Assignment - 3
Machine Learning (CS 5710) CRN: 22002

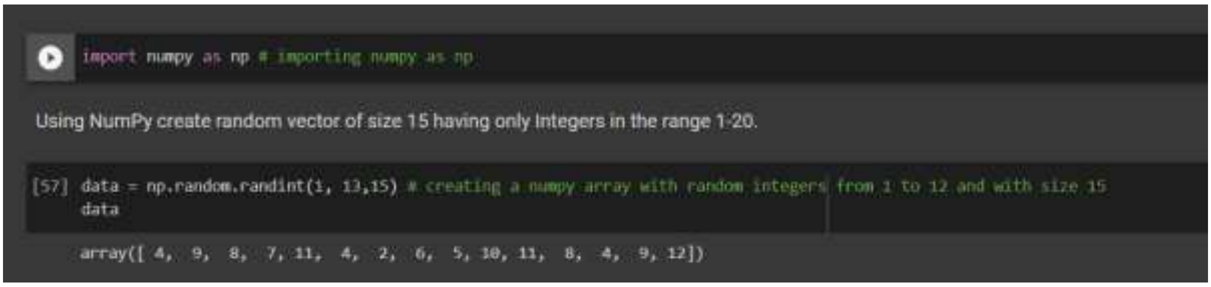
Vinay Kumar Camarushi
700740428

GitHub Link: https://github.com/VXC04280/In_Class_Programming_Assignment_3

Question – a)
Using NumPy create random vector of size 15 having only Integers in the range 1-20.

Solution :

- creating a numpy array with random integers from 1 to 12 and with size 15



```
import numpy as np # importing numpy as np
```

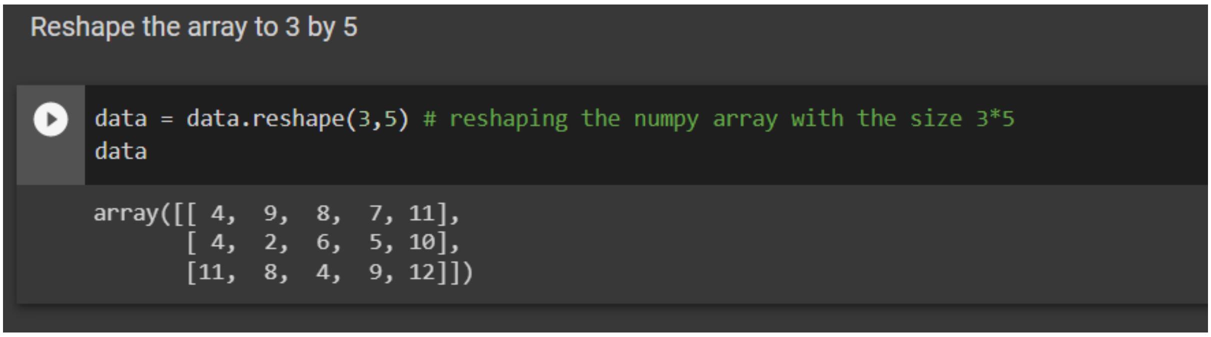
Using NumPy create random vector of size 15 having only Integers in the range 1-20.

```
[57] data = np.random.randint(1, 13,15) # creating a numpy array with random integers from 1 to 12 and with size 15
     data

     array([ 4,  9,  8,  7, 11,  4,  2,  6,  5, 10, 11,  8,  4,  9, 12])
```

Question – a1)

Reshape the array to 3 by 5

Sol: reshaping the numpy array with the size 3*5



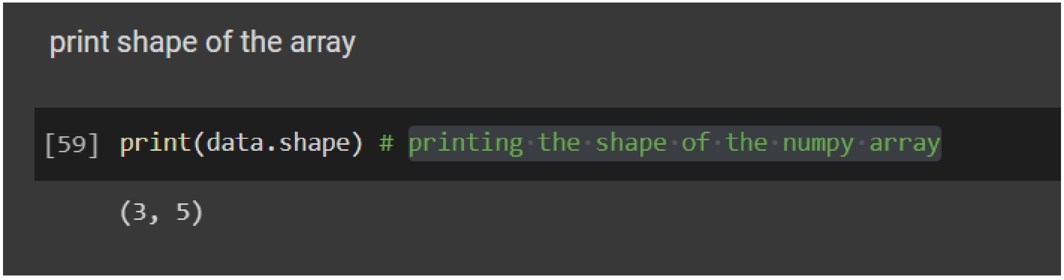
Reshape the array to 3 by 5

```
data = data.reshape(3,5) # reshaping the numpy array with the size 3*5
data

array([[ 4,  9,  8,  7, 11],
       [ 4,  2,  6,  5, 10],
       [11,  8,  4,  9, 12]])
```

Question – a2)

print shape of the array

- printing the shape of the numpy array using shape function.



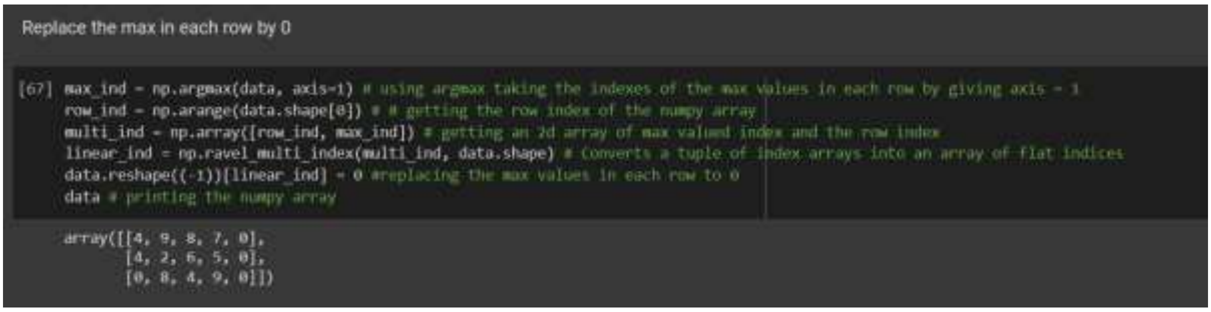
print shape of the array

```
[59] print(data.shape) # printing the shape of the numpy array

     (3, 5)
```

Question – a3)

Replace the max in each row by 0

- using argmax taking the indexes of the max values in each row by giving axis = 1 in line 1
- getting the row index of the numpy array in line 2
- getting an 2d array of max valued index and the row index in line 3
- ravel_multi_index Converts a tuple of index arrays into an array of flat indices in line 4
- replacing the max values in each row to 0 in line 5 using the assignment operator

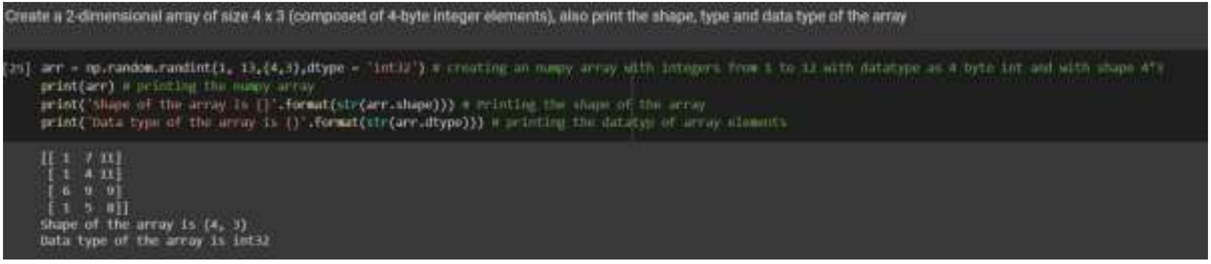


Replace the max in each row by 0

```
[67] max_ind = np.argmax(data, axis=1) # using argmax taking the indexes of the max values in each row by giving axis = 1
     row_ind = np.arange(data.shape[0]) # # getting the row index of the numpy array
     multi_ind = np.array([row_ind, max_ind]) # getting an 2d array of max valued index and the row index
     linear_ind = np.ravel_multi_index(multi_ind, data.shape) # Converts a tuple of index arrays into an array of flat indices
     data.reshape((-1))[linear_ind] = 0 #replacing the max values in each row to 0
     data # printing the numpy array

     array([[4, 9, 8, 7, 0],
            [4, 2, 6, 5, 0],
            [0, 8, 4, 9, 0]])
```

Question – 2)
Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array.

- creating an numpy array with integers from 1 to 12 with datatype as 4 byte int and with shape 4*3
- And printing the shape of the array along with its datatype



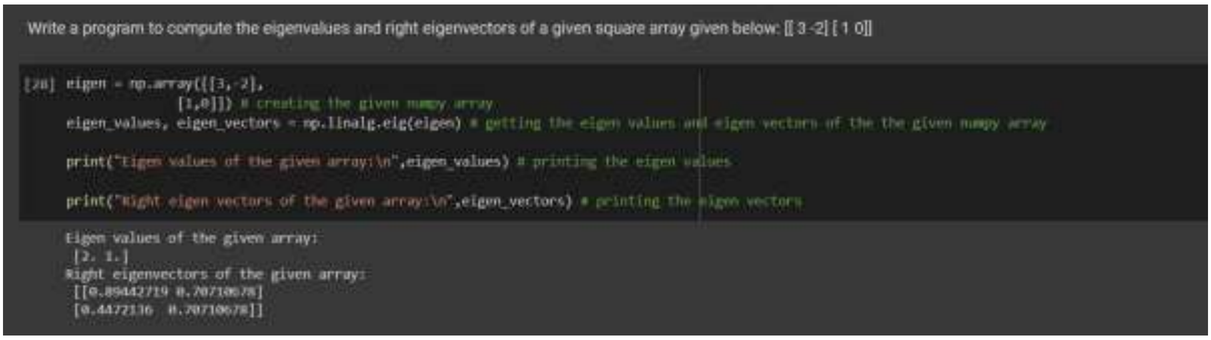
Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array

```
[25] arr = np.random.randint(1, 13,(4,3),dtype = 'int32') # creating an numpy array with integers from 1 to 12 with datatype as 4 byte int and with shape 4*3
     print(arr) # printing the numpy array
     print('Shape of the array is {}'.format(str(arr.shape))) # printing the shape of the array
     print('Data type of the array is {}'.format(str(arr.dtype))) # printing the datatyp of array elements
```

```
[[ 1  7 11]
 [ 1  4 11]
 [ 6  9  9]
 [ 1  5  8]]
Shape of the array is (4, 3)
Data type of the array is int32
```

Question – 2a)
Write a program to compute the eigenvalues and right eigenvectors of a given square array given below: [[3 -2] [1 0]]

- creating the given numpy array
- getting the eigen values and eigen vectors of the the given numpy array using the linear algebra method.
- Printing the eigen values and eigen vectors of the given numpy array.
-



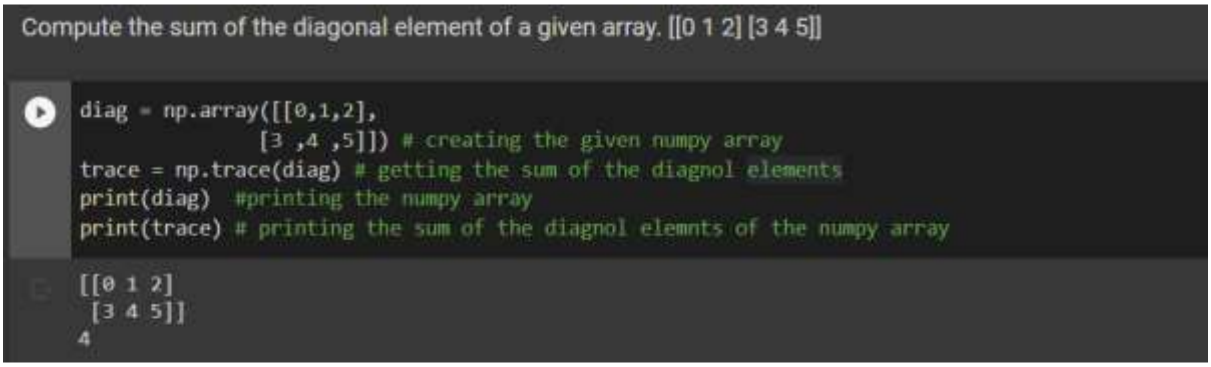
Write a program to compute the eigenvalues and right eigenvectors of a given square array given below: [[3 -2] [1 0]]

```
[28] eigen = np.array([[3,-2],
                [1,0]]) # creating the given numpy array
     eigen_values, eigen_vectors = np.linalg.eig(eigen) # getting the eigen values and eigen vectors of the the given numpy array

     print("Eigen values of the given array:\n",eigen_values) # printing the eigen values

     print("Right eigen vectors of the given array:\n",eigen_vectors) # printing the eigen vectors
```

```
Eigen values of the given array:
 [2. 1.]
Right eigenvectors of the given array:
 [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

Question – 2b)

- **Compute the sum of the diagonal element of a given array. [[0 1 2] [3 4 5]]**
- creating the given numpy array
- getting the sum of the diagnol elements using the trace method.
- Printing the trace of the given numpy array.



Compute the sum of the diagonal element of a given array. [[0 1 2] [3 4 5]]

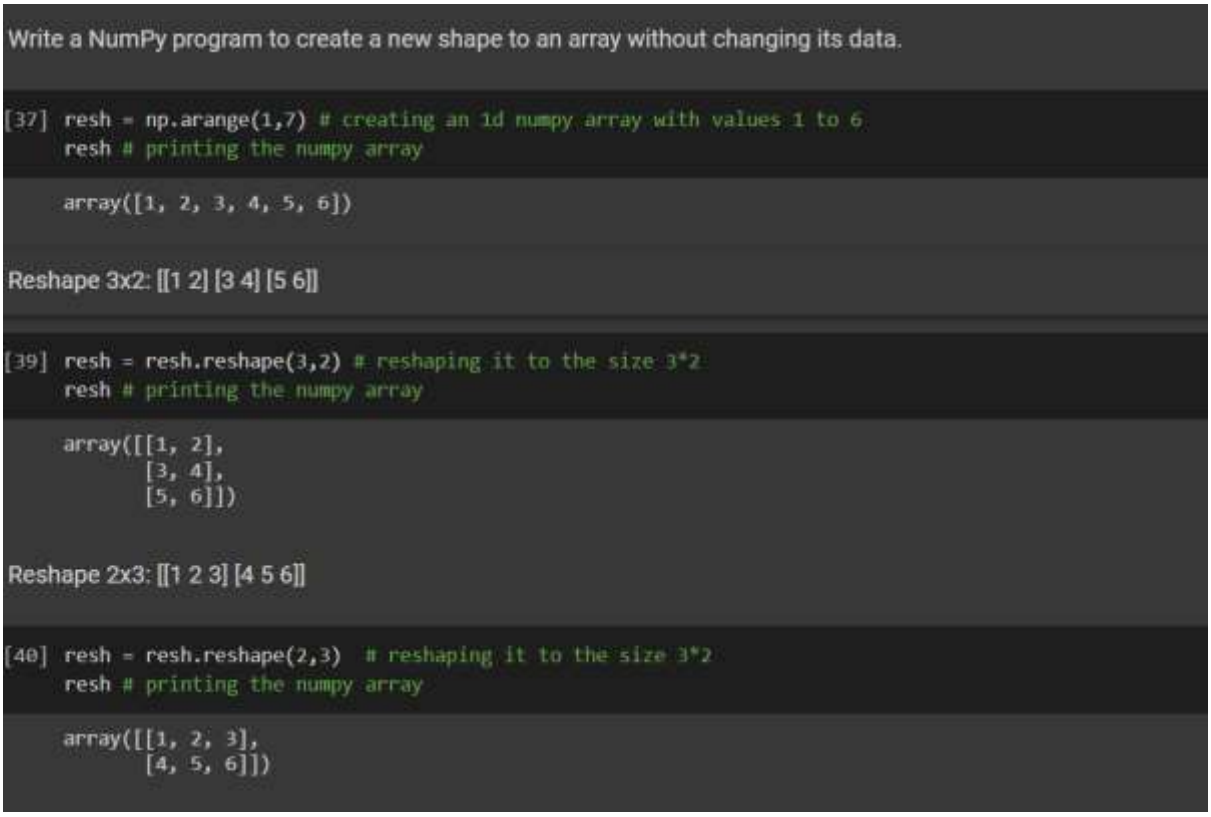
```
diag = np.array([[0,1,2],
                 [3 ,4 ,5]]) # creating the given numpy array
trace = np.trace(diag) # getting the sum of the diagnol elements
print(diag)  #printing the numpy array
print(trace) # printing the sum of the diagnol elemnts of the numpy array
```

```
[[0 1 2]
 [3 4 5]]
4
```

Question – 2c)

Write a NumPy program to create a new shape to an array without changing its data.

- creating the numpy array using np.arange method
- Printing the array after reshaping it to 3*2 and 2*3 arrays without changing its data with the help of reshape method.



Write a NumPy program to create a new shape to an array without changing its data.

```
[37] resh = np.arange(1,7) # creating an 1d numpy array with values 1 to 6
     resh # printing the numpy array

     array([1, 2, 3, 4, 5, 6])
```

Reshape 3x2: [[1 2] [3 4] [5 6]]

```
[39] resh = resh.reshape(3,2) # reshaping it to the size 3*2
     resh # printing the numpy array

     array([[1, 2],
            [3, 4],
            [5, 6]])
```

Reshape 2x3: [[1 2 3] [4 5 6]]

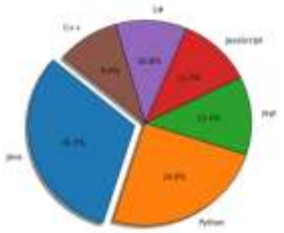
```
[40] resh = resh.reshape(2,3)  # reshaping it to the size 3*2
     resh # printing the numpy array

     array([[1, 2, 3],
            [4, 5, 6]])
```

Question – 2c)

Write a Python programming to create a below chart of the popularity of programming Languages.

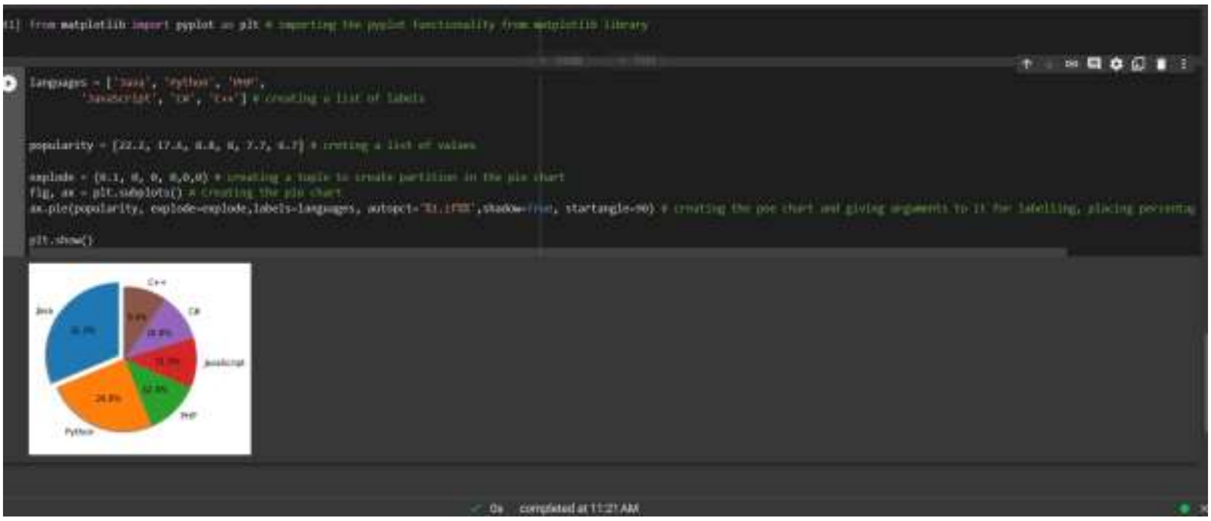
Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++

Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7



Solution :

- creating a list of labels
- creating a list of values
- creating a tuple to create partition in the pie chart
- Creating the pie chart using subplot methods and giving arguments to it for labelling, placing percentage, slicing it and giving the shadow.



```
[1] from matplotlib import pyplot as plt # importing the pyplot functionality from matplotlib library

languages = ['Java', 'Python', 'PHP',
        'JavaScript', 'C#', 'C++'] # creating a list of labels


popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7] # creting a list of values

explode = (0.1, 0, 0, 0,0,0) # creating a tuple to create partition in the pie chart
fig, ax = plt.subplots() # creating the pie chart
ax.pie(popularity, explode=explode,labels=languages, autopct='%1.1f%%',shadow=True, startangle=90) # creating the pie chart and giving arguments to it for labelling, placing percentag

plt.show()
```

✓ 0s completed at 11:21 AM