Project 1
Matrix operations (100pts)

Submission instruction:

1. This project is due at **11:55pm on Tuesday March 12.**
2. **Late submission is due at 11:55pm on Wednesday March 13 with 20 pts deduction.**
3. All students must submit their project through **Sakai dropbox**.
4. Only one .cpp file named "**project1.cpp**" is required for submission.
5. The file structure should be **<your-folder>/project1/project1.cpp**.

For this project you are required to perform Matrix operations (**Addition, Subtraction, Multiplication, Determinant, Transpose and Inverse**). For each of the operations mentioned above you have to make a function in addition to two other functions for 'Inputting' and 'Displaying' the Matrices in Row-Order Format. (In total there will be 8 functions: 6 for the operations and 2 functions for inputting and displaying the data.)

- For Addition, Subtraction, Multiplication, and Transpose, it can be 2D array of any size (say Max dimension for both row and Column be 10). So, to prevent users from having to entering 100 (for a 10X10 matrix) elements every time (even for a smaller matrix), you should ask for the number of rows and columns in the matrix before the user would input the values of the matrix elements. Basically, your code should be generic. The point is user should be free to enter 2X3, or 4X2, or 9X9 Matrix. If the operation is not supported (ex. adding two arrays of different dimensions) just print out that the operation is not supported and return to the menu.

- For Determinant and Inverse operations, we are giving you the relaxation to choose fixed matrix size of 3X3. (3 rows and 3 column). You do not need to calculate the determinate and inverse operations for any matrix that is not 3x3 (the function can just check to see if the matrix not 3x3. If it is, just print out that the operation is not supported).

- You can write the above code using either Matrices (2D arrays) or Vectors.

Requirements:

In the main function you should have a menu in which you will ask the user his or her choice for which of the operations he or she wants to perform. For Ex.

```
cout<<"\n Menu ";

cout<<"\n Choice 1";

cout<<"\n Choice 2";
```

```
cout<<"\n Choice 3";
            :
            :
cout<<"\n Choice n";
cout<<"\n Enter your Choice";
cin>>choice;
```

After taking the choice value, one should have a switch network to choose which function (for the matrix operation) needs to be called.

Extra Credit: Extra points will be given to people who program in such a manner that after completion of one operation, it should display the menu again and ask user his choice whether he wants to quit or he want to perform another operation. (10pts)

Points we will be grading this project:

1) We want to check whether you understand call by reference and call by value and you are using in right perspective. (20pts)
2) You are taking full use of the functions. (20pts)
3) You are not doing the same mistakes that's mentioned in guidelines and some other you know. (20pts)
4) How correctly you are passing data between the function. (20pts)
5) Check some sample examples. (20pts)

Example:

```
Menu
Choice 1: addition
Choice 2: subtraction
Choice 3: multiplication
Choice 4: determinant
Choice 5: transpose
Choice 6: inverse
Choice 0: exit
```

```
 Enter your choice: 1

 matrix1 # of rows: 2

 matrix1 # of columns: 2

 Enter element (0,0): 1

 Enter element (0,1): 2

 Enter element (1,0): 2

 Enter element (1,1): 1
1 2
2 1

 matrix2 # of rows: 2

 matrix2 # of columns: 2

 Enter element (0,0): 2

 Enter element (0,1): 3

 Enter element (1,0): 3

 Enter element (1,1): 4
2 3
3 4
```

```
3 5
5 5

 Menu
 Choice 1: addition
 Choice 2: subtraction
 Choice 3: multiplication
 Choice 4: determinant
 Choice 5: transpose
 Choice 6: inverse
 Choice 0: exit
 Enter your choice: 3

 matrix1 # of rows: 2

 matrix1 # of columns: 1

 Enter element (0,0): 2

 Enter element (1,0): 1
2
1


 matrix2 # of rows: 1

 matrix2 # of columns: 3
 Enter element (0,0): 1


 Enter element (0,1): 2


 Enter element (0,2): 3
1 2 3


2 4 6
1 2 3


 Menu
 Choice 1: addition
 Choice 2: subtraction
 Choice 3: multiplication
 Choice 4: determinant
 Choice 5: transpose
 Choice 6: inverse
 Choice 0: exit
 Enter your choice: 5


 matrix # of rows: 2


 matrix # of columns: 3


 Enter element (0,0): 1


 Enter element (0,1): 2


 Enter element (0,2): 3
 Enter element (1,0): 4


 Enter element (1,1): 5


 Enter element (1,2): 6
1 2 3
4 5 6


1 4
2 5
3 6


 Menu
 Choice 1: addition
 Choice 2: subtraction
 Choice 3: multiplication
 Choice 4: determinant
 Choice 5: transpose
 Choice 6: inverse
 Choice 0: exit
 Enter your choice: 4


 matrix # of rows: 3


 matrix # of columns: 3


 Enter element (0,0): 1
 Enter element (0,1): 1

 Enter element (0,2): 1

 Enter element (1,0): 1

 Enter element (1,1): 2

 Enter element (1,2): 3

 Enter element (2,0): 1

 Enter element (2,1): 3

 Enter element (2,2): 5
1 1 1
1 2 3
1 3 5

0
 Menu
 Choice 1: addition
 Choice 2: subtraction
 Choice 3: multiplication
 Choice 4: determinant
 Choice 5: transpose
 Choice 6: inverse
 Choice 0: exit
 Enter your choice: 6
 matrix # of rows: 3

 matrix # of columns: 3

 Enter element (0,0): 1

 Enter element (0,1): 0

 Enter element (0,2): 0

 Enter element (1,0): 0

 Enter element (1,1): 1

 Enter element (1,2): 0

 Enter element (2,0): 0

 Enter element (2,1): 1

 Enter element (2,2): 2
1 0 0
0 1 0
0 1 2

1 0 0
0 1 0
0 -0.5 0.5
```