## Computer Programming Lab

## CEN-392

# LAB FILE

**Name :** Vicky Gupta

**Roll No. :** 20BCS070

**Branch :** Computer Engineering

**Subject :** Computer Programming Lab

# INDEX

# Computer Programming Lab

# CEN-392

# Program 1

## Code :-

```c
#include <stdio.h>

void Addition(int arr[][10], int r1, int c1, int brr[][10], int r2, int c2)
{
    if (r1 != r2 && c1 != c2)
    {
        printf("Addition Of Two Given Matrix Not Possible!\n\n");
        return;
    }
    int ans[r1][c1];
    printf("\nAddition Answer : \n");
    for (int i = 0; i < r1; i++)
    {
        for (int j = 0; j < c1; j++)
        {
            ans[i][j] = arr[i][j] + brr[i][j];
            printf("%d ", ans[i][j]);
        }
        printf("\n");
    }
}
```

```c
void Subtraction(int arr[][10], int r1, int c1, int brr[][10], int r2, int c2)
{
    if (r1 != r2 && c1 != c2)
    {
        printf("Subtraction Of Two Given Matrix Not Possible!\ns\n");
        return;
    }
    int ans[r1][c1];
    printf("\nSubtraction Answer : \n");
    for (int i = 0; i < r1; i++)
    {
        for (int j = 0; j < c1; j++)
        {
            ans[i][j] = arr[i][j] - brr[i][j];
            printf("%d ", ans[i][j]);
        }
        printf("\n");
    }
}

void Multiplication(int arr[][10], int r1, int c1, int brr[][10], int r2, int c2)
{
    if (c1 != r2)
    {
        printf("Multiplication Of Two Given Matrix Not Possible!\n\n");
        return;
    }
    int ans[r1][c2];
    printf("\nMultiplication Answer : \n");
    for (int i = 0; i < r1; i++)
    {
        for (int j = 0; j < c2; j++)
        {
            int calc = 0;
            for (int k = 0; k < c1; k++)
            {
                calc += arr[i][k] * brr[k][j];
            }
            ans[i][j] = calc;
            printf("%d ", ans[i][j]);
        }
        printf("\n");
    }
}

int Option(int arr[][10], int r1, int c1, int brr[][10], int r2, int c2)
{
    int optn;
    printf("Enter Your Choice : ");
    scanf("%d", &optn);
    switch (optn)
    {
    case 1:
```

```
            Addition(arr, r1, c1, brr, r2, c2);
            break;
        case 2:
            Subtraction(arr, r1, c1, brr, r2, c2);
            break;
        case 3:
            Multiplication(arr, r1, c1, brr, r2, c2);
            break;
        case 4:
            return 0;
        default:
            printf("Invalid Input Try Again!\n");
        }
        return 1;
    }

    void Menu()
    {
        printf("\n___Matrix Operations___\n");
        printf("1.Addition\n");
        printf("2.Subtraction\n");
        printf("3.Multiplication\n");
        printf("4.Exit\n");
    }

    int main()
    {
        system("cls");
        while (1)
        {
            int r1, c1;
            printf("Enter The Rows And Column Of The Matrix : \n");
            scanf("%d%d", &r1, &c1);
            int arr[10][10];
            for (int i = 0; i < r1; i++)
            {
                for (int j = 0; j < c1; j++)
                {
                    scanf("%d", &arr[i][j]);
                }
            }

            int r2, c2;
            printf("Enter The Rows And Column Of The Matrix : \n");
            scanf("%d%d", &r2, &c2);
            int brr[10][10];
            for (int i = 0; i < r2; i++)
            {
                for (int j = 0; j < c2; j++)
                {
                    scanf("%d", &brr[i][j]);
                }
            }
        Previous:
```

```
        Menu();
        if (!Option(arr, r1, c1, brr, r2, c2))
            break;
        int c;
        printf("\nDo You Want To Work On Previos Input Matrix [y/n] : ");
        scanf(" %c", &c);
        printf("\n");
        if (c == 'Y' || c == 'y')
            goto Previous;
    }

    return 0;
}
```

# Output :-

```
Enter The Rows And Column Of The Matrix :
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Enter The Rows And Column Of The Matrix :
4 4
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

___Matrix Operations___
1.Addition
2.Subtraction
3.Multiplication
4.Exit
Enter Your Choice : 1

Addition Answer :
2 2 3 4
5 7 7 8
9 10 12 12
13 14 15 17

Do You Want To Work On Previos Input Matrix [y/n] : y
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Exit
Enter Your Choice : 2

Subtraction Answer :
0 2 3 4
5 5 7 8
9 10 10 12
13 14 15 15

Do You Want To Work On Previos Input Matrix [y/n] : y


___Matrix Operations___
1.Addition
2.Subtraction
3.Multiplication
4.Exit
Enter Your Choice : 3

Multiplication Answer :
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

Do You Want To Work On Previos Input Matrix [y/n] : n

Enter The Rows And Column Of The Matrix :
2 2
1 2
3 4
Enter The Rows And Column Of The Matrix :
1 1
3
```

```
___Matrix Operations___
1.Addition
2.Subtraction
3.Multiplication
4.Exit
Enter Your Choice : 1
Addition Of Two Given Matrix Not Possible!


Do You Want To Work On Previos Input Matrix [y/n] : y


___Matrix Operations___
1.Addition
2.Subtraction
3.Multiplication
4.Exit
Enter Your Choice : 4
PS E:\Programming Language\Computer Lab>
```

# Computer Programming Lab

# CEN-392

# Program 2

## Code :-

```
#include <iostream>
using namespace std;
#define size 40

int InputNum()
{
    int newnum;
    cout << "Enter The Number : ";
    cin >> newnum;
    return newnum;
}

int InsertBeg(int arr[], int n)
{
```

```
    if (n == size)
    {
        cout << "Array Overflow!" << endl;
        return n;
    }

    n++;
    for (int i = n - 1; i > 0; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[0] = InputNum();
    return n;
}

int InsertEnd(int arr[], int n)
{
    if (n == size)
    {
        cout << "Array Overflow!" << endl;
        return n;
    }

    arr[n] = InputNum();
    n++;
    return n;
}

int InsertK(int arr[], int n)
{
    if (n == size)
```

```
    {
        cout << "Array Overflow!" << endl;
        return n;
    }
    int k;
    cout << "Enter The Position [ Accoring To 1 Based Indexing ] : "
;
    cin >> k;
    if (k > n + 1)
    {
        cout << "Invalid Input!" << endl;
        return n;
    }

    n++;
    for (int i = n; i >= k; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[k - 1] = InputNum();
    return n;
}

int DeleteBeg(int arr[], int n)
{
    if (n == 0)
    {
        cout << "Array Underflow!" << endl;
        return n;
    }
    for (int i = 1; i < n; i++)
    {
```

```
        arr[i - 1] = arr[i];
    }
    n--;
    return n;
}

int DeleteEnd(int arr[], int n)
{
    if (n == 0)
    {
        cout << "Array Underflow!" << endl;
        return n;
    }
    n--;
    return n;
}

int DeleteK(int arr[], int n)
{
    if (n == 0)
    {
        cout << "Array Underflow!" << endl;
        return n;
    }
    int k;
    cout << "Enter The Position [ Accoring To 1 Based Indexing ] : "
;
    cin >> k;
    if (k > n)
    {
        cout << "Invalid Input!" << endl;
        return n;
```

```
    }
    for (int i = k; i < n; i++)
    {
        arr[i - 1] = arr[i];
    }
    n--;
    return n;
}

int DeleteMulti(int arr[], int n)
{
    if (n == 0)
    {
        cout << "Array Underflow!" << endl;
        return n;
    }
    int del;
    cout << "Enter The Element To Be Deleted : ";
    cin >> del;
    bool chk = true;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == del)
        {
            chk = false;
            for (int j = i + 1; j < n; j++)
            {
                arr[j - 1] = arr[j];
            }
            i--;
            n--;
```

```
        }
    }
    if (chk)
        cout << "No Element Found In The Array" << endl;
    return n;
}

void Print(int arr[], int n)
{
    if (n == 0)
    {
        cout << "Array Is Empty!" << endl;
        return;
    }
    cout << "Array -> ";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    cout << "    Size -> " << n;
    cout << endl;
}

void ShowMenu()
{
    cout << endl
         << "___Operations To Perform On Array___" << endl;
    cout << "1.Insert At The Beginning" << endl;
    cout << "2.Insert At The Kth Position" << endl;
    cout << "3.Insert At The End" << endl;
    cout << "4.Delete At The Beginning" << endl;
```

```
    cout << "5.Delete At The Kth Position" << endl;
    cout << "6.Delete At The End" << endl;
    cout << "7.Delete Particular Element" << endl;
    cout << "8.Exit" << endl;
    cout << "Enter Your Choice : ";
}

bool Options(int arr[], int *n)
{
    int opt;
    cin >> opt;
    if (opt >= 1 && opt <= 8)
    {
        cout << endl
             << "Operation " << opt << " Is Seleceted." << endl;
    }
    switch (opt)
    {
    case 1:
        *n = InsertBeg(arr, *n);
        break;
    case 2:
        *n = InsertK(arr, *n);
        break;
    case 3:
        *n = InsertEnd(arr, *n);
        break;
    case 4:
        *n = DeleteBeg(arr, *n);
        break;
    case 5:
```

```
            *n = DeleteK(arr, *n);
            break;
        case 6:
            *n = DeleteEnd(arr, *n);
            break;
        case 7:
            *n = DeleteMulti(arr, *n);
            break;
        case 8:
            return 0;
            break;
        default:
            cout << "Invalid Input!" << endl;
        }

        return 1;
}

int main()
{
    system("cls");
    int n;
    cout << "Enter The Size Of The Array : ";
    cin >> n;
    int arr[size];
    cout << "Enter The Element Of Array : ";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
```

```
    while (true)
    {
        ShowMenu();
        if (!Options(arr, &n))
        {
            break;
        }
        Print(arr, n);
        cout << endl;
    }
    cout<<"Exiting..."<<endl;
    return 0;
}
```

# Output :-

```
Enter The Size Of The Array : 5
Enter The Element Of Array : 2 3 5 6 7

___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 1

Operation 1 Is Seleceted.
Enter The Number : 1
Array -> 1 2 3 5 6 7     Size -> 6


___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 2

Operation 2 Is Seleceted.
Enter The Position [ Accoring To 1 Based Indexing ] : 4
Enter The Number : 4
Array -> 1 2 3 4 5 6 7     Size -> 7
```

```
___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 3

Operation 3 Is Seleceted.
Enter The Number : 8
Array -> 1 2 3 4 5 6 7 8     Size -> 8


___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 4

Operation 4 Is Seleceted.
Array -> 2 3 4 5 6 7 8     Size -> 7


___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 5

Operation 5 Is Seleceted.
Enter The Position [ Accoring To 1 Based Indexing ] : 3
Array -> 2 3 5 6 7 8     Size -> 6
```

```
___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 6

Operation 6 Is Seleceted.
Array -> 2 3 5 6 7     Size -> 5


___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 7

Operation 7 Is Seleceted.
Enter The Element To Be Deleted : 5
Array -> 2 3 6 7     Size -> 4


___Operations To Perform On Array___
1.Insert At The Beginning
2.Insert At The Kth Position
3.Insert At The End
4.Delete At The Beginning
5.Delete At The Kth Position
6.Delete At The End
7.Delete Particular Element
8.Exit
Enter Your Choice : 8

Operation 8 Is Seleceted.
Exiting...
PS D:\Study Material\2nd Year Notes\My Notes\Computer Lab\Program 2\Program>
```

# Computer Programming Lab

# CEN-392

# Program 3

## Code :-

```
#include<iostream>
using namespace std;

char Decimal_Hexadecimal_values(int n)
{
    if (n < 10)
    {
        return n + 48;
    }
    else
    {
        return 55 + n;
    }
}

int Hexadecimal_Decimal_Value(char ch)
```

```
{
    if (ch >= '0' && ch <= '9')
    {
        return ch - '0';
    }
    else
    {
        return ch - 55;
    }
}

void Decimal_Hexadecimal()
{
    cout << endl;
    cout << "Decimal To Hexadecimal Selected..." << endl;
    cout << "Enter A Decimal Number : ";
    int n;
    cin >> n;
    char str[20];
    int i = 0, num = n;

    while (n > 0)
    {
        char ch = Decimal_Hexadecimal_values(n % 16);
        str[i++] = ch;
        n /= 16;
    }
    str[i] = '\0';

    cout << "Hexadecimal Value : ";
    for (int j = i - 1; j >= 0; j--)
    {
        cout << str[j];
    }
    cout << endl;
}

void Hexadecimal_Decimal()
{
```

```
    cout << endl;
    cout << "Hexadecimal To Decimal Selected..." << endl;
    cout << "Enter A Hexadecimal Number : ";
    char str[20];
    fflush(stdin);
    gets(str);
    int len = 0;
    while (str[len] != '\0')
        len++;
    int Decimal = 0, pwr = 1;
    for (int i = len - 1; i >= 0; i--)
    {
        Decimal += pwr * Hexadecimal_Decimal_Value(str[i]);
        pwr *= 16;
    }
    cout << "Decimal Value : ";
    cout << Decimal << endl
         << endl;

}

void Menu()
{
    cout << endl
         << endl;
    cout << "_______Conversion_______" << endl;
    cout << "1.Decimal To Hexadecimal" << endl;
    cout << "2.Hexadecimal To Decimal" << endl;
    cout << "3.Exit" << endl;
    cout << "Enter Your Choice : ";
}

bool Options()
{
    int opt;
    fflush(stdin);
    cin >> opt;
    switch (opt)
    {
```

```
        case 1:
            Decimal_Hexadecimal();
            break;
        case 2:
            Hexadecimal_Decimal();
            break;
        case 3:
            return 0;
        default:
            cout << "\nInvalid Choice!\nTry Again! " << endl;
        }
        return 1;
    }

    int main()
    {
        system("cls");
        while (true)
        {
            Menu();
            if (!Options())
                break;
        }
        cout << "Exiting..." << endl;
        return 0;
    }
```

# Output :-

```
_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 1

Decimal To Hexadecimal Selected...
Enter A Decimal Number : 12315
Hexadecimal Value : 301B


_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 2

Hexadecimal To Decimal Selected...
Enter A Hexadecimal Number : 301B
Decimal Value : 12315



_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 1

Decimal To Hexadecimal Selected...
Enter A Decimal Number : 43542
Hexadecimal Value : AA16
```

```
_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 1

Decimal To Hexadecimal Selected...
Enter A Decimal Number : 43542
Hexadecimal Value : AA16


_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 2

Hexadecimal To Decimal Selected...
Enter A Hexadecimal Number : AA16
Decimal Value : 43542



_______Conversion_______
1.Decimal To Hexadecimal
2.Hexadecimal To Decimal
3.Exit
Enter Your Choice : 3
Exiting...

D:\Study Material\2nd Year Notes\My Notes\Computer Lab\Program 3\Program>
```

# Computer Programming Lab

# CEN-392

# Program 4

## Code :-

```
#include <iostream>
using namespace std;

void Highest_Mark(float stu_data[20][6], int n)
{
    cout << endl
         << "Task 2 Is Selected."
         << endl;
    float Max_Sub[3][2] = {0}; //roll no and marks
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (Max_Sub[j][1] < stu_data[i][j + 2])
            {
                Max_Sub[j][0] = stu_data[i][0];
                Max_Sub[j][1] = stu_data[i][j + 2];
            }
```

```
        }
    }
    cout << "Roll No  |     Marks       |   Subject " << endl;
    for (int j = 0; j < 3; j++)
    {
        cout << Max_Sub[j][0] << "             " << Max_Sub[j][1] <<"
              "<< j+1<<endl;
    }
}

void Highest_Percentage(float stu_data[20][6], int n)
{
    cout << endl
         << "Task 3 Is Selected."
         << endl;
    float Max_Perc[2][3] = {0};// roll no age and percentage
    for (int i = 0; i < n; i++)
    {
        if (Max_Perc[0][2] < stu_data[i][5])
        {
            Max_Perc[0][0] = stu_data[i][0];
            Max_Perc[0][1] = stu_data[i][1];
            Max_Perc[0][2] = stu_data[i][5];
        }
    }
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        if (Max_Perc[0][2] == stu_data[i][5])
            count++;
    }
    cout << "Roll No" << endl;
    if (count == 1)
    {
        cout << Max_Perc[0][0] << endl;
        return;
    }

    for (int i = 0; i < n; i++)
    {
        if (Max_Perc[0][2] == stu_data[i][5] && Max_Perc[0][0] != stu_
data[i][0])
        {
            Max_Perc[1][2] = stu_data[i][5];
            Max_Perc[1][0] = stu_data[i][0];
```

```
            Max_Perc[1][1] = stu_data[i][1];
        }
    }
    if (Max_Perc[0][1] == Max_Perc[1][1])
    {
        cout << Max_Perc[0][0] << endl;
        cout << Max_Perc[1][0] << endl;
    }
    else
    {
        if (Max_Perc[0][1] < Max_Perc[1][1])
        {
            cout << Max_Perc[0][1] << endl;
        }
        else
        {
            cout << Max_Perc[1][1] << endl;
        }
    }
}

void Show_Data(float stu_data[20][6], int n)
{
    cout << endl
         << "Task 1 Is Selected."
         << endl;
    cout << "Roll No  |  Percentage " << endl;
    for (int i = 0; i < n; i++)
    {
        cout << int(stu_data[i][0]) << "            ";
        cout << stu_data[i][5]<<"%"<< endl;
    }
}

void Menu()
{
    cout << endl
         << endl
         << "_____________________Task To Perform_____________________
_" << endl;
    cout << "1.Show Percentage of All Student Along With Thier Roll No
." << endl;
    cout << "2.Highest Marks In Each Subject Along With Roll No." << e
ndl;
    cout << "3.Highest Percentage." << endl;
```

```
    cout << "4.Exit" << endl;
    cout << "Enter Your Choices : ";
}

bool Option(float stu_data[20][6], int n)
{
    int opt;
    cin >> opt;
    switch (opt)
    {
    case 1:
        Show_Data(stu_data, n);
        break;

    case 2:
        Highest_Mark(stu_data, n);
        break;
    case 3:
        Highest_Percentage(stu_data, n);
        break;
    case 4:
        return 0;
    default:
        cout << "Invalid_Entry" << endl;
    }
    return 1;
}

int main()
{
    system("cls");
    cout << "_________Vicky Gupta 20BCS070__________" << endl
         << endl;
    int n;
    cout << "Enter The No Of Students In Class : ";
    cin >> n;
    float stu_data[20][6];
    for (int i = 0; i < n; i++)
    {
        cout << endl
             << "Enter The Info Of Student " << i + 1 << "." << endl;
        cout << "Enter The Roll No Of The Student : ";
        cin >> stu_data[i][0];
        cout << "Enter The Age Of The Student : ";
        cin >> stu_data[i][1];
```

```
        float prcntge = 0;
        for (int j = 2; j < 5; j++)
        {
            cout << "Enter The Marks Of The Subject " << j - 1 << " :
";
            cin >> stu_data[i][j];
            prcntge += stu_data[i][j];
        }
        stu_data[i][5] = prcntge / 3;
    }

    while (true)
    {
        Menu();
        if (!Option(stu_data, n))
            break;
    }
    cout << "Exiting..." << endl;
    return 0;
}
```

# Output :-

```
_________Vicky Gupta 20BCS070__________

Enter The No Of Students In Class : 3

Enter The Info Of Student 1.
Enter The Roll No Of The Student : 1
Enter The Age Of The Student : 17
Enter The Marks Of The Subject 1 : 56
Enter The Marks Of The Subject 2 : 76
Enter The Marks Of The Subject 3 : 42

Enter The Info Of Student 2.
Enter The Roll No Of The Student : 2
Enter The Age Of The Student : 16
Enter The Marks Of The Subject 1 : 54
Enter The Marks Of The Subject 2 : 42
Enter The Marks Of The Subject 3 : 78

Enter The Info Of Student 3.
Enter The Roll No Of The Student : 3
Enter The Age Of The Student : 17
Enter The Marks Of The Subject 1 : 76
Enter The Marks Of The Subject 2 : 56
Enter The Marks Of The Subject 3 : 34


_____________________Task To Perform_____________________
1.Show Percentage of All Student Along With Thier Roll No.
2.Highest Marks In Each Subject Along With Roll No.
3.Highest Percentage.
4.Exit
Enter Your Choices : 1

Task 1 Is Selected.
Roll No  |  Percentage
1            58%
2            58%
3            55.3333%
```

```
____________________Task To Perform____________________
1.Show Percentage of All Student Along With Thier Roll No.
2.Highest Marks In Each Subject Along With Roll No.
3.Highest Percentage.
4.Exit
Enter Your Choices : 3

Task 3 Is Selected.
Roll No
16


____________________Task To Perform____________________
1.Show Percentage of All Student Along With Thier Roll No.
2.Highest Marks In Each Subject Along With Roll No.
3.Highest Percentage.
4.Exit
Enter Your Choices : 2

Task 2 Is Selected.
Roll No  |     Marks       |   Subject
3             76              1
1             76              2
2             78              3


____________________Task To Perform____________________
1.Show Percentage of All Student Along With Thier Roll No.
2.Highest Marks In Each Subject Along With Roll No.
3.Highest Percentage.
4.Exit
Enter Your Choices : 4
Exiting...
```

# Computer Programming Lab

# CEN-392

# Program 5

## Code :-

```
#include <iostream>
using namespace std;

int matrix[10][10];
int n=0, m=0;

void Print_Matrix()
{
    cout << endl
         << "____Matrix___" << endl<<endl;;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << matrix[i][j] << "   ";
        }
        cout << endl;
```

```
    }
}

void Print_Helical()
{
    if(n==0||m==0)
    {
        cout<<endl<<"Matrix Input First!"<<endl;
        return;
    }
    cout << endl
         << "Operation Helical Order Is Selected." << endl;

    Print_Matrix();

    int rows = 0, rowe = n - 1, cols = 0, cole = m - 1;

    int total = n * m;

    cout << endl
         << "Helical Order Of The Input Matrix : " << endl;

    while (total > 0)
    {
        for (int i = cols; i <= cole && total-- > 0; i++)
        {
            cout << matrix[rows][i] << " ";
        }
        rows++;

        for (int i = rows; i <= rowe && total-- > 0; i++)
        {
            cout << matrix[i][cole] << " ";
        }
        cole--;

        for (int i = cole; i >= cols && total-- > 0; i--)
        {
            cout << matrix[rowe][i] << " ";
        }
        rowe--;
```

```
        for (int i = rowe; i >= rows && total-- > 0; i--)
        {
            cout << matrix[i][cols] << " ";
        }
        cols++;
    }
    cout << endl;
}

void Input()
{
    cout << endl
         << "Opertion New Matrix Input Is Selected." << endl;
    cout << "Number Of Rows : ";
    cin >> n;

    cout << "Number Of Column : ";
    cin >> m;

    cout << "Enter The Element Of The Matrix : " << endl;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> matrix[i][j];
        }
    }
}

void Menu()
{
    cout << endl
         << "___Task To Performs___" << endl;
    cout << "1.New Matrix Input." << endl;
    cout << "2.Helical Order." << endl;
    cout << "3.Exit." << endl;
    cout << "Enter Your Choice : ";
}

bool Options()
{
```

```
    int opt;
    cin >> opt;
    switch (opt)
    {
    case 1:
        Input();
        break;
    case 2:
        Print_Helical();
        break;
    case 3:
        return 0;
    default:
        cout << "Invalid Input!\nTry Again!" << endl;
        break;
    }
    return 1;
}

int main()
{
    system("cls");
    cout << "__Vicky_Gupta_20BCS070__" << endl;
    while (true)
    {
        Menu();
        if (!Options())
            break;
    }
    cout << endl
         << "Exiting..." << endl;

    return 0;
}
```

# Output :-

```
__Vicky_Gupta_20BCS070__

___Task To Performs___
1.New Matrix Input.
2.Helical Order.
3.Exit.
Enter Your Choice : 1

Opertion New Matrix Input Is Selected.
Number Of Rows : 4
Number Of Column : 5
Enter The Element Of The Matrix :
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
25 26 27 28 29

___Task To Performs___
1.New Matrix Input.
2.Helical Order.
3.Exit.
Enter Your Choice : 2

Operation Helical Order Is Selected.

____Matrix___

10   11   12   13   14
15   16   17   18   19
20   21   22   23   24
25   26   27   28   29

Helical Order Of The Input Matrix :
10 11 12 13 14 19 24 29 28 27 26 25 20 15 16 17 18 23 22 21

___Task To Performs___
1.New Matrix Input.
2.Helical Order.
3.Exit.
Enter Your Choice : 3

Exiting...
```

# Computer Programming Lab

# CEN-392

# Program 6

## Code :-

```
#include <iostream>
using namespace std;

int strlen(char str[])
{
    int i = 0;
    while (str[i] != '\0')
        i++;
    return i;
}

void strlength()
{
    cout<<"\nString Length Operation Is Selected.\n";
    char str[100];
    fflush(stdin);
```

```
    cout << "Enter The String : ";
    cin.getline(str, 100);
    int slen = strlen(str);
    cout << "\nString Length : " << slen << "\n";
}

void strrev()
{
    cout<<"\nString Reverse Operation Is Selected.\n";
    char str[100];
    fflush(stdin);
    cout << "Enter The String : ";
    cin.getline(str, 100);

    int slen = strlen(str);
    for (int i = 0; i < slen / 2; i++)
    {
        char ch = str[i];
        str[i] = str[slen - i - 1];
        str[slen - i - 1] = ch;
    }
    cout << "\nReversed String : " << str << "\n";
}

void strcpy()
{
    cout<<"\nString Copy Operation Is Selected.\n";
    char str1[100], str2[100];
    fflush(stdin);
    cout << "Enter The String : ";
    cin.getline(str2, 100);
    int s2len = strlen(str2);
    for (int i = 0; i < s2len; i++)
    {
        str1[i] = str2[i];
    }
    str1[s2len] = '\0';
    cout << "\nString Is Copied : " << str1 << "\n";
}
```

```
void strcmp()
{
    cout<<"\nString Compare Operation Is Selected.\n";
    char str1[100], str2[100];
    fflush(stdin);
    cout << "Enter The String_1 : ";
    cin.getline(str1, 100);
    cout << "Enter The String_2 : ";
    cin.getline(str2, 100);
    int s1len = strlen(str1);
    int s2len = strlen(str2);
    if (s1len != s2len)
    {
        cout << "\n'"<<str1<<"' And '"<<str2<<"' Are Not
Same\n";
        return;
    }
    for (int i = 0; i < s1len; i++)
    {
        if (str1[i] != str2[i])
        {
            cout << "\n'"<<str1<<"' And '"<<str2<<"' Are Not
Same\n";
            return;
        }
    }
    cout <<"\n '" <<str1<<"' And '"<<str2<<"' Are Same\n";
}

void strcat()
{
    cout<<"\nString Concatation Operation Is Selected.\n";
    char str1[100], str2[100];
    fflush(stdin);
    cout << "Enter The String_1 : ";
    cin.getline(str1, 100);
    cout << "Enter The String_2 : ";
    cin.getline(str2, 100);
```

```
    int s1len = strlen(str1);
    int s2len = strlen(str2);

    for (int i = 0; i < s2len; i++)
    {
        str1[i + s1len] = str2[i];
    }
    str1[s1len + s2len] = '\0';
    cout << "\nConcated String : " << str1 << "\n";
}

void isPalindrome()
{
    cout<<"\nString Pallindrome Operation Is Selected.\n";
    char str[100];
    fflush(stdin);
    cout << "Enter The String : ";
    cin.getline(str, 100);
    int slen = strlen(str);
    for (int i = 0; i < slen / 2; i++)
    {
        if (str[i] != str[slen - i - 1])
        {
            cout <<str<<" Is Not A Pallindrome\n";
            return;
        }
    }
    cout <<"\n"<<str<<" Is Pallindrome\n";
}

void Seach()
{
    cout<<"\nString Search Substring Operation Is
Selected.\n";
    char str1[100], str2[100];
    fflush(stdin);
    cout << "Enter The String_1 : ";
    cin.getline(str1, 100);
```

```
    cout << "Enter The String_2 : ";
    cin.getline(str2, 100);
    int s1len = strlen(str1);
    int s2len = strlen(str2);
    if (s1len < s2len)
    {
        cout << "Substring Not Found\n";
        return;
    }
    bool check=false;
    for (int i = 0; i < s1len; i++)
    {
        int j = 0;
        for (; j < s2len && i + j < s1len; j++)
        {
            if (str1[i + j] != str2[j])
            {
                break;
            }
        }
        if (j == s2len)
        {
            if(!check)cout<<"\nSubstring Found!\n";
            cout << "Index : " << i << "\n";
            check=true;
        }
    }
    if(!check)
    cout << "\nSubstring Not Found!\n";
}

void Menu()
{
    cout << "\n\n____String_Operations____\n";
    cout << "1.Length\n";
    cout << "2.Reverse\n";
    cout << "3.Copy\n";
    cout << "4.Compare\n";
    cout << "5.Concatnate\n";
```

```
    cout << "6.Pallindrome\n";
    cout << "7.Search Substring\n";
    cout << "8.Exit\n";
    cout << "Enter Your Choice : ";
}

void AnsBar()
{
    cout<<"_____________________________________________________
_____________\n";
}

bool Options()
{

    int opt;
    fflush(stdin);
    cin >> opt;
    AnsBar();
    switch (opt)
    {
    case 1:
        strlength();
        break;
    case 2:
        strrev();
        break;
    case 3:
        strcpy();
        break;
    case 4:
        strcmp();
        break;
    case 5:
        strcat();
        break;
    case 6:
        isPalindrome();
        break;
```

```
        case 7:
            Seach();
            break;
        case 8:
        cout<<"Exit Is Selected.\n";
        AnsBar();
            return 0;
        default:
            cout << "Invalid Entry!\n";
            break;
        }
        AnsBar();
        return 1;
    }

    int main()
    {
        system("cls");
        cout << "____Vicky_Gupta_20BCS070____\n";
        while (true)
        {
            Menu();
            if (!Options())
                break;
        }
        cout << "Exiting...\n";
        return 0;
    }
```

# Output :-

```
____Vicky_Gupta_20BCS070____


____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 1
_____________________________________________________________

String Length Operation Is Selected.
Enter The String : Vicky Gupta

String Length : 11
_____________________________________________________________


____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 2
_____________________________________________________________

String Reverse Operation Is Selected.
Enter The String : Vicky

Reversed String : ykciV
_____________________________________________________________
```

```
____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 3
_____________________________________________________________

String Copy Operation Is Selected.
Enter The String : Hello World

String Is Copied : Hello World
_____________________________________________________________


____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 4
_____________________________________________________________

String Compare Operation Is Selected.
Enter The String_1 : Vicky
Enter The String_2 : Vicky

 'Vicky' And 'Vicky' Are Same
_____________________________________________________________
```

```
____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 6
_____________________________________________________________

String Pallindrome Operation Is Selected.
Enter The String : NitiN

NitiN Is Pallindrome
_____________________________________________________________


____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 7
_____________________________________________________________

String Search Substring Operation Is Selected.
Enter The String_1 : Hey Someone Is Here, Hello
Enter The String_2 : He

Substring Found!
Index : 0
Index : 15
Index : 21
_____________________________________________________________
```

```
____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 5
_____________________________________________________________

String Concatation Operation Is Selected.
Enter The String_1 : Computer
Enter The String_2 : Engineer

Concated String : Computer Engineer
_____________________________________________________________


____String_Operations____
1.Length
2.Reverse
3.Copy
4.Compare
5.Concatnate
6.Pallindrome
7.Search Substring
8.Exit
Enter Your Choice : 8
_____________________________________________________________
Exit Is Selected.
_____________________________________________________________
Exiting...
```

# Computer Programming Lab

# CEN-392

# Program 7

## Code :-

```
#include <iostream>
using namespace std;

int Length(char str[])
{
    int len = 0;
    while (str[len] != '\0')
        len++;
    return len;
}

int Vowels(char str[])
{
    int len = Length(str);
    int vowels = 0;
    for (int i = 0; i < len; i++)
```

```
    {

        if (str[i] == 'a' || str[i] == 'A' || str[i] == 'e'
|| str[i] == 'E' ||
            str[i] == 'i' || str[i] == 'I' || str[i] == 'o'
|| str[i] == 'O' ||
            str[i] == 'u' || str[i] == 'U')
            vowels++;
    }
    return vowels;
}

void Remove_Extra_Space(char paragraph[])
{
    int length = Length(paragraph);
    for (int i = 1; i < length; i++)
    {
        if (paragraph[i] == ' ' && paragraph[i] ==
paragraph[i - 1])
        {
            for (int j = i; j < length; j++)
            {
                paragraph[j - 1] = paragraph[j];
            }
            length--;
            i--;
            paragraph[length] = '\0';
        }
    }
}

int Count_Spaces(char paragraph[])
{
    int length = Length(paragraph);
    int spaces = 0;
    for (int i = 0; i < length; i++)
    {
        if (paragraph[i] == ' ')
            spaces++;
```

```
    }
    return spaces;
}

int Count_Tabs(char paragraph[])
{
    int length = Length(paragraph);
    int tabs = 0;
    for (int i = 0; i < length; i++)
    {
        if (paragraph[i] == 9)
            tabs++;
    }
    return tabs;
}

int Count_Sentences(char paragraph[])
{
    int length = Length(paragraph);
    int sentence = 0;
    for (int i = 0; i < length; i++)
    {

        if (paragraph[i] == '.')
            sentence++;
    }
    return sentence;
}

int Count_Lines(char paragraph[])
{
    int length = Length(paragraph);
    int lines = 1;
    for (int i = 0; i < length; i++)
    {
        if (paragraph[i] == '\n')
            lines++;
    }
    return lines;
```

```
}

void Bars()
{
    cout << "-----------------------------------------------
------------------------" << endl;
}
int main()
{
    system("cls");
    cout << "_____Vicky_Gupta_20BCS070____" << endl
         << endl;
    char paragraph[300];
    cout << "Enter A Paragraph : " << endl;
    cin.getline(paragraph, 300, '$');

    cout << endl;
    int vowels = Vowels(paragraph);
    int length = Length(paragraph);
    int spaces = Count_Spaces(paragraph);
    int tabs = Count_Tabs(paragraph);
    int lines = Count_Lines(paragraph);
    int sentence = Count_Sentences(paragraph);

    Bars();
    cout << "No Of Spaces : " << spaces << endl;
    cout << "No Of Tabs : " << tabs << endl;
    cout << "No Of Sentence : " << sentence << endl;
    cout << "No Of Lines : " << lines << endl;
    cout << "No Of Vowels : " << vowels << endl
         << endl;
    Bars();
    cout << "Extra Spaces Removed : " << endl;

    Remove_Extra_Space(paragraph);
    cout << paragraph << endl;
    Bars();
    return 0;
}
```

# Output :-

```
_____Vicky_Gupta_20BCS070_____

Enter A Paragraph :
My Name Is Vicky Gupta. And My Branch Is     Computer   Engineering.
I Like To Play     Football.$

------------------------------------------------------------------------
No Of Spaces : 21
No Of Tabs : 1
No Of Sentence : 3
No Of Lines : 2
No Of Vowels : 25

------------------------------------------------------------------------
Extra Spaces Removed :
My Name Is Vicky Gupta. And My Branch Is Computer       Engineering.
I Like To Play Football.
------------------------------------------------------------------------
```

# Computer Programming Lab

# CEN-392

# Program 8

## Code :-

```
#include <iostream>
using namespace std;

int Leap_Year_Count(int year, int month)
{
    if (month <= 2)
        year--;
    return year / 4 - year / 100 + year / 400;
}

int Month_Days_Count(int month)
{
    int Day_In_Month[12] = {31, 28, 31, 30, 31, 30,
                            31, 31, 30, 31, 30, 31};
    int Days = 0;
    for (int i = 0; i < month - 1; i++)
```

```
        Days += Day_In_Month[i];
    return Days;
}

int main()
{
    system("cls");
    cout << "_____Vicky_Gupta_20BCS070_____" << endl;
    cout << "Enter The Two Dates Sepearted By Special
Characters : \n";
    char dates[100];
    gets(dates);
    int len = 0;
    while (dates[len] != '\0')
        len++;

    int day[2] = {0}, month[2] = {0}, year[2] = {0};

    int no_of_dates = 0, it = 0;

    while (no_of_dates < 2)
    {
        int multi = 10;

        while (dates[it] >= '0' && dates[it] <= '9')
        {
            day[no_of_dates] += (dates[it] - '0') * multi;
            multi /= 10;
            it++;
        }
        it++;
        multi = 10;
        while (dates[it] >= '0' && dates[it] <= '9')
        {
            month[no_of_dates] += (dates[it] - '0') * multi;
            multi /= 10;
            it++;
        }
        it++;
```

```
        multi = 1000;
        while (dates[it] >= '0' && dates[it] <= '9' && it <
len)
        {
            year[no_of_dates] += (dates[it] - '0') * multi;
            multi /= 10;
            it++;
        }
        it++;
        no_of_dates++;
    }
    long long Days_Till_first = (long long)year[0] * 365;

    Days_Till_first += Month_Days_Count(month[0]);
    Days_Till_first += Leap_Year_Count(year[0], month[0]);
    Days_Till_first += day[0];

    long long Days_Till_Second = (long long)year[1] * 365;
    Days_Till_Second += day[1];
    Days_Till_Second += Month_Days_Count(month[1]);
    Days_Till_Second += Leap_Year_Count(year[1], month[1]);

    long long Count_Day = abs(Days_Till_first -
Days_Till_Second);
    cout << "The No Of Days Between Given Dates : ";
    cout << Count_Day << endl;
    return 0;
}
```

# Output :-

```
_____Vicky_Gupta_20BCS070_____
Enter The Two Dates Sepearted By Special Characters :
08-11-2021,15-01-1932
The No Of Days Between Given Dates : 32805
```

# Computer Programming Lab

# CEN-392

# Program 9

## Code :-

```
#include <iostream>
using namespace std;

int Length(char str[])
{
    int len = 0;
    while (str[len] != '\0')
        len++;
    return len;
}

void Replace(char txt[], char find[], char replace[], int
indx)
{
    int len1, len2, len3;
    len1 = Length(txt);
```

```
    len2 = Length(find);
    len3 = Length(replace);
    int i;
    for (i = indx; i < len1 && txt[i + len2] != '\0'; i++)
// Removing The Find String
    {
        txt[i] = txt[i + len2];
    }
    txt[i] = '\0';
    char temp[200];
    len1 = Length(txt);
    for (i = 0; i < len1; i++) // Copying The String To The
Temp From indx->end
    {
        temp[i] = txt[i + indx];
    }
    temp[i] = '\0';
    for (i = 0; i < len3; i++) // now adding the replace
string from indx
    {
        txt[indx + i] = replace[i];
    }
    indx += i;
    int len4 = Length(temp);
    for (i = 0; i < len4; i++) // now we add the temp string
after the replace string
    {
        txt[indx + i] = temp[i];
    }
    txt[indx + i] = '\0';
}

void Find(char txt[], char find[], char replace[], bool
Multi)
{
    if (Multi)
        cout << "Multi Replace Selected...\n\n";
    else
        cout << "Multi Replace Selected...\n\n";
```

```
fflush(stdin);
cout << "Enter The Word To Find : ";
cin.getline(find, 200);
cout << "Enter The Word To Replace : ";
cin.getline(replace, 200);
int len1, len2;
len1 = Length(txt);
len2 = Length(find);
int i, j;
bool check = false;
for (i = 0; i <= len1 - len2; i++)
{
    for (j = 0; j < len2; j++)
    {
        if (txt[i + j] != find[j])
            break;
    }
    if (j == len2)
    {
        check = true;
        if (Multi)
        {
            Replace(txt, find, replace, i);
        }
        else
        {
            Replace(txt, find, replace, i);
            break;
        }
    }
}
if (!check)
{
    cout << "\nWord Not Found In The Text\n"
         << endl;
}
else
{
```

```
        cout << "\nThe Modified String :\n";
        cout << txt << endl;
    }
}

void Input_Again(char txt[], char find[], char replace[])
{
    cout << "Input Again...\n\n";
    cout << "Enter The Text : \n";
    fflush(stdin);
    cin.getline(txt, 200);
}

void Add_Bars()
{
    cout << "-----------------------------------------------
-----------------------" << endl;
}

void Menu()
{
    cout << "____Operations_____" << endl;
    cout << "1.Single Replace" << endl;
    cout << "2.Multiple Replace" << endl;
    cout << "3.Input Again" << endl;
    cout << "4.Exit" << endl;
    cout << "Enter Your Choice : ";
}

bool Options(char txt[], char find[], char replace[])
{
    int opt;
    fflush(stdin);
    cin >> opt;
    Add_Bars();
    switch (opt)
    {
    case 1:
        Find(txt, find, replace, 0);
```

```
            break;
        case 2:
            Find(txt, find, replace, 1);
            break;
        case 3:
            Input_Again(txt, find, replace);
            break;
        case 4:
            return 0;
        default:
            cout << "Incorrect Input!\nTry Again!" << endl;
            break;
        }
        Add_Bars();
        return 1;
    }

    int main()
    {
        system("cls");
        cout << "___Vicky_Gupta_20BCS070___\n\n";
        char txt[200], find[200], replace[200];
        cout << "Enter The Text : \n";
        cin.getline(txt, 200);
        cout << "\n\n";
        while (true)
        {
            Menu();
            if (!Options(txt, find, replace))
                break;
        }
        cout << "Exiting..." << endl;
        Add_Bars();
        return 0;
    }
```

# Output :-

```
___Vicky_Gupta_20BCS070___

Enter The Text :
My Name Is Vicky And My University Name Is Jamia University


____Operations_____
1.Single Replace
2.Multiple Replace
3.Input Again
4.Exit
Enter Your Choice : 1
-----------------------------------------------------------------
Multi Replace Selected...

Enter The Word To Find : Vicky
Enter The Word To Replace : *****

The Modified String :
My Name Is ***** And My University Name Is Jamia University
-----------------------------------------------------------------
____Operations_____
1.Single Replace
2.Multiple Replace
3.Input Again
4.Exit
Enter Your Choice : 2
-----------------------------------------------------------------
Multi Replace Selected...

Enter The Word To Find : University
Enter The Word To Replace : College

The Modified String :
My Name Is ***** And My College Name Is Jamia College
-----------------------------------------------------------------
____Operations_____
1.Single Replace
2.Multiple Replace
3.Input Again
4.Exit
Enter Your Choice : 4
-----------------------------------------------------------------
Exiting...
-----------------------------------------------------------------
```

# Computer Programming Lab

# CEN-392

# Program 10

## Code :-

```
#include <iostream>
using namespace std;

void Largest_Subarray_Min_Sum(int n, int arr[])
{
    int minsum = arr[0], left = 0, right = 0;
    for (int i = 0; i < n; i++)
    {
        int sum = 0, len = 0;
        for (int j = i; j < n; j++)
        {
            sum += arr[j];
            if (minsum >= sum)
            {
                minsum = sum;
                if (right - left < j - i)
```

```
                {
                    left = i;
                    right = j;
                }
            }
        }
    }
    cout << "Minimum Sum : " << minsum << endl
         << endl;
    cout << "Largest Subarray : \n";
    for (int i = left; i <= right; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void Smallest_Subarray_With_Largest_Sum(int n, int arr[])
{
    int maxsum = arr[0], left = 0, right = n;
    for (int i = 0; i < n; i++)
    {
        int sum = 0, len = 0;
        for (int j = i; j < n; j++)
        {
            sum += arr[j];
            if (maxsum <= sum)
            {
                if (right - left > j - i||maxsum!=sum)
                {
                    left = i;
                    right = j;
                }
                maxsum = sum;
            }
        }
    }
    cout << "Maximum Sum : " << maxsum << endl
         << endl;
```

```
    cout << "Minimum Subarray : \n";
    for (int i = left; i <= right; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{
    system("cls");
    cout << "___Vicky_Gupta_20BCS070___" << endl;
    int n;
    cout << "Enter The Length Of The Array : \n";
    cin >> n;
    int arr[n];
    cout << "Enter The Elments Of The Array : \n";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    cout << endl
         << endl;
    Largest_Subarray_Min_Sum(n, arr);
    cout << endl;
    Smallest_Subarray_With_Largest_Sum(n, arr);
    return 0;
}
```

# Output :-

```
___Vicky_Gupta_20BCS070___
Enter The Length Of The Array :
5
Enter The Elments Of The Array :
5 7 4 1 3


Minimum Sum : 1

Largest Subarray :
1

Maximum Sum : 20

Minimum Subarray :
5 7 4 1 3
```

```
___Vicky_Gupta_20BCS070___
Enter The Length Of The Array :
7
Enter The Elments Of The Array :
5 -3 1 -5 -1 7 -5


Minimum Sum : -8

Largest Subarray :
-3 1 -5 -1

Maximum Sum : 7

Minimum Subarray :
7
```

# Computer Programming Lab

# CEN-392

# Program 11

## Code :-

```c
#include <stdio.h>
#include <stdbool.h>
struct Student
{
    char Name[100];
    int Roll_No;
    float Sub_1, Sub_2, Sub_3, Percentage;
};

void Insert_Row()
{
    printf("Insert Operation Is Selected...\n");
    FILE *fptr;
    fptr = fopen("Data.txt", "a");
    if (fptr == NULL)
    {
```

```
        printf("Error In Opening File!");
        return;
    }
    struct Student Stud;
    printf("Enter The Name : ");
    fflush(stdin);
    gets(Stud.Name);
    printf("Enter The Roll No : ");
    scanf("%d", &Stud.Roll_No);
    printf("Enter The Marks Of Subject 1 : ");
    scanf("%f", &Stud.Sub_1);
    printf("Enter The Marks Of Subject 2 : ");
    scanf("%f", &Stud.Sub_2);
    printf("Enter The Marks Of Subject 3 : ");
    scanf("%f", &Stud.Sub_3);
    Stud.Percentage = (Stud.Sub_1 + Stud.Sub_2 + Stud.Sub_3)
/ 3;
    int num = 4;
    fwrite(&Stud, sizeof(Stud), 1, fptr);
    fclose(fptr);
    printf("\nRecord Inserted Successfully!\n");
}

void Display()
{
    printf("Display...\n");
    FILE *fptr;
    fptr = fopen("Data.txt", "r");
    if (fptr == NULL)
    {
        printf("Error In Opening File!\n");
        return;
    }
    struct Student Temp;
    printf("| Name | Roll No | Subject 1 | Subject 2 |
Subject 3 | Percentage |\n\n");
    while (fread(&Temp, sizeof(Temp), 1, fptr))
    {
```

```
        printf("%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n",
Temp.Name, Temp.Roll_No, Temp.Sub_1, Temp.Sub_2, Temp.Sub_3,
Temp.Percentage);
    }
    fclose(fptr);
}

void Remove_Row()
{
    FILE *fptr = NULL, *tptr = NULL;
    fptr = fopen("Data.txt", "r");
    if (fptr == NULL)
    {
        printf("Error In Opening File!\n");
        return;
    }
    tptr = fopen("temp.txt", "a");
    printf("Remove Operation Is Selected...\n");
    int Roll_No;
    printf("Enter The Roll No Of Student : ");
    scanf("%d", &Roll_No);
    struct Student Temp;
    bool Found = false;
    while (fread(&Temp, sizeof(Temp), 1, fptr))
    {
        if (Roll_No == Temp.Roll_No)
        {
            Found = true;
            continue;
        }
        fwrite(&Temp, sizeof(Temp), 1, tptr);
    }
    fclose(fptr);
    fclose(tptr);
    remove("Data.txt");
    rename("temp.txt", "Data.txt");
    if (Found == false)
        printf("\nNo Such Roll No Found In Data Base\n");
    else
```

```
        printf("\nRow Successfully Removed!\n");
}

void Update_Row()
{
    FILE *fptr = NULL, *tptr = NULL;
    fptr = fopen("Data.txt", "r");
    if (fptr == NULL)
    {
        printf("Error In Opening File!\n");
        return;
    }
    tptr = fopen("temp.txt", "a");
    printf("Update Operation Is Selected...\n");
    int Roll_No;
    printf("Enter The Roll No Of Student : ");
    scanf("%d", &Roll_No);
    struct Student Temp;
    bool Found = false;
    while (fread(&Temp, sizeof(Temp), 1, fptr))
    {
        if (Roll_No == Temp.Roll_No)
        {
            Found = true;
            printf("Enter The Name : ");
            fflush(stdin);
            gets(Temp.Name);
            printf("Enter The Roll No : ");
            scanf("%d", &Temp.Roll_No);
            printf("Enter The Marks Of Subject 1 : ");
            scanf("%f", &Temp.Sub_1);
            printf("Enter The Marks Of Subject 2 : ");
            scanf("%f", &Temp.Sub_2);
            printf("Enter The Marks Of Subject 3 : ");
            scanf("%f", &Temp.Sub_3);
            Temp.Percentage = (Temp.Sub_1 + Temp.Sub_2 +
Temp.Sub_3) / 3;
        }
        fwrite(&Temp, sizeof(Temp), 1, tptr);
```

```
    }
    fclose(fptr);
    fclose(tptr);
    remove("Data.txt");
    rename("temp.txt", "Data.txt");
    if (Found == false)
        printf("\nNo Such Roll No Found In Data Base\n");
    else
        printf("\nRow Successfully Updated!\n");
}

void Add_Bars()
{
    printf("-----------------------------------------------
----------------------\n");
}

void Menu()
{
    printf("___Operation___\n");
    printf("1.Insert Row\n");
    printf("2.Remove Row\n");
    printf("3.Update Row\n");
    printf("4.Display\n");
    printf("5.Exit\n\n");
    printf("Enter Your Choice : ");
}

int Options()
{
    int opt;
    fflush(stdin);
    scanf("%d", &opt);
    Add_Bars();
    switch (opt)
    {
    case 1:
        Insert_Row();
        break;
```

```
        case 2:
            Remove_Row();
            break;
        case 3:
            Update_Row();
            break;
        case 4:
            Display();
            break;
        case 5:
            return 0;
        default:
            printf("Incorrect Input!\nTry Again!\n");
            break;
        }
        Add_Bars();
        return 1;
    }

    int main()
    {
        system("cls");
        printf("___Vicky_Gupta_20BCS070___\n\n");
        while (1)
        {
            Menu();
            if (!Options())
                break;
        }
        printf("Exiting...");
        Add_Bars();
        return 0;
    }
```

# Output :-

```
___Vicky_Gupta_20BCS070___

___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 1
--------------------------------------------------
Insert Operation Is Selected...
Enter The Name : Vicky Gupta
Enter The Roll No : 70
Enter The Marks Of Subject 1 : 100
Enter The Marks Of Subject 2 : 80
Enter The Marks Of Subject 3 : 90

Record Inserted Successfully!
--------------------------------------------------
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 1
--------------------------------------------------
Insert Operation Is Selected...
Enter The Name : Ijlal Ahmed
Enter The Roll No : 60
Enter The Marks Of Subject 1 : 100
Enter The Marks Of Subject 2 : 90
Enter The Marks Of Subject 3 : 100

Record Inserted Successfully!
--------------------------------------------------
```

```
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 1
-----------------------------------------------------------------------
Insert Operation Is Selected...
Enter The Name : Mohd Haider
Enter The Roll No : 45
Enter The Marks Of Subject 1 : 90
Enter The Marks Of Subject 2 : 100
Enter The Marks Of Subject 3 : 96

Record Inserted Successfully!
-----------------------------------------------------------------------
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 4
-----------------------------------------------------------------------
Display...
| Name | Roll No | Subject 1 | Subject 2 | Subject 3 | Percentage |

Vicky Gupta     70      100.00  80.00   90.00   90.00
Ijlal Ahmed     60      100.00  90.00   100.00  96.67
Mohd Haider     45      90.00   100.00  96.00   95.33
-----------------------------------------------------------------------
```

```
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 2
--------------------------------------------------------------------------------
Remove Operation Is Selected...
Enter The Roll No Of Student : 45

Row Successfully Removed!
--------------------------------------------------------------------------------
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 4
--------------------------------------------------------------------------------
Display...
| Name | Roll No | Subject 1 | Subject 2 | Subject 3 | Percentage |

Vicky Gupta     70      100.00  80.00   90.00   90.00
Ijlal Ahmed     60      100.00  90.00   100.00  96.67
--------------------------------------------------------------------------------
```

```
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 3
--------------------------------------------------------------------------------
Update Operation Is Selected...
Enter The Roll No Of Student : 70
Enter The Name : Vicky Gupta
Enter The Roll No : 70
Enter The Marks Of Subject 1 : 96
Enter The Marks Of Subject 2 : 88
Enter The Marks Of Subject 3 : 84

Row Successfully Updated!
--------------------------------------------------------------------------------
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 4
--------------------------------------------------------------------------------
Display...
| Name | Roll No | Subject 1 | Subject 2 | Subject 3 | Percentage |

Vicky Gupta     70      96.00   88.00   84.00   89.33
Ijlal Ahmed     60      100.00  90.00   100.00  96.67
--------------------------------------------------------------------------------
___Operation___
1.Insert Row
2.Remove Row
3.Update Row
4.Display
5.Exit

Enter Your Choice : 5
--------------------------------------------------------------------------------
Exiting...--------------------------------------------------------------------
```

# Computer Programming Lab

# CEN-392

# Program 12

## Code :-

```
#include <iostream>
using namespace std;

int lenght(char complex[])
{
    int len = 0;
    while (complex[len] != '\0')
        len++;
    return len;
}

void reverse(char str[])
{
    int len = lenght(str);
    for (int i = 0; i < len / 2; i++)
    {
```

```
            char ch = str[i];
            str[i] = str[len - 1 - i];
            str[len - 1 - i] = ch;
        }
    }

    bool check_decimal(char str[])
    {
        int itr = 0;
        while (str[itr] != '\0')
        {
            if (str[itr++] == '.')
                return true;
        }
        return false;
    }

    void String_Integer(char str[], int s, int n, float arr[],
    int indx)
    {
        int ten_pow = 1;
        arr[indx] = 0;
        while (n >= s)
        {
            arr[indx] += ten_pow * (str[n--] - '0');
            ten_pow *= 10;
        }
    }

    void String_Decimal(char str[], int s, int n, float arr[],
    int indx)
    {
        float ten_pow = 0.1;
        while (s <= n)
        {
            arr[indx] += ten_pow * (str[s++] - '0');
            ten_pow /= 10;
        }
    }
```

```
void Addition(float real[], float imaginary[])
{
    cout << "Addition Operation Is Selected..."
         << "\n";

    float r = real[0] + real[1], img = imaginary[0] +
imaginary[1];
    cout << "Addition : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Subtract(float real[], float imaginary[])
{
    cout << "Subtraction Operation Is Selected..."
         << "\n";
    float r = real[0] - real[1], img = imaginary[0] -
imaginary[1];
    cout << "Subtraction : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Multiply(float real[], float imaginary[])
{
    cout << "Multiplication Operation Is Selected..."
         << "\n";
    float r = real[0] * real[1] - imaginary[0] *
imaginary[1];
    float img = real[0] * imaginary[1] + imaginary[0] *
real[1];
    cout << "Multiplication : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}
```

```
void Division(float real[], float imaginary[])
{
    cout << "Devision Operation Is Selected..."
         << "\n";
    float devide = real[1] * real[1] + imaginary[1] *
imaginary[1];
    float r = real[0] * real[1] + imaginary[0] *
imaginary[1];
    float img = real[1] * imaginary[0] - imaginary[1] *
real[0];
    r /= devide;
    img /= devide;
    cout << "Division : |" << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Menu()
{
    cout << "\n____String_Operations____\n";
    cout << "1.Add\n";
    cout << "2.Subtract\n";
    cout << "3.Multiply\n";
    cout << "4.Devide\n";
    cout << "5.Exit\n";
    cout << "Enter Your Choice : ";
}

void AnsBar()
{
    cout <<
"
_____\n";
}

bool Options(float real[], float imaginary[])
{
```

```
    int opt;
    fflush(stdin);
    cin >> opt;
    AnsBar();
    switch (opt)
    {
    case 1:
        Addition(real, imaginary);
        break;
    case 2:
        Subtract(real, imaginary);
        break;
    case 3:
        Multiply(real, imaginary);
        break;
    case 4:
        Division(real, imaginary);
        break;
    case 5:
        return 0;
    default:
        cout << "Invalid Input!\nTry Again!\n";
    }
    AnsBar();
    return 1;
}

int main()
{
    system("cls");
    cout << "_____Vicky_Gupta_20BCS070_____\n\n";
    char complex[100];
    cout << "Enter The Complex Number : \n";
    gets(complex);
    float real[2], imaginary[2];
    int itr = 0;
    int clen = lenght(complex);
    int complex_counter = 0;
```

```
    int prev_indx = 0;
    while (clen > itr)
    {
        int citr = itr; // complex iterator
        while (complex[citr] != ',' && complex[citr] !=
'\0')
            citr++;
        itr = citr + 1;
        citr--;
        char r[50], img[50];
        int iitr = 0, ritr = 0; // imaginary iterator real
iterator
        if (complex[citr] == 'i')
        {
            citr--;
            while (citr >= prev_indx && complex[citr] != '+'
&& complex[citr] != '-')
                img[iitr++] = complex[citr--];
            if (citr >= prev_indx)
            {
                img[iitr++] = complex[citr--];
            }
            img[iitr] = '\0';
            reverse(img);
        }
        else
            img[0] = '\0';
        if (citr > prev_indx)
        {
            while (citr >= prev_indx && complex[citr] != '+'
&& complex[citr] != '-')
                r[ritr++] = complex[citr--];
            if (citr >= prev_indx)
            {
                r[ritr++] = complex[citr--];
            }
            r[ritr] = '\0';
            reverse(r);
        }
```

```
        else
            r[0] = '\0';

        if (r[0] != '\0') // for real
        {
            bool isDecimal = check_decimal(r);
            if (isDecimal)
            {
                int decimal_index = 0;
                while (r[decimal_index] != '.')
                    decimal_index++;
                if (r[0] == '+' || r[0] == '-')
                    String_Integer(r, 1, decimal_index - 1,
real, complex_counter);
                else
                    String_Integer(r, 0, decimal_index - 1,
real, complex_counter);
                String_Decimal(r, decimal_index + 1,
lenght(r) - 1, real, complex_counter);
            }
            else
            {
                if (r[0] == '+' || r[0] == '-')
                    String_Integer(r, 1, lenght(r) - 1,
real, complex_counter);
                else
                    String_Integer(r, 0, lenght(r) - 1,
real, complex_counter);
            }
            if (r[0] == '-')
                real[complex_counter] = -
real[complex_counter];
        }
        else
            real[complex_counter] = 0;
        if (img[0] != '\0') // for imaginary
        {
            bool isDecimal = check_decimal(img);
            if (isDecimal)
```

```
            {
                int decimal_index = 0;
                while (img[decimal_index] != '.')
                    decimal_index++;
                if (img[0] == '+' || img[0] == '-')
                    String_Integer(img, 1, decimal_index -
1, imaginary, complex_counter);
                else
                    String_Integer(img, 0, decimal_index -
1, imaginary, complex_counter);
                String_Decimal(img, decimal_index + 1,
lenght(img) - 1, imaginary, complex_counter);
            }
            else
            {
                if (img[0] == '+' || img[0] == '-')
                    String_Integer(img, 1, lenght(img) - 1,
imaginary, complex_counter);
                else
                    String_Integer(img, 0, lenght(img) - 1,
imaginary, complex_counter);
            }
            if (img[0] == '-')
                imaginary[complex_counter] = -
imaginary[complex_counter];
        }
        else
            imaginary[complex_counter] = 0;
        complex_counter++;
        prev_indx = itr;
    }
    cout << "\nComplex Number \n";
    for (int i = 0; i < 2; i++)
    {
        cout << i + 1 << ". " << real[i] << " " <<
imaginary[i] << "i\n";
    }
    cout << "\n";
    while (true)
```

```
    {
        Menu();
        if (!Options(real, imaginary))
            break;
    }
    cout << "Exiting...\n";
    AnsBar();
    return 0;
}
```

# Output :-

```
_____Vicky_Gupta_20BCS070_____

Enter The Complex Number :
-12.56+8.6i,24.6-9.8i

Complex Number
1. -12.56 8.6i
2. 24.6 -9.8i


____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 1
_____________________________________________________
Addition Operation Is Selected...
Addition : | 12.04-1.2i |
_____________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 2
_____________________________________________________
Subtraction Operation Is Selected...
Subtraction : | -37.16 + 18.4i |
_____________________________________________________
```

```
____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 3
_____________________________________________________
Multiplication Operation Is Selected...
Multiplication : | -224.696 + 334.648i |
_____________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 4
_____________________________________________________
Devision Operation Is Selected...
Division : |-0.560833 + 0.126172i |
_____________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 5
_____________________________________________________
Exiting...
_____________________________________________________
```