==========

Roll Number: SYCOC303 Division: C

PRN Number: 122B2B303 Batch: C4

Name: VINAYAK MADAN SHETE

==========

**Problem Statement:**

- ⇨ Implement matrix class as ADT. Write a program to perform matrix addition, subtraction, and multiplication. In read matrix function, raise an exception if any attempt is made to have rows and columns beyond the array size. Raise an exception if any attempt is made to perform matrix operations on matrices which does not satisfy the matrix order criteria. Implement using C++.

==========

# INPUT:

```
/*
 *     ====================================
 *            Program Name: MatrixOperations.cpp
 *     Created on: December 06, 2022
 *       Author: Vinayak Shete
 *  ====================================
 */

#include<iostream>
#define maxSize 25
using namespace std;

class Matrix
{
	int mat1[10][10],mat2[10][10],resAdd[10][10],resSub[10][10],resMul[10][10];
	int nrows,ncols;
	public:
		void readData()
		{
```

```
                int matSize,e;
                try
                {
                        cout<<"\nEnter the number of rows and columns: ";
                        cin>>nrows>>ncols;
                        matSize=nrows*ncols;
                        if(matSize>maxSize)
                        {
                                throw e;
                        }
                        else
                        {
                                cout<<"\nEnter the values row wise in the MATRIX
1:\n";
                                for(int i=0;i<nrows;i++)
                                {
                                        for(int j=0;j<ncols;j++)
                                        {
                                                cin>>mat1[i][j];
                                        }
                                }

                                cout<<"\nEnter the values row wise in the MATRIX
2:\n";
                                for(int i=0;i<nrows;i++)
                                {
                                        for(int j=0;j<ncols;j++)
                                        {
                                                cin>>mat2[i][j];
                                        }
                                }
                        }
                }
                catch(int e)
                {
                        cout<<"\nException Caught!!==>>>\nThe size of the matrix
eceeds than the MAX SIZE...Stopping the Program!";
                        exit(0);
```

```
                }
                cout<<"\n======\nThe data is read successfully and stored in both
the matrices!\n======";
        }


        void addMatrices()
        {
                for(int i=0;i<nrows;i++)
                {
                        for(int j=0;j<ncols;j++)
                        {
                                resAdd[i][j]=mat1[i][j]+mat2[i][j];
                        }
                }
                cout<<"\nThe resultant matrix after ADDITION:\n";
                for(int i=0;i<nrows;i++)
                                {
                                        for(int j=0;j<ncols;j++)
                                        {
                                                cout<<resAdd[i][j]<<"  ";
                                        }
                                        cout<<endl;
                                }
        }


        void subMatrices()
        {
                for(int i=0;i<nrows;i++)
                {
                        for(int j=0;j<ncols;j++)
                        {
                                resSub[i][j]=mat1[i][j]-mat2[i][j];
                        }
                }
                cout<<"\nThe resultant matrix after SUBTRACTION:\n";
                for(int i=0;i<nrows;i++)
                                {
```

```
                                        for(int j=0;j<ncols;j++)
                                        {
                                                cout<<resSub[i][j]<<"  ";
                                        }
                                        cout<<endl;
                                }
                }

                void mulMatrices()
                {
                        for(int i=0;i<nrows;i++)
                        {
                                for(int j=0;j<ncols;j++)
                                {
                                        resMul[i][j]=0;
                                        for(int k=0;k<ncols;k++)
                                        {
                                                resMul[i][j]+=mat1[i][k]*mat2[j][k];
                                        }
                                }
                        }
                        cout<<"\nThe resultant matrix after MULTIPLICATION:\n";
                        for(int i=0;i<nrows;i++)
                                        {
                                                for(int j=0;j<ncols;j++)
                                                {
                                                        cout<<resMul[i][j]<<"  ";
                                                }
                                                cout<<endl;
                                        }
                }
};
int main()
{
        int ch;
        Matrix m;
        cout<<"\n======WELCOME=======";
```

```
        cout<<"\nAdd Values for the Matrices:";
        m.readData();
        cout<"\n=======ADDITION OF MATRICES==========";
        m.addMatrices();
        cout<"\n=======SUBTRACTION OF MATRICES==========";
        m.subMatrices();
        cout<"\n=======MULTIPLICATION OF MATRICES==========";
        m.mulMatrices();
        cout<<"\n=====THANK YOU!======";
        return 0;
}
```

==========

# OUTPUT:

```
======WELCOME=======
Add Values for the Matrices:
Enter the number of rows and columns: 3 3

Enter the values row wise in the MATRIX 1:
2 2 1
1 5 0
0 0 1

Enter the values row wise in the MATRIX 2:
5 7 1
0 3 0
1 0 8

======
The data is read successfully and stored in both the matrices!
======
The resultant matrix after ADDITION:
7  9  2
1  8  0
1  0  9

The resultant matrix after SUBTRACTION:
-3  -5  0
1  2  0
-1  0  -7

The resultant matrix after MULTIPLICATION:
25  6  10
40  15  1
1  0  8

=====THANK YOU!======
--------------------------------
Process exited after 35.72 seconds with return value 0
Press any key to continue . . .
```

**The program will stop its execution as Exception is caught:**

```
======WELCOME=======
Add Values for the Matrices:
Enter the number of rows and columns: 6 5

Exception Caught!!==>>>
The size of the matrix eceeds than the MAX SIZE...Stopping the Program!
--------------------------------
Process exited after 58.87 seconds with return value 0
Press any key to continue . . .
```

==========