CS1303: Introduction to Programming
Assignment 9:
Matrix Multiplication and Inverse

Submission Deadline:
~~Monday,11th November 2019~~ Tuesday, 12th November 2019, 10:00 PM

Problem Statement:
Use two-dimensional arrays to solve the following problem. You have to write a program to determine the multiplication and inverse of two matrices. The condition for matrix multiplication is that the number of columns of the 1st matrix must be equal to the number of rows of the 2nd one. As a result of matrix multiplication, you will get a new matrix that has the number of rows equal to the 1st matrix and the number of columns equal to the 2nd one. For example, if you multiply a matrix of **a*b** by **b*c** size you'll get a new one of **a*c** dimension. For the inverse of the matrix, the condition is that the matrix should be square matrix and determinant of the matrix should be non-zero.

You have to develop an application that multiplies two matrices that will be entered by the user. If the orders of the matrices are such that they can't be multiplied by each other, then an error message is displayed, and the user is asked to enter the correct order of the matrices again. The user will enter the order of the matrices first and then enter the elements of both the matrices.

For example the user will first enter the order of matrix **A** as **m n**, where **m** is the number of rows and **n** is the number of columns in A. Then enter the order of matrix **B** as **p q** where **p** is the number of rows and **q** is the number of columns in **B**. If order of the matrices is such that they cant be multiplied the user will be asked to enter the orders again. Next, enter the elements of matrix **A** and matrix **B** as shown in the sample test case. After this, you have to multiply **A[m][n] * B[p][q]** and store the multiplication output in another matrix **C[m][q]**. Then you have to find the inverse of the matrix **C[m][q]**. It may be possible that the inverse of the matrix does not exist, in that case, you have to print the message that inverse not possible.

So you have to develop **two functions** (for more clarity please refer below example):

1. Function-1 is multiplication of two matrices **A** and **B**: i.e., **C = A * B**
2. Function-2 is inverse of the resultant matrix **C** i.e., you have to find $\mathbf{C}^{-1}$

1) Function-1: multiplication of two matrices:

After reading all the values for matrix **A** and matrix **B**, develop a function that multiplies **A** and **B** and stores the output in another matrix **C**. You have to do necessary error handling for the order of the matrix for multiplication.

Order of matrix A[m][n] = **2 2**
Order of Matrix B[p][q] = **2 2**
A[2][2] =

```
1    2
3    4
```

B[2][2] =

```
5    6
7    8
```

Output 1:

C[2][2] =

```
19    22
43    50
```

2) Function-2: inverse of the resultant matrix C:

Once you get the matrix C from function-1 you have to find the inverse of C. If C is not a square matrix or determinant of the C is 0 then the inverse of C does not exist and the program should print the message **“the inverse of the matrix is not possible”** otherwise print the inverse matrix of C.

Output 2:

C^{-1}[2][2] =

```
 12.5      -5.5
-10.75      4.75
```

Inputs to the Program:

1. Order of matrix **A** and **B**, i.e., the value of **m** and **n** for **A** and **p** and **q** for **B**.
2. Elements of matrix **A (in a row-wise order)**
3. Elements of matrix **B (in a row-wise order)**

Output:

1. Multiplication of two matrices **A** and **B** stored in **C**.
2. The inverse of the resultant matrix **C** if exist i.e., **C^{-1}**.

Notes:

1. The smallest matrix should be 2*2 If a user enters the order of the matrix smaller than 2*2 (including 2*1,1*2 and 1*1) then the program should ask the user to enter the order again. Please follow the input- and output format strictly.
2. For whatever not mentioned in the problem statement, you are free to take a design decision but mention that in the Design.txt file.

Sample Test Cases:

1) **Case1:**

```
Enter the order of matrix A: 2  2
Enter the order of matrix B: 2  2

Enter the elements of matrix A[2][2]:
1  2 3  4
Enter the elements of matrix B[2][2]:
5  6 7  8

Multiplication Matrix C[2][2]:
19      22
43      50
Inverse Matrix of C:
12.5      -5.5
-10.75    4.75
```

2) **Case2:**

```
Enter the order of matrix A: 2  3
Enter the order of matrix B: 2  5
Error: Enter the orders again:
Enter the order of matrix A: 2  3
Enter the order of matrix B: 3  2

Enter the elements of matrix A[2][3]:
1 2 3 4 5 6
Enter the elements of matrix B[3][2]:
7 8 9 10 11 12

Multiplication Matrix C[2][2]:
58      64
139     154
Inverse Matrix of C:
 4.27   -1.77
-3.86   1.61
```

3) **Case3:**

```
Enter the order of matrix A: 1  3
Enter the order of matrix B: 2  1
Error: Enter the orders again:
Enter the order of matrix A: 3  2
Enter the order of matrix B: 2  4

Enter the elements of matrix A[3][2]:
1  2 3  4 5 6
Enter the elements of matrix B[2][4]:
7 8 9 10 11 12 13 14

Multiplication Matrix C[3][4]:
29      32      35      38
65      72      79      86
101     112     123     134

Inverse Matrix of C:
The inverse of the matrix is not possible
```

Submission Details:

Please submit the following information:

- **Source Code:** Your source program. The name of your file should be in this format: **Matrix-roll no.c** where you replace "roll no" with your roll number.

- **Readme.txt:** In this file, you should explain how to compile and run your program. The name of your file should be in this format: **Matrix-Readme-roll no.txt** where you replace "roll no" with your roll number.

- **Design.txt:** In this file, you explain the design of your program (control flow of your program). Your objective should be such that the TA reading this file should easily understand the working of your program. Please add details about how you have handled corner cases - i.e. for what inputs you have printed "Error". The name of your file should be in this format: **Matrix-Design-roll no.txt** where you replace "roll no" with your roll number.

Zip all these files and name it as **Matrix-roll no.zip**. **Please follow the naming convention strictly. Otherwise, your program will not be evaluated.** Then, submit it on google classroom for this assignment by the above-mentioned deadline.

Plagiarism policy: If we find a case of plagiarism in your assignment (i.e. copying of code from each other, in part or whole), you will be awarded **zero marks**. **Note** that we will not distinguish between a person who has copied, or has allowed his/her code to be copied; both will be equally awarded **zero** marks for the submission. Follow the link below for more information about plagiarism policy: **https://cse.iith.ac.in/academics/plagiarism-policy.html**

Evaluation Policy:
The TAs will use the following evaluation policy:

- Design: 30%
- Execution: 60%
- Indentation and Documentation (with comments): 10%

Late Submission Penalty:
For each day after the deadline, your submission will be penalized by 10 marks.