Exercise

Attempt the following exercises. When you see an output, **DO NOT WRITE IN THAT CELL**, A cell will be provided **below the output cell** where you can run your code.

IF YOU RESTART AND RUN ALL CELLS, THE OUTPUT WILL DISAPPEAR

Question 1

Create an **1-d array** from a list of **15 elements** and attribute it to a variable name **array_values**

```
In [3]: import numpy as np
        array_values = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
        one_d_array = np.array(array_values)
        one_d_array
```

```
Out[3]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

Question 2

Calculate the **mean** and **standard deviation** of the above array

```
In [5]: np.mean([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
```

```
Out[5]: 8.0
```

```
In [6]: np.std([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
```

```
Out[6]: 4.320493798938574
```

Question 3

Select out values that are greater than the mean value of the **array_values** and attribute it to a variable name **above_mean**

```
In [60]: above_mean = array_values [8:]
         array_values = np.array(above_mean)
         array_values

Out[60]: array([ 9, 10, 11, 12, 13, 14, 15])
```

```
In [ ]:
```

Question 4

Create a **2-d array** from a list with **four rows and three columns** and attribute it to a variable **array_2d**

```
In [30]: array_2d = ([1,2,3],[4,5,6],[7,8,9],[10,11,12])
         matrix = np.array(array_2d)
         matrix

Out[30]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])
```

Question 5

Select the **second element from the fourth row** of the **array_2d** variable

```
In [35]: matrix[3, 1]

Out[35]: 11
```

Question 6

Convert the **array_2d** variable to a **1-d array** using the appropriate numpy function

```
In [41]: nest

Out[41]: ([1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12])
```

Question 7

Generate 10 linearly spaced numbers between 15 and 17.

```
In [2]: # Your output should look like this

Out[2]: array([15.        , 15.22222222, 15.44444444, 15.66666667, 15.88888889,
               16.11111111, 16.33333333, 16.55555556, 16.77777778, 17.        ])
```

```
In [42]: np.linspace(15,17,10)

Out[42]: array([15.        , 15.22222222, 15.44444444, 15.66666667, 15.88888889,
                16.11111111, 16.33333333, 16.55555556, 16.77777778, 17.        ])
```

Question 8

Using a seed of 5, generate a dimension of random numbers with 4 rows and 5 columns

```
In [3]: # The output should look like this

Out[3]: array([[0.22199317, 0.87073231, 0.20671916, 0.91861091, 0.48841119],
               [0.61174386, 0.76590786, 0.51841799, 0.2968005 , 0.18772123],
               [0.08074127, 0.7384403 , 0.44130922, 0.15830987, 0.87993703],
               [0.27408646, 0.41423502, 0.29607993, 0.62878791, 0.57983781]])
```

```
In [49]: import numpy.random as  random
         random.seed (5)
         random.rand(4,1,5)
```

```
Out[49]: array([[[0.22199317, 0.87073231, 0.20671916, 0.91861091, 0.48841119]],

                [[0.61174386, 0.76590786, 0.51841799, 0.2968005 , 0.18772123]],

                [[0.08074127, 0.7384403 , 0.44130922, 0.15830987, 0.87993703]],

                [[0.27408646, 0.41423502, 0.29607993, 0.62878791, 0.57983781]]])
```

Question 9

Create an **identity matrix** containing **6 rows and 7 columns** and attribute it to a variable **identity**

```
In [4]: # Your output should look like this
```

```
Out[4]: array([[1., 0., 0., 0., 0., 0., 0.],
               [0., 1., 0., 0., 0., 0., 0.],
               [0., 0., 1., 0., 0., 0., 0.],
               [0., 0., 0., 1., 0., 0., 0.],
               [0., 0., 0., 0., 1., 0., 0.],
               [0., 0., 0., 0., 0., 1., 0.]])
```

```
In [57]: identity = np.eye(6,7)
         identity_matrix = np.array(identity)
         identity_matrix
```

```
Out[57]: array([[1., 0., 0., 0., 0., 0., 0.],
                [0., 1., 0., 0., 0., 0., 0.],
                [0., 0., 1., 0., 0., 0., 0.],
                [0., 0., 0., 1., 0., 0., 0.],
                [0., 0., 0., 0., 1., 0., 0.],
                [0., 0., 0., 0., 0., 1., 0.]])
```

Question 10

Generate **50 random integers between 1 and the length of array_values**. Calculate the:

- mean
- sum
- maximum and minimum with their respective indexes.

In [ ]:

Good Luck!!!