Exercise 1 50% (MATLAB)

Replace the symbols X in the matlab code below to solve the linear system

\[ Ax = b \]

when \( A \) is tri-diagonal, i.e. the coefficients satisfy \( a_{ij} = 0 \) whenever \( |i - j| > 1 \).

```matlab
%%% GaussTri %%%
%% Input: tridiagonal square matrix A (not checked)
%% vector b of corresponding size (not checked)
%% Output: solution to Ax=b

function x=GaussTri(A,b)

N=size(A,1);
for i=1:N-1
    %multiply the ith row by pivot
    p=1/A(i,i);
    A(i,X) = p*A(i,X);
    b(i) = p*b(i);
    % eliminate the ith column
    A{1,i+1} = A{1,i+1} - X;
    b[i+1] = b(i+1) - A(X,X)*b(i);
end
%last step
p = 1/A(N,N);
end
b(N)=p*b(N);
%once the matrix is upper triangular (with one on the diagonal)
%solve (the solution is stored in b)
for i=N-1:-1:1
    b(i)=b(i)-A{1,i+1}*b(i+1);
end
x=b;
%%% END %%%% 

Test your routine with

\[ A = \text{gallery('tridiag',50,-1,2,-1)}; \]
\[ b = [1:50]' \].
Exercise 2  50% (MATLAB)

The MATLAB code below is an implementation of the Gaussian elimination algorithm to solve the linear system

\[ Ax = b. \]

As we shall see in class, it may happen that the algorithm finds a vanishing (or small) pivot. To cope with this issue, at the beginning of the first for loop, it is necessary to flip the \( i \)th row with the \( j \)th row such that \( j \geq i \) and \( |a_{ji}| = \max_{k \geq j} |a_{ki}| \). This is called partial pivoting. Modify the Gaussian algorithm below to include partial pivoting. Test your routine with

\[
A = \begin{bmatrix} 0 & 1 & 3 \\ 5 & 2 & 3 \\ 6 & 8 & 1 \end{bmatrix}; \\
b = \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix};
\]

%%%% Gauss %%%%
%% Input: Square matrix A (not checked)
%% vector b of corresponding size (not checked)
%% Output: solution to Ax=b
function x=Gauss(A,b)

N=size(A,1);
for i=1:N-1
  % multiply ith row by pivot
  p=1/A(i,i);
  for j=i+1:N
    A(i,j) = p*A(i,j);
  end
  b(i) = p*b(i);

  % eliminate the ith column
  for k=i+1:N
    for j=i+1:N
      A(k,j)=A(k,j)−A(k,i)*A(i,j);
    end
    b(k) = b(k)−A(k,i)*b(i);
  end
end

% last row
p = 1/A(N,N);
b(N)=p*b(N);

% solve for x
for i=N-1:-1:1
  for j=i+1:N
    b(i) = b(i) − A(i,j)*b(j);
  end
end

x=b;
%%% END %%%