2. Gaussian Elimination

1. Write an M-file to compute Gaussian Elimination. The name of the file be `gaussel1`

   - Input parameters: the coefficient matrix (A) and the right-side vector ($\underline{b}$) of LES.
   - Output argument: the solution vector $\underline{x}$
   - Use the Matlab row-operations for organisation of algorithm.
   - If GE can't be solved without row or coloumn swap write an error message and terminate the program.
   - In case of underdetermined LES give a base solution and warn the user of this.
   - In case the user asked it, display the matrices $A^{(i)}$ during computation.
   - To checking our function we can use the exercises from numerical I.

   +1 We can prepare our function to accept LES with multiple right sides.

2. Extend the previous m-file (but save as a new name for example `gaussel2`) with steps of partial and whole pivoting method.

   - Using partial or whole pivoting could be choosen by user (for example according to a boolean input parameter), but if the partial pivoting is stucked then automatically switch to whole pivoting method. If we used whole pivoting despite user has choosen partial pivoting, then inform user about the switch
   - Give an opportunity for displaying matrices $A^{(i)}$ during computation. Don't forget that pivoting method can be changed the matrix so we have to display it after row and coloumn swap.
   - Don't forget that the pivoting method can be changed the solution.

3. Apply Gaussian elimantion for computing inverse of an square matrix.

   The name of function: `gaussel3`

   - Check input argument(s) before computing
   - Compute the determinant of matrix.
   - Give the LU decomposition of the matrix using the matrices $L^{(i)}$.

   – If you have written the function `gaussel1` such that it accepts multiple right-sides, then you can call it during computation.