Programming Project 2 (due 09/30/2020)

Write a program that asks the user for the number of linear equations to solve (let's say n <=10) using the following two methods.
1) Jacobi iterative method
2) Gauss-Seidel method (again, the user has to enter the desired error that will be used for stopping condition).

Ask the user to first enter the number of equations and then give them the choice to enter the coefficients from the command line (by asking for each row that includes the b value) or have them enter a file name which has the augmented coefficient matrix (including the b values) in a simple text file format as seen below for an example of 3 equations, similar to Project 1:

E.g. the contents of a file for 3 linear equations 5x-y = 7, -x+3y-z=4, -y+2z=5 will be
5 -1 0 7
-1 3 -1 4
0 -1 2 5

After reading in the equation data, ask the user to enter the desired stopping error. The user is then asked to enter the starting solution for the iterative methods, eg for the above equations, it could be

0 0 0

For both the iterative methods, calculate the L2 norm after each iteration and when this value is less than the error the user specified, then stop the iterations.
Print the x column vector (in the row format) after each iteration. If the iterative methods don't achieve the desired error in 50 iterations, end the iterative method and mention that the error was not reached.

The output of your program should be the solution at each iteration in the following format:
$[2.0870\ 3.4348\ 4.2174]^T$

Upload your code and executable (in a zip file like last project) and report (if you did extra credit part) Please make sure the report is in PDF or Word DOC format.

Extra credit: Test your program with large values of n, say like 20, 50, 100 etc. Generate the coefficients randomly for such large equations. Comments on the time taken by your program as the number of equations increase. Do you run into any other problems? Write a one page report on this extra credit part. Show a graph of how the time increases as the number of equations increase (3 curves on this graph, one for each method, including the Gaussian Elimination with Scaled partial pivoting method from previous project)