



JAMIA MILLIA ISLAMIA, NEW DELHI

C PROGRAMMING LAB

NAME: FAIZAN CHOUDHARY

ROLL NO: 20BCS021

SUBJECT CODE: CEN 392

SEMESTER: 3rd

COURSE: B.TECH. (COMPUTER ENGG.)

DEPT: DEPT OF COMPUTER ENGG.

PROGRAM NO	DATE	PROGRAM	REMARKS
1	13/09/2021	Operations on two matrices	
2	20/09/2021	Array operations	
3	27/09/2021	Number systems	
4	04/10/2021	Student data	
5	11/10/2021	Helical order of 2D matrix	
6	25/10/2021	String operations	
7	01/11/2021	Text information	
8	08/11/2021	Date difference	
9	15/11/2021	Find and replace utility in strings	
10	22/11/2021	Largest subarray with minimum sum	
11	29/11/2021	Data file handling	
12	13/12/2021	Operations on complex numbers (using string concepts)	

CODE:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
void display(int a[100][100], int r, int c)           //to display resultant array
```

```
{
    printf("\nResultant array is:\n");
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<c; j++)
        {
            printf("%d", a[i][j]);
            printf(" ");
        }
        printf("\n");
    }
}
```

```
void add (int a[100][100],int r1, int c1, int b[100][100], int r2, int c2)    //addition function
```

```
{
    int c[100][100];
    if (r1==r2 && c1==c2)                //checking for condition
    {
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                c[i][j]=a[i][j]+b[i][j];
            }
        }
        display (c, r1, c2);
    }
    else
```

```

{
    printf("\nThe matrices cannot be added!");
}
}

```

```

void subtract (int a[100][100],int r1, int c1, int b[100][100], int r2, int c2)           //subtraction function

```

```

{
    int c[100][100];                               //checking for condition
    if (r1==r2 && c1==c2)
    {
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                c[i][j]=a[i][j]-b[i][j];
            }
        }
        display (c, r1, c2);
    }
    else
    {
        printf("\nThe matrices cannot be subtracted!");
    }
}

```

```

void multiply (int a[100][100],int r1, int c1, int b[100][100], int r2, int c2)       //multiplication function

```

```

{
    int c[100][100];
    if (c1==r2)                                   //checking for condition
    {
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c2; j++)
            {

```

```

        c[i][j]=0;
        for (int k=0; k<r2; k++)
            c[i][j]+=a[i][k]*b[k][j];
    }
}
display (c, r1, c2);
}
else
{
    printf("The matrices cannot be multiplied! ");
}
}

```

```

int main()
{
    while (1)
    {
        int r1,c1,r2,c2, a[100][100], b[100][100];
        printf("\nEnter the no. of rows and columns for matrix 1: ");
        scanf("%d%d", &r1, &c1);
        printf("Enter the no. of rows and columns for matrix 2: ");
        scanf("%d%d", &r2, &c2);
        printf("Enter the matrix 1(row wise):\n");
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                scanf("%d", &a[i][j]);
            }
        }
        printf("Enter the matrix 2(row wise):\n");
        for (int i=0; i<r2; i++)
        {
            for (int j=0; j<c2; j++)

```

```

{
    scanf("%d", &b[i][j]);
}
}

```

A:

```

printf("\nMENU: \n1. Addition of two given matrices. \n2. Subtraction of two given matrices. \n3. Multiplication
of two given matrices. \n4. Exit.");

```

```

printf("\nEnter your choice: ");

```

```

int ch;

```

```

scanf("%d", &ch);

```

```

switch (ch)

```

```

{
    case 1: printf("Matrix 1: \n");
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                printf("%d", a[i][j]);
                printf(" ");
            }
            printf("\n");
        }
        printf("\nMatrix 2: \n");
        for (int i=0; i<r2; i++)
        {
            for (int j=0; j<c2; j++)
            {
                printf("%d", b[i][j]);
                printf(" ");
            }
            printf("\n");
        }
        add (a, r1, c1, b, r2, c2);

```

```

        break;
case 2: printf("Matrix 1: \n");
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                printf("%d", a[i][j]);
                printf(" ");
            }
            printf("\n");
        }
        printf("\nMatrix 2: \n");
        for (int i=0; i<r2; i++)
        {
            for (int j=0; j<c2; j++)
            {
                printf("%d", b[i][j]);
                printf(" ");
            }
            printf("\n");
        }
        subtract (a, r1, c1, b, r2, c2);
        break;
case 3: printf("Matrix 1: \n");
        for (int i=0; i<r1; i++)
        {
            for (int j=0; j<c1; j++)
            {
                printf("%d", a[i][j]);
                printf(" ");
            }
            printf("\n");
        }
        printf("\nMatrix 2: \n");

```

```
    for (int i=0; i<r2; i++)
    {
        for (int j=0; j<c2; j++)
        {
            printf("%d", b[i][j]);
            printf(" ");
        }
        printf("\n");
    }
    multiply (a, r1, c1, b, r2, c2);
    break;
case 4: exit(0);
default: printf("Wrong choice entered! Try again! ");
        goto A;
    }
}
return 0;
}
```


OUTPUT:

```
Enter the matrix 1(row wise):
1
2
3
4
Enter the matrix 2(row wise):
5
6
7
8

MENU:
1. Addition of two given matrices.
2. Subtraction of two given matrices.
3. Multiplication of two given matrices.
4. Exit.
Enter your choice: 1
Matrix 1:
1  2
3  4

Matrix 2:
5  6
7  8

Resultant array is:
6  8
10 12
```

```
Enter the no. of rows and columns for matrix 1: 3
3
Enter the no. of rows and columns for matrix 2: 3
3
Enter the matrix 1(row wise):
1
1
1
1
1
1
1
1
1
1
1
Enter the matrix 2(row wise):
1
1
1
1
1
1
1
1
1
1
1
```

```
MENU:
1. Addition of two given matrices.
2. Subtraction of two given matrices.
3. Multiplication of two given matrices.
4. Exit.
```

```
Enter your choice: 2
```

```
Matrix 1:
```

```
1 1 1
1 1 1
1 1 1
```

```
Matrix 2:
```

```
1 1 1
1 1 1
1 1 1
```

```
Resultant array is:
```

```
0 0 0
0 0 0
0 0 0
```

```
Enter the no. of rows and columns for matrix 1: 3
2
Enter the no. of rows and columns for matrix 2: 2
3
Enter the matrix 1(row wise):
1
1
2
2
3
3
Enter the matrix 2(row wise):
1
1
1
2
2
2

MENU:
1. Addition of two given matrices.
2. Subtraction of two given matrices.
3. Multiplication of two given matrices.
4. Exit.
Enter your choice: 3
Matrix 1:
1  1
2  2
3  3

Matrix 2:
1  1  1
2  2  2

Resultant array is:
3  3  3
6  6  6
9  9  9
```

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

20th September 2021

CODE:

```
#include <iostream>

using namespace std;

int N;          //Global variable for max size


//FUNCTION DEFINITIONS

void display (int a[], int n)
{
    cout<<"\nArray elements:\n ";
    for (int i=0; i<n; i++)
        cout<<a[i]<<" ";
    cout<<endl;
}


int insertion_beg (int arr[], int ele, int size)
{
    if(size == N){
        cout<<"\nCannot Insert: OverFlow !\n";
        return size;
    }

    for(int i = size-1; i>=0; i--)
    {
        arr[i+1] = arr[i];
```

```
}  
arr[0] = ele;  
size++;  
return size;  
}
```

```
int insertion_end (int arr[], int ele, int size)  
{  
    if(size == N){  
        cout<<"\nCannot Insert: OverFlow !\n";  
        return size;  
    }
```

```
arr[size] = ele;  
size++;  
return size;  
}
```

```
int insertion_pos(int arr[], int ele, int pos, int size)  
{  
    if(size == N){  
        cout<<"\nCannot Insert: Overflow !\n";  
        return size;  
    }
```

```
    if(pos<0){  
        cout<<"\nPlease enter a valid position\n";  
        return size;  
    }
```

```
    else if(pos>size-1){  
        cout<<"\nCannot insert outside the bounds. Please enter a valid position\n";
```

```
    return size;
}
```

```
for(int i = size-1; i>=pos; i--){
    arr[i+1] = arr[i];
}
arr[pos] = ele;
size++;
```

```
    return size;
}
```

```
int deletion_beg(int arr[], int size)
{
    if(size==0){
        cout<<"\nEmpty array!\n";
        return size;
    }
    for(int i = 0; i<size-1; i++){
        arr[i] = arr[i+1];
    }
    size--;
    return size;
}
```

```
int deletion_end(int arr[], int size)
{
    if(size==0){
        cout<<"\nUnderFlow! Empty array!\n";
        return size;
    }
}
```

```
size--;  
return size;  
}
```

```
int deletion_pos(int arr[], int pos, int size)
```

```
{  
    if(size==0){  
        cout<<"\nUnderFlow! Empty array!\n";  
        return size;  
    }
```

```
    if(pos<0)  
    {  
        cout<<"\nPlease enter a valid position to be deleted\n";  
        return size;  
    }
```

```
    else if(pos>size-1){  
        cout<<"\nPlease enter a valid position to be deleted\n";  
        return size;  
    }
```

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

```
int main()  
{  
    int ch, ele, index;
```

```

int size=0;

cout<<"FAIZAN CHOUDHARY\n20BCS021\n";

cout<<"\nEnter the maximum size of the array: ";

cin>>N;

int a[N];


while (1)
{
    A:
    cout<<"\n\nMENU:\n1. Insert element at beginning";
    cout<<"\n2. Insert element at end";
    cout<<"\n3. Insert element at a given index";
    cout<<"\n4. Delete element at beginning";
    cout<<"\n5. Delete element at end";
    cout<<"\n6. Delete element at a given index";
    cout<<"\n7. Exit";
    cout<<"\nEnter your choice: ";
    cin>>ch;
    switch (ch)
    {
        case 1: cout<<"\nEnter element to be inserted at beginning: ";
            cin>>ele;
            size= insertion_beg(a,ele,size);
            display (a,size);
            break;
        case 2: cout<<"\nEnter element to be inserted at end: ";
            cin>>ele;
            size= insertion_end (a,ele,size);
            display (a,size);
            break;
    }
}

```



```

case 3: cout<<"\nEnter element to be inserted at a given index: ";
        cin>>ele;

        cout<<"\nEnter index at which insertion to be carried out: ";
        cin>>index;

        size= insertion_pos (a,ele,index,size);

        display (a,size);

        break;

case 4: cout<<"\nDeleting at the beginning...\n ";
        size= deletion_beg (a,size);

        display (a,size);

        break;

case 5: cout<<"\nDeleting at end...\n ";
        size= deletion_end (a,size);

        display (a,size);

        break;

case 6: cout<<"\nEnter index at which deletion to be carried out: ";
        cin>>index;

        size= deletion_pos (a,index,size);

        display (a,size);

        break;

case 7: exit(0);

default: cout<<"Wrong choice entered! Try again! ";

        goto A;

    }

}

return 0;

}

```

OUTPUT:

```
FAIZAN CHOUDHARY  
20BCS021
```

```
Enter the maximum size of the array: 10
```

```
MENU:
```

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

```
Enter your choice: 1
```

```
Enter element to be inserted at beginning: 1
```

```
Array elements:
```

```
1
```

```
MENU:
```

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

```
Enter your choice: 2
```

```
Enter element to be inserted at end: 2
```

```
Array elements:
```

```
1 2
```

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 3

Enter element to be inserted at a given index: 0

Enter index at which insertion to be carried out: 0

Array elements:

0 1 2

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 4

Deleting at the beginning...

Array elements:

1 2

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 5

Deleting at end...

Array elements:

1

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 2

Enter element to be inserted at end: 2

Array elements:

1 2

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 2

Enter element to be inserted at end: 3

Array elements:

1 2 3

MENU:

1. Insert element at beginning
2. Insert element at end
3. Insert element at a given index
4. Delete element at beginning
5. Delete element at end
6. Delete element at a given index
7. Exit

Enter your choice: 6

Enter index at which deletion to be carried out: 1

Array elements:

1 3

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

27th September 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>
char num[200]; //to store the converted hexa number

int length (char *a)
{
    int len=0;
    for (int i=0; a[i]!='\0'; i++)
        len++;
    return len;
}

void decimal_to_hex (int n)
{
    int i=0;
    while (n!=0)
    {
        int temp=n%16;
        if (temp<10)                //to check for temp being a digit
        {
            num[i] = temp + 48;    //using ASCII values (base value at 48 is 0)
            i++;
        }
        else
        {
            num[i] = temp + 55;    //55+10= 65 is the base value of A in ASCII
            i++;
        }
        n/=16;                    //dividing number each time by 16 until it becomes 0
    }
}

int hex_to_decimal (char *n)
{
    int base=1;                  //16^0
    int val=0;
    for (int i=length(n); i>=0; i--)    //reading string in reverse order
    {
        if (n[i]>='0' && n[i]<='9')
        {
```

```

        val+= ((int)n[i]-48) * base;    //using ASCII values (base value at 48 is 0)
        base*=16;
    }
    else if (n[i]>='A' && n[i] <= 'F')
    {
        val+= ((int)n[i]-55) * base;    //char-
55 gives a digit since alphabets start from 65
        base*=16;
    }
    else if (n[i]>='a' && n[i] <= 'f')
    {

    }
}
return val;
}

int main()
{
    int ch,n, dec;
    char hex[200];
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n");
    while (1)
    {
        A:
        printf("\n\nCONVERSION MENU\n1. Decimal to Hexadecimal\n2. Hexadecimal to Decimal\n3. Exit\n");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: printf("Enter the decimal number: ");
                    scanf("%d", &n);
                    decimal_to_hex(n);
                    printf("\nThe number after the conversion is: ");
                    for (int j=length(num); j>=0; j--)    //to print in reverse
                        printf("%c" , num[j]);
                    break;

            case 2: printf("Enter the hexadecimal number (in standard format): ");
                    scanf("%s", &hex);
                    dec = hex_to_decimal(hex);
                    printf("\nThe number after the conversion is: %d", dec, "\n");
                    break;

            case 3: exit(0);
            default: printf("\nWrong choice! Enter again...\n");
                    goto A;
        }
    }
    return 0;
}

```

OUTPUT:

```
FAIZAN CHOUDHARY  
20BCS021
```

```
CONVERSION MENU
```

1. Decimal to Hexadecimal
2. Hexadecimal to Decimal
3. Exit

```
1
```

```
Enter the decimal number: 10
```

```
The number after the conversion is: A
```

```
CONVERSION MENU
```

1. Decimal to Hexadecimal
2. Hexadecimal to Decimal
3. Exit

```
2
```

```
Enter the hexadecimal number (in standard format): FF
```

```
The number after the conversion is: 255
```

```
CONVERSION MENU
```

1. Decimal to Hexadecimal
2. Hexadecimal to Decimal
3. Exit

```
3
```


FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

4th October 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>
int a[25][5];

//Assuming that in a subject only one student has highest marks

int main()
{
    int r,ch;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");
    printf("Enter the number of students(max 25): ");
    scanf("%d", &r);
    for (int i=0; i<r; i++)
    {
        printf("Enter the roll number of student %d: ", i+1);
        scanf("%d", &a[i][0]);
        printf("Enter the age of student %d: ", i+1);
        scanf("%d", &a[i][1]);
        printf("Enter the marks in 3 subjects of student %d: ", i+1);
        for (int j=2; j<5; j++)
            scanf("%d", &a[i][j]);
    }

    printf("\nThe entries are as follows:\n");
    printf("\n\nRoll Number\t\tAge\t\tSubject_1\t\tSubject_2\t\tSubject_3\n");
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<5; j++)
            printf("%d\t\t\t", a[i][j]);
        printf("\n");
    }

    while (1)
    {
        float pr[r], mx=0;
        int index, roll1, roll2, flag=0;
        int max[3]={0,0,0}, rollno[3];
        printf("\n\nMENU\n1. Percentage secured by each student along with their roll numbers.\n2. Highest marks in each subject along with roll number\n3. Student scoring the highest percentage.\n4. Exit.\n");
```

```

scanf("%d", &ch);
switch (ch)
{
    case 1: printf("\nRoll Number\t\tPercentage scored (%%)\n");
            float per;
            for (int i=0; i<r; i++)
            {
                per=0;
                printf("%d\t\t\t", a[i][0]);                //printing roll no
                for (int j=2; j<5; j++)                        //index starting from
subject marks only
                {
                    per+=a[i][j];
                }
                per/=3.0;
                printf("%.2f\n", per);
            }
            break;

    case 2: for (int i=0; i<r; i++)
            {
                for (int j=2; j<5; j++)
                {
                    if(a[i][j]>max[j-2])
                    {
                        max[j-2]=a[i][j];                //as according to the array given
                        rollno[j-2]=a[i][0];
                    }
                }
            }
            printf("\nSubject\t\tHighest Marks\t\tRoll no who secured highest mark
s\n");

            for (int j=0; j<3; j++)
            {
                printf("Subject_%d\t\t", j+1);
                printf("%d\t\t", max[j]);
                printf("%d\t\t\t", rollno[j]);
            }
            printf("\n");
            break;

    case 3:
            for (int i=0; i<r; i++)                        //finding percentage of each indiv
            dual and storing them in pr[]
            {
                pr[i]=0;
                for (int j=2; j<5; j++)
                {
                    pr[i]+=a[i][j];
                }
                pr[i]/=3.0;
                if (pr[i]>mx)                                //finding max percentage
                {

```

```

        mx=pr[i];
        index=i;
        roll1=a[i][0];
    }
}
for (int i=0; i<r; i++)                //traversal
{
    if (pr[i]==mx && i!=index)
    {
        if (a[i][1]<a[index][1])        //required age condition
        {
            roll1=a[i][0];
            index=i;
        }
        else if (a[i][1]==a[index][1])
        {
            roll2=a[i][0];
            flag=1;
        }
    }
}
printf("\nStudent(s) who has/have scored the highest percentage are:\n");

if (flag==0)
{
    printf("Roll No- %d", roll1);
    printf(" with a percentage of %.2f\n", mx);
}
else if (flag==1)
{
    printf("Roll No- %d", roll1);
    printf(" with a percentage of %.2f\n", mx);
    printf("Roll No- %d", roll2);
    printf(" with a percentage of %.2f\n", mx);
}
break;
case 4: exit(0);
}
}
return 0;
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

```
Enter the number of students(max 25): 3
Enter the roll number of student 1: 1
Enter the age of student 1: 19
Enter the marks in 3 subjects of student 1: 94
96
98
Enter the roll number of student 2: 2
Enter the age of student 2: 20
Enter the marks in 3 subjects of student 2: 89
60
48
Enter the roll number of student 3: 3
Enter the age of student 3: 19
Enter the marks in 3 subjects of student 3: 95
97
96
```

The entries are as follows:

Roll Number	Age	Subject_1	Subject_2	Subject_3
1	19	94	96	98
2	20	89	60	48
3	19	95	97	96

MENU

1. Percentage secured by each student along with their roll numbers.
 2. Highest marks in each subject along with roll number
 3. Student scoring the highest percentage.
 4. Exit.
- 1

Roll Number	Percentage scored (%)
1	96.00
2	65.67
3	96.00

MENU

1. Percentage secured by each student along with their roll numbers.
 2. Highest marks in each subject along with roll number
 3. Student scoring the highest percentage.
 4. Exit.
- 2

Subject	Highest Marks	Roll no who secured highest marks
Subject_1	95	3
Subject_2	97	3
Subject_3	98	1

MENU

1. Percentage secured by each student along with their roll numbers.
2. Highest marks in each subject along with roll number
3. Student scoring the highest percentage.
4. Exit.

3

Student(s) who has/have scored the highest percentage are:

Roll No- 1 with a percentage of 96.00

Roll No- 3 with a percentage of 96.00

MENU

1. Percentage secured by each student along with their roll numbers.
2. Highest marks in each subject along with roll number
3. Student scoring the highest percentage.
4. Exit.

4

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

11th October 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>

void display (int a[10][10], int r, int c)
{
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<c; j++)
            printf("\t%d", a[i][j]);
        printf("\n");
    }
}

void display_helical (int a[10][10], int r, int c)
{
    int i, r_index=0, c_index=0;           //c_index and r_index chosen to access indexes
    in spiral order
    while (r_index<r && c_index<c)         //to print the array in spiral form within
    bounds
    {
        for (i=c_index; i<c; i++)          //prints the first row of the 2d array in the
        first step and the reduced 2d array in subsequent steps
            printf("%d ", a[r_index][i]);
        r_index++;                         //to increment the row index so as to start
        with next row (2d array reduction)
        for (i=r_index; i<r; i++)          //prints the last column of the 2d array in
        the first step and the reduced 2d array in subsequent steps
            printf("%d ", a[i][c-1]);
        c--;                               //to decrement the no of columns so as to
        start with previous column (2d array reduction)
        if (r_index<r)                     //for printing the remaining row (last row in
        first iteration)
        {
            for (i=c-1; i>=c_index; i--)
                printf("%d ", a[r-1][i]);
            r--;                             //decrementing no of rows to reduce 2d array
        }
        if (c_index<c)                     //for printing the remaining column (first
        column in first iteration)
        {
```

```

        for (i=r-1; i>=r_index; i--)
            printf("%d ", a[i][c_index]);
        c_index++;
        //incrementing no of columns to reduce 2d
array
    }
}
printf("\n");
}

void display_helical_anti (int a[10][10], int r, int c)
{
    int i, r_index=0, c_index=0;
    while (r_index<r && c_index<c)
    {
        for (i=c-1; i>=c_index; i--)
            printf("%d ", a[r_index][i]);
        r_index++;
        for (i=r_index; i<r; i++)
            printf("%d ", a[i][0]);
        c_index++;
        if (r_index<r)
        {
            for (i=c_index; i<c; i++)
                printf("%d ", a[r-1][i]);
            r--;
        }
        if (c_index<c)
        {
            for (i=r-1; i>=r_index; i--)
                printf("%d ", a[i][c-1]);
            c--;
        }
    }
    printf("\n");
}

int main()
{
    int r, c, ch;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");
    A:
    printf("Enter the number of rows and columns for the 2d matrix (max 10 each):\n");
    scanf("%d%d", &r, &c);
    int a[10][10];
    printf("Enter the elements (row major):\n");
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<c; j++)
            scanf("%d", &a[i][j]);
    }
    printf("\nArray elements:\n");
    display(a,r,c);
    while (1)
    {

```

```

        B:
        printf("\n\nMENU\n1. Print in helical order.\n2. Print in reverse helical
order.\n3. Enter another array.\n4. Exit.\n");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: printf("\nArray elements printed in helical order:\n");
                    display_helical(a,r,c);
                    break;
            case 2: printf("\nArray elements printed in reverse helical order:\n");
                    display_helical_anti(a,r,c);
                    break;
            case 3: goto A;
            case 4: exit(0);
            default: printf("\nWrong choice! Enter again...\n");
                    goto B;
        }
    }
    return 0;
}

```

OUTPUT:

```

Enter the number of rows and columns for the 2d matrix (max 10 each):
3 4
Enter the elements (row major):
1 2 3 4
5 6 7 8
9 10 11 12

Array elements:
    1      2      3      4
    5      6      7      8
    9     10     11     12

```

```

MENU
1. Print in helical order.
2. Print in reverse helical order.
3. Enter another array.
4. Exit.
1

Array elements printed in helical order:
1 2 3 4 8 12 11 10 9 5 6 7

```

```

MENU
1. Print in helical order.
2. Print in reverse helical order.
3. Enter another array.
4. Exit.
2

Array elements printed in reverse helical order:
4 3 2 1 5 9 10 11 12 8 7 6

```


FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

25th October 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>

char string[100]; //temp string
// char *s=(char *)malloc(100*sizeof(char));
char s[100];      //main string input
char b[100];      //secondary input

int strlen (char *s)
{
    int count=0;
    for (int i=0; s[i]!='\0'; i++)
        count++;
    return count;
}

char * strrev (char *s)
{
    int k=strlen(s);
    for (int i=0; s[i]!='\0'; i++)
        string[i]=s[k-1-i];
    return string;
}

char * strcpy (char *s,char *b)
{
    // int k = strlen (b);
    // char *s=(char *)realloc(s, (k+1) * sizeof(char));
    char *temp=s;
    while (*b!='\0')
    {
        *s=*b;
        s++;
        b++;
    }
    *s='\0';
    return temp;
}

void strcmp (char *a, char *b)
```

```

{
    while (*a)
    {
        if (*a!=*b)
        {
            printf("\nThe two strings are not equal!\n");
            return;
        }
        a++;
        b++;
    }
    printf("\nThe two strings are equal!\n");
}

char *strcat (char *a, char *b)
{
    int k=strlen(a);
    int l=strlen(b);
    for (int i=0; i<k; i++)
        string[i]=a[i];
    for (int i=0; i<l; i++)
        string[k+i]=b[i];
    string[strlen(string)]='\0';
    return string;
}

void palindrome (char *s)
{
    int len=strlen(s), j, flag=1;
    for (int i=0; i<len-1, j<len/2; i++, j--)
    {
        if (s[i]!=s[j])
        {
            flag=0;
            break;
        }
    }
    if (!flag)
        printf("\nThe given string is not a palindrome.\n");
    else
        printf("\nThe given string is a palindrome.\n");
}

int substr(char *a, char *b)
{
    int k=strlen(a);    //substring
    int l=strlen(b);    //main string
    for (int i=0; i<=(l-k); i++)    //to traverse in larger string upto the length of
smaller string
    {
        int j;
        for (j=0; j<k; j++)
            if (b[i+j] != a[j])
                break;
    }
}

```

```

        if (j==k)
            return i;
    }
    return -1;
}

int main()
{
    int ch;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");
    A:
    printf("Enter the main string: ");
    scanf("%[^\n]", &s);
    while (1)
    {
        B:
        printf("\n\nMENU\n1. strlen\n2. strrev\n3. strcpy\n4. strcmp\n5. strcat\n6. Check
for palindrome.\n7. Search for substring.\n8. Enter new main string.\n9. Exit.\n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1: printf("\nLength of the string is: %d\n", strlen(s));
                    break;
            case 2: printf("\nString after reversing: %s\n", strrev(s));
                    break;
            case 3: printf("\nEnter the string to be copied: ");
                    getchar();
                    scanf("%[^\n]", &b);
                    printf("\nMain String: %s\nSecondary String: %s", s,b);
                    printf("\nAfter Copying, Main String: %s\n",strcpy(s,b));
                    break;
            case 4: printf("\nEnter the second string to be compared: ");
                    getchar();
                    scanf("%[^\n]", &b);
                    strcmp(s,b);
                    break;
            case 5: printf("\nEnter the second string to be concatenated: ");
                    getchar();
                    scanf("%[^\n]", &b);
                    printf("\nMain String: %s\nSecondary String: %s", s,b);
                    printf("\nAfter concatenation: %s",strcat(s,b));
                    break;
            case 6: palindrome(s);
                    break;
            case 7: printf("\nEnter the substring to be checked for: ");
                    getchar();
                    scanf("%[^\n]", &b);
                    printf("\nMain String: %s\nSubstring: %s", s,b);
                    int c=substr(b,s);
                    if (c==-1)
                        printf("\nSubstring is not present!\n");
                    else
                        printf("\nSubstring is present at position: %d\n", c);
                    break;

```

```
        case 8: goto A;
        case 9: exit(0);
        default: printf("\nWrong choice! Enter again...\n");
                goto B;
    }
}
return 0;
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021
```

```
Enter the main string: Faizan
```

```
MENU
```

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- ```
1
```

```
Length of the string is: 6
```

```
MENU
```

1. strlen
  2. strrev
  3. strcpy
  4. strcmp
  5. strcat
  6. Check for palindrome.
  7. Search for substring.
  8. Enter new main string.
  9. Exit.
- ```
2
```

```
String after reversing: naziaF
```

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 3

Enter the string to be copied: Jamia Millia Islamia

Main String: Faizan

Secondary String: Jamia Millia Islamia

After Copying, Main String: Jamia Millia Islamia

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 4

Enter the second string to be compared: Jamia Millia Islamia

The two strings are equal!

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 5

Enter the second string to be concatenated: , New Delhi

Main String: Jamia Millia Islamia

Secondary String: , New Delhi

After concatenation: Jamia Millia Islamia, New Delhi

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 3

Enter the string to be copied: racecar

Main String: Jamia Millia Islamia

Secondary String: racecar

After Copying, Main String: racecar

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 6

The given string is a palindrome.

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 7

Enter the substring to be checked for: car

Main String: racecar

Substring: car

Substring is present at position: 4

MENU

1. strlen
 2. strrev
 3. strcpy
 4. strcmp
 5. strcat
 6. Check for palindrome.
 7. Search for substring.
 8. Enter new main string.
 9. Exit.
- 9

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

1st November 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char ch = 'y';
    char text[400];
    int i=0, j, c=0;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");

    do
    {
        printf("\nEnter the piece of text (terminating with $): \n");
        scanf("%[^$]", &text);

        long no_of_spaces = 0, no_of_tabs =0, no_of_sentences = 0, no_of_vowels = 0;
        // long no_of_lines = 0;

        for (i=0; text[i]!='\0'; i++)
        {
            if (text[i]== ' ')
                no_of_spaces++;
            if (text[i]== '\t')
                no_of_tabs++;
            if (text[i]== '.' || text[i]== '!' || text[i]== '?')
                no_of_sentences++;
            // if (text[i] == '\n');
            // no_of_lines++;
            if (text[i]=='A' || text[i]=='a' || text[i]=='E' || text[i]=='e' ||
text[i]=='I' || text[i]=='i' || text[i]=='O' || text[i]=='o' || text[i]=='U' ||
text[i]=='u')
                no_of_vowels++;

        }

        // to format the extra spaces to one space
        i=0, j=0;
        char second[400];
        while (text[i] != '\0')
        {
```

```

        if (text[i] == ' ')
        {
            int temp = i+1;
            if (text[temp] != '\0')
            {
                while (text[temp] == ' ' && text[temp] != '\0')
                {
                    if (text[temp] == ' ')
                        i++;
                    temp++;
                }
            }
            second[j] = text[i];
            i++;
            j++;
        }
        second[j] = '\0';

printf("\nNumber of spaces : %d\n", no_of_spaces);
printf("Number of tabs: %d\n", no_of_tabs);
printf("Number of sentences: %d\n", no_of_sentences);
// printf("Number of lines: %d\n", no_of_lines);
printf("Number of vowels: %d\n", no_of_vowels);

printf("\nFormatted piece of text:\n");
for (i=0; second[i]!='\0'; i++)
    printf("%c", second[i]);
printf("\n");

A:
printf("\nDo you want to enter more? (Y/N): ");
getchar();
getchar();
scanf("%c", &ch);
} while (ch == 'y' || ch == 'Y');

return 0;
}

```

OUTPUT:

```
FAIZAN CHOUDHARY  
20BCS021
```

```
Enter the piece of text (terminating with $):
```

```
Hello I am Faizan Choudhary, a student of Jamia Millia Islamia,      New Delhi.  
I live in Mumbai and I love      reading books.  
How are you?  
I hope you are doing great!$
```

```
Number of spaces : 30
```

```
Number of tabs: 2
```

```
Number of sentences: 4
```

```
Number of vowels: 59
```

```
Formatted piece of text:
```

```
Hello I am Faizan Choudhary, a student of Jamia Millia Islamia,      New Delhi.  
I live in Mumbai and I love reading books.  
How are you?  
I hope you are doing great!
```

```
Do you want to enter more? (Y/N): █
```


FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

8th November 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>

// 08-11-2021,09-05-1972      18080 days
// 06-01-2016,28-02-2020      1514 days
// 12-01-1880,14-10-2021      51775 days
// 08-01-2017,08-01-2021      1461 days

int isLeap(int y)
{
    if ((y%4==0 && y%100!=0) || (y%400==0))
        return 1;
    return 0;
}

long leapYear(int y1, int y2)
{
    long j, s=0;
    for (j=y1; j<y2; j++)
    {
        s+=365;
        if (isLeap(j)==1)
            // for 366 days
            s++;
    }
    return s;
}

long daysofyear (int d, int m, int y)
{
    long res, day[13];
    // initialising each month with the sum of its previous months
    day[1] = 0; day[2] = 31; day[3] = 59; day[4] = 90;
    day[5] = 120; day[6] = 151; day[7] = 181; day[8] = 212;
    day[9] = 243; day[10]= 273; day[11]= 304; day[12]= 334;
    res=day[m]+d;
    // since if the month is Jan or Feb, we wont include the leap day
    if(isLeap(y)==1 && (m>2))
        res++;
    return res;
}
```

```

long daysdiff (int d1, int m1, int y1, int d2, int m2, int y2)
{
    long days;
    // finding out difference in dates
    days = daysofyear(d2, m2, y2) - daysofyear(d1, m1, y1);
    if (y1!=y2)
    {
        if (y1<y2)
            days+=leapYear(y1,y2);
        else
            days-=leapYear(y2,y1);
    }
    return days>0 ? days : -days;
}

int valid(int d, int m, int y)
{
    int is_valid = 1, is_leap = 0;

    if (y >= 1800 && y <= 2021)
    {
        if ((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))
        {
            is_leap = 1;
        }

        if (m >= 1 && m <= 12)
        {
            if (m == 2)
            {
                if (is_leap && d == 29)
                {
                    is_valid = 1;
                }
                else if (d > 28)
                {
                    is_valid = 0;
                }
            }

            else if (m == 4 || m == 6 || m == 9 || m == 11)
            {
                if (d > 30)
                {
                    is_valid = 0;
                }
            }

            else if (d > 31)

```

```

        {
            is_valid = 0;
        }
    }

    else
    {
        is_valid = 0;
    }
}
else
{
    is_valid = 0;
}

return is_valid;
}

int main()
{
    char date[25], date1[11], date2[11], ch='y';
    int i;
    int d1=0, m1=0, y1=0, d2=0, m2=0, y2=0;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");

    do
    {
        A:
        printf("\nEnter the string containing two dates (separated by comma): ");
        scanf("%s", &date);

        for (i=0; date[i]!=','; i++)
        {
            date1[i] = date[i];
            date2[i] = date[i+11];
        }

        d1 = (date1[0]-'0')*10 + date1[1]-'0';
        d2 = (date2[0]-'0')*10 + date2[1]-'0';

        m1 = (date1[3]-'0')*10 + date1[4]-'0';
        m2 = (date2[3]-'0')*10 + date2[4]-'0';

        y1 = (date1[6]-'0')*1000 + (date1[7]-'0')*100 + (date1[8]-'0')*10 + date1[9]-'0';
        y2 = (date2[6]-'0')*1000 + (date2[7]-'0')*100 + (date2[8]-'0')*10 + date2[9]-'0';

        if (!valid(d1,m1,y1))
        {
            printf("\nFirst date is invalid.\n");
            goto A;
        }

        if (!valid(d2,m2,y2))
        {

```

```

        printf("\nSecond date is invalid.\n");
        goto A;
    }

    // printf("\n%d %d %d\n", d1, m1, y1);
    // printf("\n%d %d %d\n", d2, m2, y2);
    long res = daysdiff(d1, m1, y1, d2, m2, y2);
    printf("\nNo of Days: %ld", res);

    printf("\nDo you want to enter more? (y/n) ");
    getchar();
    scanf("%c", &ch);
} while(ch=='y' || ch=='Y');
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

Enter the string containing two dates (separated by comma): 84-21-2110,19-11-2021

First date is invalid.

Enter the string containing two dates (separated by comma): 18-01-2021,09-13-1990

Second date is invalid.

Enter the string containing two dates (separated by comma): 08-11-2021,09-05-1972

No of Days: 18080

Do you want to enter more? (y/n) y

Enter the string containing two dates (separated by comma): 06-01-2016,28-02-2020

No of Days: 1514

Do you want to enter more? (y/n) y

Enter the string containing two dates (separated by comma): 12-01-1880,14-10-2021

No of Days: 51775

Do you want to enter more? (y/n) y

Enter the string containing two dates (separated by comma): 08-01-2017,08-01-2021

No of Days: 1461

Do you want to enter more? (y/n) n

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

15th November 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char mainstr[100], str1[50], str2[50];
int ind, flag=0, j;

int index_of_substr(char mainstr[], char str1[])
{
    flag=0;
    for (int a=0; mainstr[a]!='\0'; a++)
    {
        for (int i=a; mainstr[i]!='\0'; i++)           //to take out smaller strings
        out of mainstr
        {
            if (mainstr[i]==str1[0])
            {
                ind=i;
                j=0;
                while (mainstr[i]==str1[j])           //to let the check go on until
                substr or mainstr ends, and ensuring that the substr is exactly in the mainstr
                {
                    i++;
                    j++;
                }
                if (j>=strlen(str1))                   //if the length in string to be
                checked is less than the checking condition
                {
                    flag=1;
                    break;
                }
            }
        }
        if (flag)
            return ind;
    }
    return -1;
}

void substr_before(char mainstr[], char strb[], int index)
```

```

{
    int c=index;
    int i=0;
    while (i!=c)
    {
        strb[i] = mainstr[i];
        i++;
    }
}

void substr_after(char mainstr[], char stra[], int index)
{
    int c=index;
    int i=0;

    for (i=index; mainstr[i]!=' '; i++)
        c++;
    i=0;
    while (mainstr[i+c]!='\0')
    {
        stra[i] = mainstr[i+c];
        i++;
    }
    // for last word
    if (stra[0]=='.')
        stra[0]=' ';
}

void replace (char replace[], char strb[], char str2[], char stra[])
{
    int k=strlen(strb);
    int l=strlen(str2);
    int m=strlen(stra);
    for (int i=0; i<k; i++)
        replace[i]=strb[i];
    for (int i=0; i<l; i++)
        replace[k+i]=str2[i];
    for (int i=0; i<m; i++)
        replace[k+l+i]=stra[i];
    replace[k+l+m]='\0';
    printf("%s\n", replace);
}

int main()
{
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");

    int ch, count=0;
    while(1)
    {
        A:
        count++;
        printf("Enter the main string (ending with .): ");
        if (count>1)

```

```

{
    getchar();
    scanf("%[^\\n]", &mainstr);
}
else
    scanf("%[^\\n]", &mainstr);
B:
// str1[50]={'\0'};
printf("Enter the substring to be found (case sensitive): ");
getchar();
scanf("%[^\\n]", &str1);

if (index_of_substr(mainstr, str1)==-1)
{
    printf("\nSubstring is not present in the main string!");
    C:
    printf("\nChoose: \n1. Start Over.\n2. Re-enter substring.\n3. Exit.\n");
    scanf("%d", &ch);
    switch (ch)
    {
        case 1: goto A;
                break;
        case 2: goto B;
                break;
        case 3: exit(0);
        default: printf("\nWrong choice! Enter again...\n");
                 goto C;
    }
}
else
{
    char stra[50]={'\0'};
    char strb[50]={'\0'};
    substr_before(mainstr, strb, ind);
    substr_after(mainstr, stra, ind);
    printf("\nSubstring is present in the main string at index %d!", ind);
    D:
    printf("\nEnter the string to replace: ");
    char str2[50]={'\0'};
    getchar();
    scanf("%[^\\n]", &str2);
    printf("\nAfter replacement: ");
    replace(mainstr, strb, str2, stra);
    printf("\nChoose: \n1. Start Over.\n2. Continue with same string. \n3.
Exit.\n");
    scanf("%d", &ch);
    switch (ch)
    {
        case 1: goto A;
                break;
        case 2: goto B;
        case 3: exit(0);
        default: printf("\nWrong choice! Enter again...\n");
                 goto D;
    }
}
}

```

```
    }  
    }  
}  
  
return 0;  
}
```

OUTPUT:

```
FAIZAN CHOUDHARY  
20BCS021
```

```
Enter the main string (ending with .): Jamia Hamdard is my University.  
Enter the substring to be found (case sensitive): Hamdard
```

```
Substring is present in the main string at index 6!  
Enter the string to replace: Millia Islamia
```

```
After replacement: Jamia Millia Islamia is my University.
```

```
Choose:
```

1. Start Over.
2. Continue with same string.
3. Exit.

```
2
```

```
Enter the substring to be found (case sensitive): my
```

```
Substring is present in the main string at index 24!  
Enter the string to replace: our
```

```
After replacement: Jamia Millia Islamia is our University.
```

```
Choose:
```

1. Start Over.
2. Continue with same string.
3. Exit.

```
3
```


FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

22nd November 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>

// #define min(x,y) (x<y ? x : y)

void min_sum_of_max_subarray (int arr[], int n)
{
    int sum = __INT16_MAX__;
    int s = 0, e = 0;
    for(int i = 0; i<n; i++)
    {
        for(int j = i; j<n; j++)
        {
            int pref_sum = 0;
            for(int k = i; k<=j; k++)
                pref_sum += arr[k];
            if(pref_sum < sum)
            {
                sum = pref_sum;
                s = i;
                e = j;
            }
            else if(pref_sum == sum && (j - i) > (e - s))
            {
                s = i;
                e = j;
            }
        }
    }

    printf("\nThe minimum sum of the largest subarray is: \n");
    for (int i=s; i<=e; i++)
        printf("%d ", arr[i]);
    printf("\n = %d\n", sum);
}

int main()
{
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");

    int n;
```

```

char ch;
do
{

    printf("\nEnter the size of the array: ");
    scanf("%d", &n);
    int arr[n];

    printf("\nEnter the array elements: ");
    for (int i=0; i<n; i++)
        scanf("%d", &arr[i]);

    min_sum_of_max_subarray(arr,n);

    printf("\nDo you want to enter again? (y/n): ");
    getchar();
    scanf("%c", &ch);
} while (ch == 'y' || ch == 'Y');
return 0;
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

Enter the size of the array: 7

Enter the array elements: 5 -3 1 -5 -1 7 -5

The minimum sum of the largest subarray is:
-3 1 -5 -1 = -8

Do you want to enter again? (y/n): y

Enter the size of the array: 6

Enter the array elements: -5 -3 3 -4 -2 -2

The minimum sum of the largest subarray is:
-5 -3 3 -4 -2 -2 = -13

Do you want to enter again? (y/n): y

Enter the size of the array: 5

Enter the array elements: 5 7 3 1 4

The minimum sum of the largest subarray is:
1 = 1

Do you want to enter again? (y/n): y

Enter the size of the array: 6

Enter the array elements: -5 -3 13 -4 -2 -2

The minimum sum of the largest subarray is:
-4 -2 -2 = -8

Do you want to enter again? (y/n): n

FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

29th November 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdlib.h>

struct Student
{
    char name[50];
    int roll;
    float sub[3];
    float per;
};

int check_roll (char *f_name, int r)
{
    struct Student t;
    int f=1;
    FILE *fp;
    fp = fopen (f_name, "r");
    if (fp == NULL)
    {
        printf("\nCannot open file!\n");
        return -1;
    }
    while (fread (&t, sizeof(struct Student), 1, fp))
    {
        if (t.roll == r)
        {
            printf("\nRoll Number already exists! Try again...\n");
            f=0;
        }
    }
    return f;
}

void insert_data (char *f_name)
{
    struct Student s;
    printf("\nEnter the name of the student: ");
    getchar();
    scanf("%[^\\n]", s.name);
    B:
```

```

printf("\nEnter the roll number of the student: ");
scanf("%d", &s.roll);
int k = check_roll(f_name, s.roll);
if (!k)
    goto B;
printf("\nEnter the marks of the student in three subjects: ");
for (int i=0; i<3; i++)
    scanf("%f", &s.sub[i]);
s.per = (s.sub[0] + s.sub[1] + s.sub[2]) / 3.0;

FILE *fp;
fp = fopen (f_name, "a");
if (fp == NULL)
{
    printf("\nCannot open file!\n");
    return;
}
fseek (fp, 0, SEEK_END);
fwrite (&s, sizeof(struct Student), 1, fp);
if(fwrite != 0)
    printf("\nInserted row successfully!\n");
else
    printf("Error writing file!\n");
fclose(fp);
}

void delete_data (char *f_name, int roll)
{
    FILE *fp, *fp_tmp;
    struct Student record;
    // flag for checking if record present or not
    int f=0;
    fp = fopen (f_name, "r");
    if (fp == NULL)
    {
        printf("\nCannot open file!\n");
        return;
    }
    // temp file to copy the rest of the records
    fp_tmp = fopen ("temp.txt", "w");
    if (fp_tmp == NULL)
    {
        printf("\nCannot open temporary file!\n");
        return;
    }
    while (fread (&record, sizeof(struct Student), 1, fp))
    {
        if (record.roll == roll)
        {
            printf("\nRecord with the given roll number found, and deleted successfully!\n");
            f=1;
        }
        else

```

```

        fwrite (&record, sizeof(struct Student), 1, fp_tmp);
    }
    if (!f)
    {
        printf("\nNo record found with the given roll number!\n");
        return;
    }

    fclose(fp);
    fclose(fp_tmp);

    remove(f_name);
    rename("temp.txt", f_name);
}

void update_data (char *f_name, int roll)
{
    FILE *fp, *fp_tmp;
    struct Student temp, record;

    printf("\nEnter new data:\n");
    printf("Name: ");
    getchar();
    scanf("%[^\n]", temp.name);
    printf("\nRoll number: ");
    scanf("%d", &temp.roll);
    printf("\nMarks in three subjects: ");
    for (int i=0; i<3; i++)
        scanf("%f", &temp.sub[i]);
    temp.per = (temp.sub[0] + temp.sub[1] + temp.sub[2]) / 3.0;

    // flag for checking if record present or not
    int f=0;
    fp = fopen (f_name, "r");
    if (fp == NULL)
    {
        printf("\nCannot open file!\n");
        return;
    }
    // temp file to copy the rest of the records
    fp_tmp = fopen ("temp.txt", "w");
    if (fp_tmp == NULL)
    {
        printf("\nCannot open temporary file!\n");
        return;
    }
    while (fread (&record, sizeof(struct Student), 1, fp))
    {
        if (record.roll == roll)
        {
            fwrite (&temp, sizeof(struct Student), 1, fp_tmp);
            printf("\nSuccessfully updated record!\n");
            f=1;
        }
    }
}

```

```

        else
            fwrite (&record, sizeof(struct Student), 1, fp_tmp);
    }

    if(!f)
    {
        printf("\nNo record found with the given roll number!\n");
        return;
    }

    fclose(fp);
    fclose(fp_tmp);

    remove(f_name);
    rename("temp.txt", f_name);
}

void display (char *f_name)
{
    FILE *fp = fopen (f_name, "r");
    if (fp == NULL)
    {
        printf("\nCannot open file!\n");
        return;
    }
    struct Student disp;
    printf("\n-----Student Details-----\n");
    printf("\nName\t\t\tRoll no\t\tSub 1\t\tSub 2\t\tSub 3\t\tPercentage");
    while (fread (&disp, sizeof(struct Student), 1, fp))
    {
        printf("\n%s\t%d\t\t%.2f\t\t%.2f\t\t%.2f\t\t%.2f %%", disp.name, disp.roll,
disp.sub[0], disp.sub[1], disp.sub[2], disp.per);
    }
    printf("\n\n-----\n");
    fclose(fp);
}

int main()
{
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");

    int n,ch,r;
    char *f = "student.txt";
    while (1)
    {
        A:
        printf("\nMENU\n1. Insert row.\n2. Delete row.\n3. Update row.\n4. Display.\n5.
Exit.\n");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: insert_data(f);
                    break;

```

```

        case 2: printf("\nEnter the roll number whose record is to be deleted: ");
                scanf("%d", &r);
                delete_data(f,r);
                break;

        case 3: printf("\nEnter the roll number whose record is to be updated: ");
                scanf("%d", &r);
                update_data(f,r);
                break;

        case 4: display(f);
                break;
        case 5: exit(0);
        default: printf("\nWrong choice! Enter again...\n");
                goto A;
                break;
    }
}
return 0;
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

MENU

1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.

1

Enter the name of the student: Faizan Choudhary

Enter the roll number of the student: 12

Enter the marks of the student in three subjects: 95

69

98

Inserted row successfully!

MENU

1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.

1

Enter the name of the student: Tirth B. Dalwadi

Enter the roll number of the student: 13

Enter the marks of the student in three subjects: 97

96

96

Inserted row successfully!

MENU

1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.

1

Enter the name of the student: M. Abbas Ansari

Enter the roll number of the student: 14

Enter the marks of the student in three subjects: 99

96

98

Inserted row successfully!

MENU

1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.

4

-----Student Details-----

Name	Roll no	Sub 1	Sub 2	Sub 3	Percentage
Faizan Choudhary	12	95.00	69.00	98.00	87.33 %
Tirth B. Dalwadi	13	97.00	96.00	96.00	96.33 %
M. Abbas Ansari	14	99.00	96.00	98.00	97.67 %

MENU

1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.

3

Enter the roll number whose record is to be updated: 12

Enter new data:

Name: Faizan Choudhary

Roll number: 12

Marks in three subjects: 95

96

95

Successfully updated record!

```

MENU
1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.
4

-----Student Details-----

Name          Roll no    Sub 1    Sub 2    Sub 3    Percentage
Faizan Choudhary  12      95.00    96.00    95.00    95.33 %
Tirth B. Dalwadi  13      97.00    96.00    96.00    96.33 %
M. Abbas Ansari   14      99.00    96.00    98.00    97.67 %

```

```

MENU
1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.
2

Enter the roll number whose record is to be deleted: 13

Record with the given roll number found, and deleted successfully!

```

```

MENU
1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.
4

-----Student Details-----

Name          Roll no    Sub 1    Sub 2    Sub 3    Percentage
Faizan Choudhary  12      95.00    96.00    95.00    95.33 %
M. Abbas Ansari   14      99.00    96.00    98.00    97.67 %

```

```

MENU
1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.
1

Enter the name of the student: Abbas Haider

Enter the roll number of the student: 14

Roll Number already exists! Try again...

Enter the roll number of the student: 15

Enter the marks of the student in three subjects: 93
95
95

Inserted row successfully!

```

```

MENU
1. Insert row.
2. Delete row.
3. Update row.
4. Display.
5. Exit.
4

-----Student Details-----

Name          Roll no    Sub 1    Sub 2    Sub 3    Percentage
Faizan Choudhary  12      95.00    96.00    95.00    95.33 %
M. Abbas Ansari   14      99.00    96.00    98.00    97.67 %
Abbas Haider      15      93.00    95.00    95.00    94.33 %

```


FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

13th December 2021

CODE: (code pasted in this format for readability)

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

struct Complex
{
    double real;
    double imag;
};
struct Complex A, B, C;

void display (struct Complex C) {
    printf("%lf + %lfi\n", C.real, C.imag);
}

void add () {
    C.real = A.real + B.real;
    C.imag = A.imag + B.imag;
}

void subtract () {
    C.real = A.real - B.real;
    C.imag = A.imag - B.imag;
}

void multiply () {
    // z1 = p + iq and z2 = r + is
    // z1 * z2 = (pr - qs) + i(ps + qr)
    double p = A.real, q = A.imag, r = B.real, s = B.imag;
    C.real = (p * r) - (q * s);
    C.imag = (p * s) + (q * r);
}

void divide () {
    // z1 = a + ib and z2 = c + id
    // z1 / z2 = (ac + bd)/(c*c + d*d) + i(bc - ad)/(c*c + d*d)
    double a = A.real, b = A.imag, c = B.real, d = B.imag;
    C.real = (a*c + b*d) / (c*c + d*d);
```

```

    C.imag = (b*c - a*d) / (c*c + d*d);
}

double extract (char c[]) {
    int digits=1;
    bool flag=false;
    double number = 0;
    int num=0,i;
    for (i=0; i<strlen(c); i++) {
        if (c[i] == ' ')
            continue;
        if (c[i] == '.')
            // for decimal point numbers
            flag=true;
        else if (isdigit (c[i])) {
            // appending digits
            num = c[i] - '0';
            number = number * 10 + num;
            // if flag is true, it means there is some decimal point number (. has been
encountered)
            if (flag)
                digits *= 10;
        }
    }
    // updating number to have correct decimal place after division
    number = number / digits;
    // printf("%lf\n", number);
    return number;
}

void parse (char c[100]) {
    int i, k=0, counter=0;
    int neg = 1;
    double n;
    // to parse out numbers and pass it to extract() fxn
    char subs[50];
    for (i=0; i<strlen(c); i++) {
        if (c[i] == '-')
            // for negative numbers
            neg = -1;
        if (c[i] == 'i' || c[i] == ' ') {
            subs[k++] = '\0';
            n = extract(subs);
            k = 0;
            counter++;
            if (counter == 1) {
                A.real = (neg * n);
                neg = 1;
            }
            else if (counter == 2) {
                A.imag = (neg * n);
                neg = 1;
            }
            else if (counter == 3) {

```

```

        B.real = (neg * n);
        neg = 1;
    }
    else if (counter == 4) {
        B.imag = (neg * n);
        neg = 1;
    }
    // to skip ' + ' and ', '
    i+=2;
}
// copying number into a secondary string
subs[k++] = c[i];
}
}

int main() {
    char input[100];
    int i,ch;
    int flag = 0;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n");

    B:
    printf("\nEnter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -
4.23 + 6.9i): ");
    if (flag == 1)
        getchar();
    scanf("%[^\\n]*c", &input);

    parse(input);
    // printf ("%lf %lf %lf %lf", A.real, A.imag, B.real, B.imag);
    while (1) {
        flag = 1;
        printf("\nEnter complex numbers:\n");
        display(A);
        display(B);

        A:
        printf("\nMENU:\n1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5.
Enter number again\n6. Exit\n");
        scanf("%d", &ch);

        switch (ch)
        {
            case 1: printf("\nAfter Addition: ");
                    add();
                    display(C);
                    break;
            case 2: printf("\nAfter Subtraction: ");
                    subtract();
                    display(C);
                    break;
            case 3: printf("\nAfter Multiplication: ");
                    multiply();
                    display(C);

```

```

        break;
    case 4: printf("\nAfter Division: ");
            divide();
            display(C);
            break;
    case 5: goto B;
    case 6: exit(0);
    default: printf("\nWrong choice! Enter again...\n");
             goto A;
             break;
    }
}
return 0;
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

Enter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -4.23 + 6.9i): -5 + 8.69i, 7.42 + -8.69i

Entered complex numbers:

-5.000000 + 8.690000i

7.420000 + -8.690000i

MENU:

1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Enter number again
 6. Exit
- 1

After Addition: 2.420000 + 0.000000i

Entered complex numbers:

-5.000000 + 8.690000i

7.420000 + -8.690000i

MENU:

1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Enter number again
 6. Exit
- 3

After Multiplication: 38.416100 + 107.929800i

Entered complex numbers:

-5.000000 + 8.690000i

7.420000 + -8.690000i

MENU:

1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Enter number again
 6. Exit
- 5

Enter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -4.23 + 6.9i): 0.123 + -9.0i, -4.23 + 6.9i

Entered complex numbers:

$0.123000 + -9.000000i$

$-4.230000 + 6.900000i$

MENU:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Enter number again

6. Exit

2

After Subtraction: $4.353000 + -15.900000i$