You are to construct a code for Gaussian elimination with column pivoting. That code will produce the factorization

\[ A = LU^T \]

where \( L \) and \( U \) are upper triangular, and \( P \) is permutation matrix. It will be constructed using a function

\[ \text{function } [L, U] = \text{columnLU}(A) \]

where the column permutations are incorporated into \( U \) and \( L \) is always lower triangular.

At the beginning of your function, put the statements

\[ n = \text{length}(A); \quad \% \text{You may assume that } A \text{ is square} \]
\[ p = 1:n; \]

At each step of Gaussian elimination, you are to put the largest uneliminated element in row \( k \) into the pivot position. Thus at each step, you find \( j_{\text{max}} \) such that

\[ |a_{k,p(j_{\text{max}})}| = \max_{k \leq j \leq n} |a_{k,p(j)}| \]

Then if \( j_{\text{max}} > k \) swap indices \( p(j_{\text{max}}) \) and \( p(k) \) of \( A \), but leave the columns of \( A \) in position.

To estimate the condition number, you may use a MATLAB function that I have written for you with the calling sequence

\[ [\text{cond, vec, magick}] = \text{Cond1}(L, U, A). \]

Here \( \text{cond} \) is the condition number \( \kappa_1(A) = \|A\|_1\|A^{-1}\|_1 \), \( \text{vec} \) is the vector such that \( \|\text{vec}\|_\infty = 1 \), \( \|A^{-T}\text{vec}\|_\infty \approx \|A^{-1}\|_1 = \|A^{-T}\|_\infty \), and \( \text{magick} \) is a column of the identity matrix such that \( \|A^{-1}\text{e}_{\text{magick}}\|_1 = \|A^{-1}\|_1 \). This function will be in the m-file \text{Cond1.m} on the class web page. Test your routine with the matrices generated by the m-files \text{matrix1.m}, \text{matrix2.m}, \text{matrix3.m} on the class web page.