1. [This exercise is to be done ‘by hand’ not with matlab]
   (a) Solve the linear system $Ax = b$ by Gaussian elimination where:
   \[
   A = \begin{pmatrix}
   1 & -2 & -1 & 0 \\
   -2 & 3 & 2 & 1 \\
   -1 & 2 & 3 & -2 \\
   0 & -1 & -4 & 6
   \end{pmatrix},
   b = \begin{pmatrix}
   -3 \\
   5 \\
   1 \\
   4
   \end{pmatrix}.
   \]
   (b) What is the LU factorization of $A$? What is its determinant?
   (c) Using the LU factors obtained in (b) find the last column of the inverse of $A$, without computing the whole inverse.

2. Let $A = LU$ the LU factorization of $A \in \mathbb{R}^{n \times n}$, with $|l_{ij}| \leq 1$. Verify the equation:
   \[
   u_{i,:} = a_{i,:} - \sum_{j=1}^{i-1} l_{ij} u_{j,:},
   \]
   [Matlab notation used: $x_{i,:}$ is the $i$th row of matrix X] Then use this relation to show that $\|U\|_{\infty} \leq 2^{n-1}\|A\|_{\infty}$ (Hint: Use induction).

3. (a) Determine the standard LU factorization of the matrix on the right.
   (b) Compute the determinant of $A$
   (c) Compute the first column of the inverse of $A$.
   (d) Repeat the above questions when partial pivoting is used, i.e., find the permutation matrix $P$ and the matrices $L, U$ such that $PA = LU$, compute the determinant of $A$ based on this factorization, and compute the first column of the inverse of $A$, based on this factorization.

4. We saw that Gaussian elimination is equivalent to multiplying the initial matrix $A$ by a sequence of Gaussian transformations (called $M_k$ in the notes) from the left. This exercise explores what happens in the case of Gauss-Jordan (GJ) elimination.
   (a) Show that for Gauss-Jordan we have the same result:
   \[
   A_k = M_k A_{k-1}, \ k = 1, 2, \ldots, n,
   \]
   with $A_0 = A$ (Note that now $k$ runs from 1 to $n$ steps instead of 1 to $n - 1$). However the matrices $M_k$ are different. What are the new transformations $M_k$? [Hint: they are of the form $I - u_k e_k^T$ again. Find $u_k$]
   (b) In the case of Gaussian elimination the product of the $M_k$’s is lower triangular and this gives the LU factorization. Is the product of the $M_k$’s triangular for Gauss-Jordan elimination? Show that the last matrix (diagonal) obtained by GJ satisfies
   \[
   D = M.A \quad \text{with} \quad M = M_n M_{n-1} \cdots M_1.
   \]
Show how you can practically store all the matrices $M_1, M_2, \cdots, M_n$ in a single $n \times n$ array? Write a ‘Gauss-Jordan factorization’ script which produces the diagonal $D$ as a vector and the matrices $M_1, \ldots, M_n$ stored as just specified. [Hint: if a zero pivot is encountered just exit with an error. The input is a matrix $A$, the output consists of an array $M$ and a vector $d$]

(c) Recall that the LU factorization can help solve several linear systems with the same matrix $A$. How would you exploit the output of the factorization described in (b) for the same task? Write a matlab script that uses this output to solve $p$ linear systems with $A$ when the right-hand sides are stored in $B \in \mathbb{R}^{n \times p}$.

(d) Explain how you would compute the inverse of a matrix using the Gauss-Jordan algorithm. Apply this method and the scripts you developed in (b) and (c) to compute the inverse of the $4 \times 4$ matrix $A$ of question 1. [Hint: Print the (few) lines showing the execution of the matlab commands you use to get the inverse.]

5. An $n \times n$ matrix $A$ is said to be ‘strictly row diagonally dominant’, if:

$$|a_{ii}| - \sum_{j=1,j \neq i}^{n} |a_{ij}| \equiv \delta_i > 0, \text{ for } i = 1, \ldots, n.$$ 

Consider such a matrix and define $\delta_{\text{min}} = \min_{i=1,\ldots,n} \delta_i$.

(a) Show that for any vector $y$ we have $\|Ay\|_\infty \geq \delta_{\text{min}}\|y\|_\infty$.

(b) Deduce from (a) that $A$ is nonsingular.

(c) Show that $\|A^{-1}\|_\infty \leq \frac{1}{\delta_{\text{min}}}$.

(d) [5 Bonus pts] Show that when Gaussian elimination with partial pivoting is applied to a strictly column diagonally dominant ($A^T$ is row-diagonal dominant) then rows are never permuted. [Hint: Show (by induction) that the matrices $A_k$ obtained at each step of GE remain column Diag. Dom.]

6. Consider the following two algorithms to compute the function $f(x) = (e^x-1)/x = \sum_{i=0}^{\infty} x^i/(i+1)!$ which arises in many applications:

```
Algorithm 1
if x == 0
    f = 1
else
    f = (exp(x)-1)/x;
end
```

```
Algorithm 2
y = exp(x)
if y == 1
    f = 1;
else
    f = (y-1)/log(y);
end
```

Using matlab, tabulate results of both algorithms for $x = 10^{-5}, 10^{-6}, 10^{-7}, \cdots, 10^{-16}$. In a third column show also the values of $1 + x/2 + x^2/6$ which are obtained from the Taylor series expansion (second order accurate). Use ‘format long’ to display your results. Discuss and explain what you observe [Hint: in first algorithm you need to show that a cancellation error is amplified. For second algorithm note that the first step computes $f(y) = e^x(1 + \delta)$ which you can write as $\hat{y} = e^\hat{x}$. How accurate is the rest of the computation? What is $x - \hat{x}$ approximately?]