**Doolittle’s method**: The Doolittle method first factors an nxn matrix **A** into **LU** where **L** is a lower triangular matrix with 1’s on the diagonal and **U** is an upper triangular matrix. The solution to the matrix equation **Ax**=**b** is found by, **Ly**=**b →** solved for **y → Ux=y →** solve for **x**. You are to write a program that solves the following set of equations using the Doolittle method.

*Part 1*: Create a function def LUFactorization(A): that takes **A** as an argument and returns a tuple containing (**L**, **U**).

To accomplish the **LU** factorization, we can use the following steps where **L**[j][k]=mjk and **U**[j][k]=ujk:

Step 1: Create two matrices **L** & **U** that are the same shape as **A** with:

1’s on the diagonal of **L**

The first row of **U** equal to first row of **A.**

The first column of **L** equal to the first column of **A**/**U**[0][0] except for **L**[0][0]=1

**Note**: You must use list comprehensions to accomplish step 1 in two lines of code

Step 2(a): Set where, j≥1 and k=j,…,n-1

Step 2(b): For each k≥ 1 in 5(a), set where, i=k+1,…,n-1

*Part 2*: Create a function def Doolittle(Aaug): that takes the augmented matrix from the equations above as an argument and returns the solution vector x as a list.

*Part 3:* Create a function def main(): that formulates the matrix equation, calls Doolittle and prints the solution vector to the screen along with a calculation of **b** using matrix multiplication **Ax**.

**Notes:**

1. You may not use numpy or any other modules except: *(i)* the ones we developed in class or previous homework, *(ii)* math, copy, random
2. Your output should use formatted floating point numbers with 3 to 4 decimal places of precision.
3. You must document your code using docstrings and comments.

The Doolittle method is discussed in §20.2 of the MAE 3013 text.