**ESE 224, Project 3: Matrix Operations, ECE, SBU, M. Subbarao, Fall 2018**

**Assigned: 9/18/2018. Due: 10/7/2018. Weight: 8%.**

**Part A:** Write a program to do the following:

1. The main() function calls a function to read and MxN matrix (double A[M][N]) of double type real numbers from a specified file— input1.txt. M and N are given constants. Write a general readmat() function to do this. Use #include <fstream> to open and read and write file streams (do not use cin and cout ).
2. The main() function calls a function to print A[M][N] to a specified file— output1.txt. Write a general printmat() function to do this.
3. The main() function calls a function to copy A[M][N] to another array D[M][N]. Write a general copymat() function to do this.
4. The main() function calls a function to initialize a matrix B[N][N} to be an identity matrix. Write a general identmat() function to do this.
5. The main() function calls a function invmat() to find the inverse of A using the Gauss-Jordan elimination method. The function invmat() should be called with A and B as parameters. The inverse of A is returned in B, and the identity matrix is returned in A. The invmat() function should be general in that it should work for any value of N.

References:

[**https://www.intmath.com/matrices-determinants/inverse-matrix-gauss-jordan-elimination.php**](https://www.intmath.com/matrices-determinants/inverse-matrix-gauss-jordan-elimination.php)

**http://www.it.uom.gr/teaching/linearalgebra/chapt6.pdf**

[**https://www.codewithc.com/wp-content/uploads/2014/05/gauss-elimination-flowchart-forward-elimination-and-back-substitution.png**](https://www.codewithc.com/wp-content/uploads/2014/05/gauss-elimination-flowchart-forward-elimination-and-back-substitution.png) **[Note: I think there is a typo in the last box on left; it should be**

**bij = bij – mik \* bkj ; check this.]**

1. The main functions calls the printmat() function to print A[M][N], and B[M][N] to the specified file— output1.txt.
2. The main() function calls a function multmat() to set C= D\*B where C[N][N] is the product of the matrices D and B. The function multmat() should be called with D, B, and C matrices and M,N, as parameters. The multmat() function should be general in that it should work for any value of M,N (in this case M=N but the function should work even when M is not equal to N).
3. The main functions calls the printmat() function to print C[M][N].

Store the following data in a file named input1.txt in the same directory as the source code of the main function.

7 13 9 14 12

4 8 10 15 16

11 6 5 17 2

18 19 20 21 22

23 24 25 26 27

Use “const int M=5, N=5;” to specify the M,N, as **global constants outside the main() above all other function definitions.**

Run the program and write its output into output1.txt file.

Submit your source code, input file, and the output file.

**Part B:**

Solve the same problem as in Part A by defining a **class** named **matrix** of fixed size MxN where M=N=5 are constant integers. Define suitable methods and members.