

        for (j = 0; j < c; j++)

        {

            scanf("%d", (\*(ptr2 + i) + j));

        }

    }

    for (i = 0; i < r; i++)

    {

        for (j = 0; j < c; j++)

        {

            arr3[i][j] = arr1[i][j] + arr2[i][j];

        }

    }

    printf("\n-------Sum of two matrix-------\n");

    for (i = 0; i < r; i++)

    {

        for (j = 0; j < c; j++)

        {

            printf(" %d\t", \*(\*(ptr3 + i) + j));

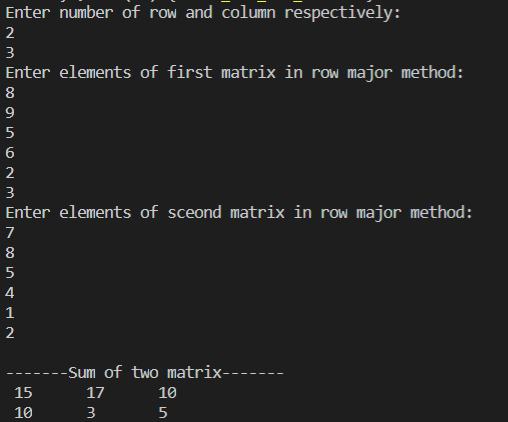
        }

        printf("\n");

    }

    return 0;

}

OUTPUT: 

10. Write a C program to find length of string using pointers.

PROGRAM:

#include <stdio.h>

int main()

{

    char str[20];

    char \*ptr;

    int count = 0;

    ptr = str;

    printf("Enter a string;\n");

    gets(str);

    while (\*ptr != '\0')

    {

        // printf("%c",\*ptr);

        count++;

        ptr++;

    }

    printf("Length = %d", count);

    return 0;

}

OUTPUT:  
