Name: _____ **ESMI Applied Math**
Due: 06/14 **Worksheet 6**

Problem 1. Write a python function that takes in two numpy arrays (matrices) as inputs and returns the matrix-matrix product if possible. If the matrix-matrix product is not possible, your function should print "these matrices are not compatible for matrix-matrix multiplication". Code this up first in jupyter notebooks and once you are happy with your answer, please write your function below.

```python
# Solution to problem 1 - matrix multiplication from scratch

def matmul(mat1, mat2):
    n = np.shape(mat1)[0]
    m = np.shape(mat1)[1]
    m2 = np.shape(mat2)[0]
    p = np.shape(mat2)[1]

    #check if matrices are compatible for matrix-matrix multiplication
    if m != m2:
        print("these matrices are not compatible for matrix-matrix multiplication")
        return

    out = np.zeros((n,p))

    #compute matrix-matrix multiplication
    for i in range(n):
        for j in range(p):
            #compute dot product
            s = 0
            for k in range(m):
                s = s + mat1[i][k]*mat2[k][j]

            out[i][j] = s

    return out
```

Problem 2. Write a function that takes in an adjacency list (in the form of a python dictionary) of a network and outputs the adjacency matrix of that network. Write your pseudocode first, code it up, then include both the pseudocode and the actual code below.

```
# Write a function that takes in an adjacency list of a network and outputs the adjacency matrix

def AdjListToMat(adjList):
    defList = {}
    i = 0
    #give all elements of the adjacency list a number label
    for item in adjList:
        defList[item] = i
        i = i + 1

    #initialize adjacency matrix
    A = np.zeros((i, i))

    #loop through each edge and fill in the edge weight into the appropriate spot in A
    for item in adjList:
        for destinations in adjList[item]:
            idx1 = defList[item]
            idx2 = defList[destinations[0]]
            A[idx1][idx2] = destinations[1]

    return A
```