ANA-522-OL1 Spring 2022
Mod02 Week03 Lab: NumPy
Due: Friday January 28th at midnight

1 Create a null vector (zeros) of size 40

(a) create in one dimension, as array A1

(b) create in two dimension with 8 rows and 5 columns, as array A2

(c) create a new array as A3, which is a copy of A2, except that all values on the third column are 1
Print A1, A2, and A3.

```
Array A1:
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Array A2:
 [[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
Array A3:
 [[0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0.]]
```

2 2. Create a vector with values ranging from 1 to 40

(a) create in one dimension, as array B1

(b) create in two dimenion with 8 rows and 5 columns, as array B2

(c) create a new array B3 by adding B2 with A3 (,or increase all values of fourth column of B2 by 1)

(d) create a new array B4 with the values from B3 subtracted by 11
Print B1, B2, B3, and B4

```
Array B1:
 [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]
Array B2:
 [[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]
 [26 27 28 29 30]
 [31 32 33 34 35]
 [36 37 38 39 40]]
Array B3:
 [[ 1.  2.  4.  4.  5.]
 [ 6.  7.  9.  9. 10.]
 [11. 12. 14. 14. 15.]
 [16. 17. 19. 19. 20.]
 [21. 22. 24. 24. 25.]
 [26. 27. 29. 29. 30.]
 [31. 32. 34. 34. 35.]
 [36. 37. 39. 39. 40.]]
Array B4:
 [[-10.  -9.  -7.  -7.  -6.]
 [ -5.  -4.  -2.  -2.  -1.]
 [  0.   1.   3.   3.   4.]
 [  5.   6.   8.   8.   9.]
 [ 10.  11.  13.  13.  14.]
 [ 15.  16.  18.  18.  19.]
 [ 20.  21.  23.  23.  24.]
 [ 25.  26.  28.  28.  29.]]
```

3 3. Create a 12x12 matrix and fill it with a checkerboard pattern

(a) first row and first column starts from 0

(b) first row and first column starts from 1

(c) with all numbers are 1 (integer)

(d) with all numbers are 1.0 (floating-point)
Print all matrix above.

```
Checkerboard pattern starts from 0:
[[0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]]

Checkerboard pattern starts from 1:
[[1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1 0 1 0 1]]
```

```
All integer ones:
[[1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 1]]

All floating-point ones:
[[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

4 4. Create 7x7 matrixes

(a) P matrix with row values ranging from 0 to 6

(b) Q matrix with column values ranging from 0 to 6

(c) R matrix with cumulative sum from previous row of matrix Q

(d) S matrix with cumulative sum from previous column of matrix Q

(e) T list with cumulative sun from previous value rolled out in order of row and column from matrix Q
Print matrix P,Q,R,S, and list T

```
Matrix P:
[[0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]
 [0. 1. 2. 3. 4. 5. 6.]]
Matrix Q:
[[0. 0. 0. 0. 0. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1.]
 [2. 2. 2. 2. 2. 2. 2.]
 [3. 3. 3. 3. 3. 3. 3.]
 [4. 4. 4. 4. 4. 4. 4.]
 [5. 5. 5. 5. 5. 5. 5.]
 [6. 6. 6. 6. 6. 6. 6.]]
Matrix R:
[[ 0.  0.  0.  0.  0.  0.  0.]
 [ 1.  1.  1.  1.  1.  1.  1.]
 [ 3.  3.  3.  3.  3.  3.  3.]
 [ 6.  6.  6.  6.  6.  6.  6.]
 [10. 10. 10. 10. 10. 10. 10.]
 [15. 15. 15. 15. 15. 15. 15.]
 [21. 21. 21. 21. 21. 21. 21.]]
Matrix S:
[[ 0.  0.  0.  0.  0.  0.  0.]
 [ 1.  2.  3.  4.  5.  6.  7.]
 [ 2.  4.  6.  8. 10. 12. 14.]
 [ 3.  6.  9. 12. 15. 18. 21.]
 [ 4.  8. 12. 16. 20. 24. 28.]
 [ 5. 10. 15. 20. 25. 30. 35.]
 [ 6. 12. 18. 24. 30. 36. 42.]]
List T:
[  0.   0.   0.   0.   0.   0.   0.   1.   2.   3.   4.   5.   6.   7.
   9.  11.  13.  15.  17.  19.  21.  24.  27.  30.  33.  36.  39.  42.
  46.  50.  54.  58.  62.  66.  70.  75.  80.  85.  90.  95. 100. 105.
 111. 117. 123. 129. 135. 141. 147.]
```

5 5. Create a matrix, block segments, and integrations

(a) create matrix X of 3 rows and 12 columns of random integers range [0,100]

(b) create matrix XA of 3 rows and 3 columns of subset matrix of X with all rows, and the first three columns (0 to 2)

(c) create matrix XB of 3 rows and 3 columns of subset matrix of X with all rows, and the second quarter columns (3 to 5)

(d) create matrix XC of 3 rows and 3 columns of subset matrix of X with all rows, and tbe third quarter columns (6 to 8)

(e) create matrix XD of 3 rows and 3 columns of subset matrix of X with all rows, and tbe fourth quarter columns (9 to 11)

(f) create matrix Y of 3 rows and 12 columns by putting together in the order of XD XA XB XC from column index 0 to 11

(g) create matrix Z of 12 rows and 3 columns by putting together vertically in the order of XD XA XB XC

```
Matrix X:
[[ 3 54 99 72 41 64 54 48 57 15 51 73]
 [59 28 95 93 75 88 21 79 98 83 75 61]
 [ 0 70 54 40 37 38 68 67 90 58 46 58]]
Matrix XA:
[[ 3 54 99]
 [59 28 95]
 [ 0 70 54]]
Matrix XB:
[[72 41 64]
 [93 75 88]
 [40 37 38]]
Matrix XC:
[[54 48 57]
 [21 79 98]
 [68 67 90]]
Matrix XD:
[[15 51 73]
 [83 75 61]
 [58 46 58]]
Matrix Y:
[[15 51 73  3 54 99 72 41 64 54 48 57]
 [83 75 61 59 28 95 93 75 88 21 79 98]
 [58 46 58  0 70 54 40 37 38 68 67 90]]
```

```
Matrix Z:
[[15 51 73]
 [83 75 61]
 [58 46 58]
 [ 3 54 99]
 [59 28 95]
 [ 0 70 54]
 [72 41 64]
 [93 75 88]
 [40 37 38]
 [54 48 57]
 [21 79 98]
 [68 67 90]]
```

Hint: Use np.concatenate() to put two matrixes together (avaible from reading assignments.)

6 6. Create the following two matrixes in the program

and do the matrix multiplication where C = AB
Print matrixes A, B, and C.

```
Matrix A:
[[1 2]
 [3 4]
 [5 6]
 [7 8]]
Matrix B:
[[ 9 10 11 12]
 [13 14 15 16]]
Matrix C:
[[ 35  38  41  44]
 [ 79  86  93 100]
 [123 134 145 156]
 [167 182 197 212]]
```