Problem 1. Permutation Matrices

Create a permutation $P$ matrix that takes the vector $x = [0, 1, 2, 3, 4]^T$ to $Px = [1, 3, 4, 0, 2]$.

```python
import numpy as np
P = np.zeros((5,5))

P[:,0] = 1
P[:,1] = 1
P[:,2] = 1
P[:,3] = 1
P[:,4] = 1

print(P.dot(x))
```
Problem 2. Pivoted LU

Factor the matrix

\[
A = \begin{bmatrix}
0 & 2 & 1 \\
1 & 1 & 3 \\
2 & 4 & 4
\end{bmatrix}
\]

into a permutation matrix \(P\), a lower triangular matrix \(L\), and an upper triangular matrix \(U\).

Here are a few reminders about the process (so that you don’t have to go look these up):

- Original factorization: \(M_2P_2M_1P_1U = A\)
- \(L_2 = M_2\)
- \(L_1 = P_2M_1P_2^{-1}\)
- \(L = L_2^{-1}L_2^{-1}\)
- \(P = P_2P_1\)

```python
import numpy as np

P = np.zeros((3,3), dtype=np.float64)
P[:, 0] = 1
P[:, 1] = 1
P[:, 2] = 1

L = np.array([
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1],
])

U = np.array([
    [0, 0, 0],
    [0, 0, 1],
    [0, 0, 0],
])

print(P.dot(A)-L.dot(U))
```