1. Create a user-defined MatLab function that implements the Gauss elimination method called Gauss_alt. The input arguments would matrix [**a**] and [**b**] from linear algebra ([**a**]*[**x**]=[**b**]) and the output [**x**]. If the input argument [**b**] is a row vector, the function should be able to transform it to a column vector. Use the augmented matrix for [**a**|**b**] for all operations in the function. Solve the following set of equations **by hand** (explain the steps you are following):

$$x_1 + 2x_2 - 2x_3 = 9$$
$$2x_1 + 3x_2 + x_3 = 23$$
$$3x_1 + 2x_2 - 4x_3 = 11$$

   Use the function Gauss_alt to validate your hand-calculated solution. (25%)

2. Write a user-defined MatLab function that calculates the infinity norm of any matrix called InfNorm. The input argument will be a matrix and the output argument will be the value of infinity norm. Calculate the infinity norm of the following matrix [**A**] **by hand** and validate your hand-calculated solution by using the function InfNorm. (20%)

$$[\mathbf{A}] = \begin{bmatrix} 6 & 3 & 11 & -1 & 2 \\ 3 & -2 & 7 & 0 & 4 \\ 3 & 2 & -6 & 5 & -3 \\ -5 & 7 & 1 & -4 & 0 \end{bmatrix}$$

3. (i) Use Doolittle LU decomposition **by hand** to determine L (lower triangular matrix) and U (upper triangular matrix) of the following matrix:

$$[\mathbf{A}] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 2 \end{bmatrix}$$

   (ii) If you are asked to solve the set of equations [**A**][**x**]=[**b**] and you are given many different vectors [**b**] which method will be more efficient to use Gauss Elimination or LU Decomposition? Why?(15%)

4. For the system of equations [**A**][**x**]=[**b**] use Gauss-Jordan elimination method **by hand** to define [**x**], for (15%):

$$[\mathbf{A}] = \begin{bmatrix} 2 & -4 & 1 \\ 5 & 3 & -2 \\ -1 & 7 & -3 \end{bmatrix}; \quad [\mathbf{b}] = \begin{bmatrix} 3 \\ 8 \\ -4 \end{bmatrix}$$

5. Write a user-defined MatLab function that solves the system [**A**][**x**]=[**b**] with the use of Gauss-Jordan Elimination method. The program should include pivoting in which the pivot row is switched with the row that has a pivot element with the largest absolute numerical value. For the function name and arguments use x = GaussJordan(A,b), where A is the matrix of coefficients, b is the right-hand-side column of constants, and x is the solution. Check with question 4. (25%)

The only build-in functions you are allowed to use from MatLab are the following ones: zeros, size, sum, abs, length, clear, close, clc.