**1. WAP to check if a string is substring of another. Given two strings s1 and s2, find if s1 is a substring of s2.If yes, return the index of the first occurrence, else return -1.**

**CODE –**

**#include <stdio.h>**

**#include <string.h>**

**int isSubstring(char\* s1, char\* s2)**

**{**

**int M = strlen(s1);**

**int N = strlen(s2);**

**for (int i = 0; i <= N - M; i++)**

**{**

**int j;**

**for (j = 0; j < M; j++)**

**{**

**if (s2[i + j] != s1[j])**

**{break;}**

**}**

**if (j == M)**

**{return i;}**

**}**

**return -1;**

**}**

**void main()**

**{**

**char s1[] = "for";**

**char s2[] = "geeksforgeeks";**

**int res = isSubstring(s1, s2);**

**if (res == -1)**

**printf("Not present");**

**else**

**printf("Present at index %d", res);**

**}**

**2. Write a C program that reads a sentence and prints the frequency of each of the vowels and total count of consonants.**

**CODE –**

#include <stdio.h>

void 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);

}

**3. Write a program in C to find the largest integer among three numbers using command line arguments.**

**CODE –**

**#include <stdio.h>**

**void main()**

**{**

**int a, b, c;**

**printf("Enter the numbers a, b and c: ");**

**scanf("%d %d %d", &a, &b, &c);**

**if (a >= b && a >= c)**

**printf("%d is the largest number.", a);**

**if (b >= a && b >= c)**

**printf("%d is the largest number.", b);**

**if (c >= a && c >= b)**

**printf("The largest number is: ", c);**

**}**

**4. WAP to Return maximum occurring character in an input string. Write an efficient function to return maximum occurring character in the input string e.g., if input string is “test” then function should return ‘t’.**

**CODE –**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char MaxOccurringChar(char str[])

{

    int count[256] = { 0 };

    int length = strlen(str);

    for (int i = 0; i < length; i++)

    {

    count[(int)str[i]]++;

    }

    char maxChar;

    int maxCount = 0;

    for (int i = 0; i < length; i++)

    {

    if (count[(int)str[i]] > maxCount)

    {maxCount = count[(int)str[i]];

    maxChar = str[i];}

    }

    return maxChar;

}

void main()

{

    char name[100];

    printf("Enter a string: ");

    gets(name);

    printf("Max occurring character is: %c\n",

    MaxOccurringChar(name));

}

**5. WAP program to read name and marks of n number of students from user and store them in a file. If the file previously exits, add the information to the file.**

**CODE –**

#include <stdio.h>

void main()

{

    char name[50];

    int marks,i,n;

    printf("Enter number of students: ");

    scanf("%d",&n);

    FILE \*fptr;

    fptr=(fopen("C:\\student.txt","w"));

    if(fptr==NULL)

    {

        printf("Error!");

        exit(1);

    }

    for (i=0;i<n;++i)

    {

    printf("For student%d\nEnter name: ",i+1);

    scanf("%s",name);

    printf("Enter marks: ");

    scanf("%d",&marks);

    fprintf(fptr,"\nName: %s \nMarks=%d \n",name,marks);

    }

    fclose(fptr);

}

**6. C program to write all the members of an array of structures to a file using fwrite(). Read the array from the file and display on the screen.**

**CODE –**

#include <stdio.h>

struct student

{

    char name[50];

    int height;

};

int main()

{

    struct student stud1[5], stud2[5];

    FILE \*fptr;

    int i;

    fptr = fopen("file.txt","wb");

    for(i = 0; i < 5; ++i)

    {

        fflush(stdin);

        printf("Enter name: ");

        gets(stud1[i].name);

        printf("Enter height: ");

        scanf("%d", &stud1[i].height);

    }

    fwrite(stud1, sizeof(stud1), 1, fptr);

    fclose(fptr);

    fptr = fopen("file.txt", "rb");

    fread(stud2, sizeof(stud2), 1, fptr);

    for(i = 0; i < 5; ++i)

    {

        printf("Name: %s\nHeight: %d", stud2[i].name, stud2[i].height);

    }

    fclose(fptr);

}

**7. WAP in C to find the determinant of a matrix.**

**CODE –**

#include <stdio.h>

#define N 4

void getCofactor(int mat[N][N], int temp[N][N],int p, int q, int n)

{

    int i = 0, j = 0;

    for (int row = 0; row < n; row++)

    {

        for (int col = 0; col < n; col++)

        {

            if (row != p && col != q)

            {

                temp[i][j++] = mat[row][col];

                if (j == n - 1)

                {j = 0;i++;}

            }

        }

    }

}

int determinantOfMatrix(int mat[N][N], int n)

{int D = 0;

if (n == 1)

return mat[0][0];

int temp[N][N];

int sign = 1;

for (int f = 0; f < n; f++)

{getCofactor(mat, temp, 0, f, n);

D += sign \* mat[0][f]\* determinantOfMatrix(temp, n - 1);

sign = -sign;}

return D;}

void display(int mat[N][N],

int row, int col)

{for (int i = 0; i < row; i++)

{for (int j = 0; j < col; j++)

printf(" %d", mat[i][j]);

printf("n");}}

void main()

{

    int mat[N][N] = {{1, 0, 2, -1},{3, 0, 0, 5},{2, 1, 4, -3},{1, 0, 5, 0}};

    printf("Determinant of the matrix is : %d",

    determinantOfMatrix(mat, N));

}