Submitted by: Mounika Prathapani (700745641)

Video link: https://drive.google.com/file/d/1bFXifa9vblnAgxwt1wFDPzWf-BQ7FdSE/view?usp=drive_link

GitHub link: https://github.com/mounikaprat/CS5710_Assignment1

1) Numpy

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

```
import numpy as np
```

```
random_vector = np.random.randint(1,21,size = 15)
```

```
print(random_vector)
```

```
[ 7 10  1 15 14  9 16  1  9  4 15 11 20  2 12]
```

1. Reshape the array to 3 by 5

```
reshape_vector = random_vector.reshape(3,5)
```

```
print(reshape_vector)
```

```
[[ 7 10  1 15 14]
 [ 9 16  1  9  4]
 [15 11 20  2 12]]
```

2. Print array shape

```
print("shape of array :", reshape_vector.shape)
```

```
shape of array : (3, 5)
```

3. Replace the max in each row by 0

```
ma = np.amax(reshape_vector,axis = 1)
replace_vector = np.where(np.isin(reshape_vector,ma),0,reshape_vector)
print(replace_vector)
```

```
[[ 7 10  1  0 14]
 [ 9  0  1  9  4]
 [ 0 11  0  2 12]]
```

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.

```
new_vector = np.array([[4,3,2],[1,5,9],[6,7,8],[9,2,4]],np.int32)
print(new_vector)
```

```
[[4 3 2]
 [1 5 9]
 [6 7 8]
 [9 2 4]]
```

```
print(new_vector.shape)
print(type(new_vector))
print(new_vector.dtype)
```

```
(4, 3)
<class 'numpy.ndarray'>
int32
```

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

[1 0]]

```
# importing numpy library
import numpy as np
#creating numpy 2d-array
arr =np.array([[3,-2],[1,0]])
#finding eigen values and vectors
eigen_values, eigen_vectors = np.linalg.eig(arr)
#printing eigen values
print("eigen values of the given square array is :",eigen_values)
#printing eigen vectors
print("eigen values of the given square array is :",eigen_vectors)
```

```
eigen values of the given square array is : [2. 1.]
eigen values of the given square array is : [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

c. Compute the sum of the diagonal element of a given array. [[0 1 2]

[3 4 5]]

```python
# importing numpy library
import numpy as np

# creating numpy array
arr = np.array([[0,1,2],[3,4,5]])
diagonal_sum = 0

# calcualting diagonal sum
for i in range(len(arr)):
  for j in range(len(arr[i])):
    if i == j:
      diagonal_sum+=arr[i][j]
print(diagonal_sum)
```

```
4
```

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

Reshape 3x2: [[1 2]

[3 4]

[5 6]]

Reshape 2x3: [[1 2 3]

[4 5 6]]

```
[ ]  import numpy as np
     # creating numpy array
     arr = np.array([1,2,3,4,5,6])

     # reshaping the array to 3X2
     arr = np.reshape(arr,(3,2))
     print("3X2 Reshape :\n",arr )
     print("\n")

     # reshaping the array to 2X3
     arr = np.reshape(arr,(2,3))
     print("2X3 Reshape :\n",arr)
```

```
3X2 Reshape :
 [[1 2]
 [3 4]
 [5 6]]


2X3 Reshape :
 [[1 2 3]
 [4 5 6]]
```

2. Matplotlib

1. Write a Python programming to create a below chart of the popularity of programming Languages.
2. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

```
from matplotlib import pyplot as plt
#Data to plot
languages = 'Java','Python','PHP','JavaScript','C#','C++'
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
colors = ["#1f77b4","#ff7f0e","#2ca02c","#d62728","#9467bd","#8c564b"]
# exploding first slice
explode = (0.1,0,0,0,0,0)
#plot
plt.pie(popularity,explode=explode, labels=languages,colors = colors,autopct = '%1.1f%%',shadow=True, startangle = 140)
plt.axis('equal')
plt.show()
```