**C PROGRAMMING ASSIGNMENT:**

**19**

DATE: 13.01.2022

SUBMITTED BY: -

NAME: MUKTESH MISHRA

BRANCH: CSE

SECTION: B22

ROLL NO.: 21052258

1. WAP to find out the largest element present in a 5 X 4 matrix . In the same fun store the largest element of each row in a 1D array

***Code:***

#include <stdio.h>

*int* main()

{

*int* m,n,a[10];

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

    scanf("%d %d",&m,&n);

*int* arr[10][10];

    printf("\nEnter the elements of the matrix: \n");

    for(*int* i=0;i<m;i++)

    {

        for(*int* j=0;j<n;j++)

        {

            scanf("%d",&arr[i][j]);

        }

    }

    printf("\nThe elements in the matrix are: \n");

    for(*int* i=0;i<m;i++)

    {

        for(*int* j=0;j<n;j++)

        {

            printf("%d ",arr[i][j]);

        }

        printf("\n");

    }

*int* i = 0, j;

*int* max = 0;

*int* res[m];

    while (i < m)

    {

       for ( j = 0; j < n; j++)

       {

           if (arr[i][j] > max)

           {

              max = arr[i][j];

           }

        }

        res[i] = max;

        max = 0;

        i++;

    }

    printf("Largest element in each rows are\n" );

    for(*int* i = 0; i < n; i++)

    {

       printf(" %d \t", i);

    }

    printf("\n");

    for(*int* i = 0; i < n; i++)

    {

       printf(" %d \t", res[i]);

    }

    return 0;

}

***Output:***



1. WAP to reverse a string

***Code:***

#include <stdio.h>

#include <string.h>

*void* revstr(*char* \**str1*)

{

    static *int* i, len, temp;

    len = strlen(*str1*);

    if (i < len/2){

        temp = *str1*[i];

*str1*[i] = *str1*[len - i - 1];

*str1*[len - i - 1] = temp;

        i++;

        revstr(*str1*);

    }

}

*int* main()

    {

*char* str1[50];

        printf (" Enter the string: ");

        gets(str1);

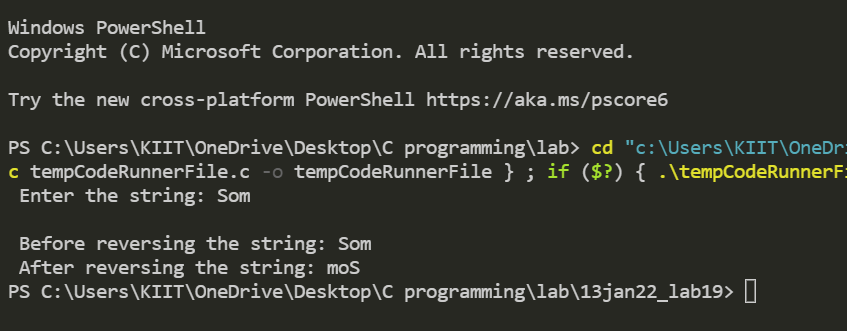
        printf (" \n Before reversing the string: %s \n", str1);

        revstr(str1);

        printf (" After reversing the string: %s", str1);

    }

***Output:***



1. Wap to count the numbers of vowels in the string

***Code:***

#include <stdio.h>

*int* main()

{

*int* c = 0, count = 0;

*char* s[1000];

    printf("Input a string\n");

    gets(s);

    while (s[c] != '\0')

    {

        if (s[c] == 'a' || s[c] == 'A' || s[c] == 'e' || s[c] == 'E' || s[c] == 'i' || s[c] == 'I' || s[c] == 'o' || s[c] == 'O' || s[c] == 'u' || s[c] == 'U')

            count++;

        c++;

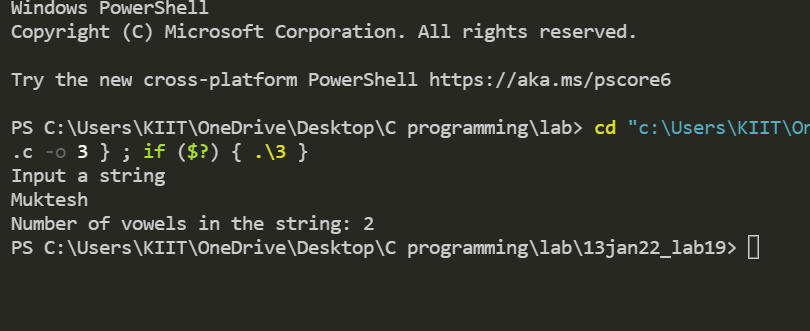
    }

    printf("Number of vowels in the string: %d", count);

    return 0;

}

***Output:***



1. Wap to find length of a string

***Code:***

#include <stdio.h>

#include <string.h>

*int* main()

{

*char* Str[100],c=0;

*int* i;

    printf("Enter the String: ");

    scanf("%s", Str);

    for (i = 0; Str[i] != '\0'; ++i)

    {

        c++;

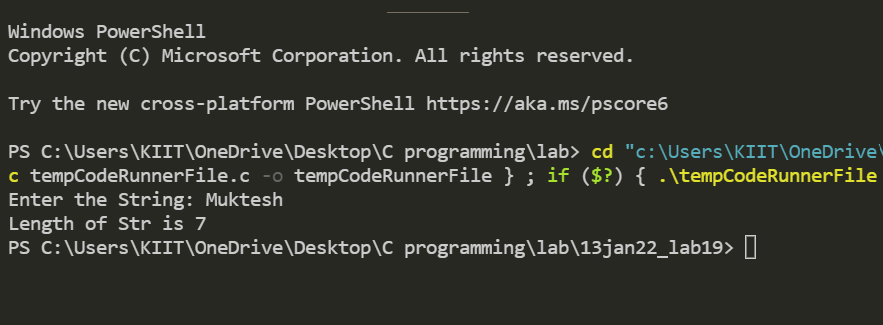
    }

    printf("Length of Str is %d", c);

    return 0;

}

***Output:***



1. Wap to find frequency of characters in a string

***Code:***

#include <stdio.h>

#include <string.h>

*int* main() {

*char* str[1000], ch;

*int* count = 0;

    printf("Enter a string: ");

    fgets(str, sizeof(str), stdin);

    printf("Enter a character to find its frequency: ");

    scanf("%c", &ch);

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

        if (ch == str[i])

            ++count;

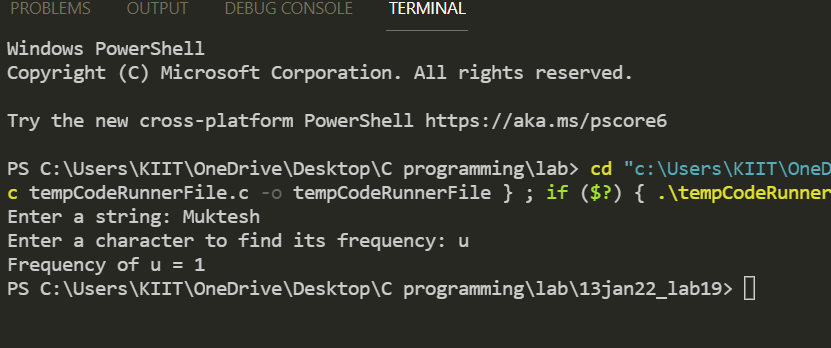
    }

    printf("Frequency of %c = %d", ch, count);

    return 0;

}

***Output:***



6.Wap to find concatenation of string

***Code:***

#include <stdio.h>

#include <string.h>

*int* length(*char* \**s*)

{

*int* k = -1;

    while (*s*[++k]);

    return k;

}

*int* main()

{

*char* s1[100], s2[100], ch;

*int* i, j;

    printf("Enter first string=");

    scanf("%s", s1);

    scanf("%c", &ch);

    printf("Enter second string=");

    scanf("%s", s2);

    i = 0;

    j = length(s1);

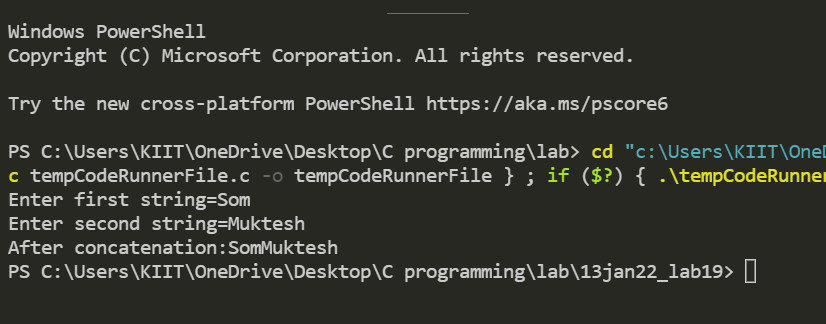
    while (s1[j++] = s2[i++]);

    printf("After concatenation:%s", s1);

    return 0;

}

***Output:***



1. wap to find copy of a string

***Code:***

#include <stdio.h>

*int* main()

{

*int* oldName[10], newName[10];

    printf("\nEnter the name: ");

    scanf("%s", oldName);

    printf("\nOriginal Name is: %s", oldName);

*int* i = 0;

    while (oldName[i] != '\0')

    {

        newName[i] = oldName[i];

        i++;

    }

    newName[i] = '\0';

    printf("\nCopied string is: %s", newName);

}

***Output:***

