Github link: https://github.com/KAR2908/Assignment1

Video Link : https://drive.google.com/file/d/1uA33SnQwsGFjdA8OZcQQraR-rvbynE0C/view?usp=drive_link

Numpy

```
import numpy as np
```

Creating random vector of size 15 with integers in range 1-20

```
k1 = np.random.randint(1, 21, size=15)
print(k1)
```

```
[19 13  6 16 13 11 18  3  8  9  7 17 16  1  8]
```

Reshaping the array size to 3 by 5

```
k2 = k1.reshape(3,5)
print(k2)
```

```
[[19 13  6 16 13]
 [11 18  3  8  9]
 [ 7 17 16  1  8]]
```

printing the array shape

```
print(k2.shape)
```

```
(3, 5)
```

Replace the max in each row by 0

```
max_indices = np.argmax(k2, axis=1)
k2[np.arange(len(k2)),max_indices] = 0
print(k2)
```

```
[[ 0 13  6 16 13]
 [11  0  3  8  9]
 [ 7  0 16  1  8]]
```

Creating 2D array of size 4x3 with 4-byte integer elements and also print the shape, type and datatype of array

```
#creating the array
sk = np.zeros((4,3),dtype=np.int32)
#printt the shape, type and datatype of array
print("shape:",sk.shape)
print("type:", type(sk))
print("data type:", sk.dtype)
```

```
shape: (4, 3)
type: <class 'numpy.ndarray'>
data type: int32
```

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

```
#Array declaration
s1 =  np.array([[3, -2], [1, 0]])

#compute eigen values and eigenvectors
eigenval, eigenvect = np.linalg.eig(s1)

#printing the results:
print("Eigenvalues are:", eigenval)
print("Right eigenvectors are: ")
for i, eigenvect in enumerate(eigenvect.T):
    print(f"Eigenvector {i+1}:", eigenvect)
```

```
Eigenvalues are: [2. 1.]
Right eigenvectors are: 
Eigenvector 1: [0.89442719 0.4472136 ]
Eigenvector 2: [0.70710678 0.70710678]
```

Sum of the diagonal elements of a given array:[[0 1 2][3 4 5]]

```
#Array declaration
s2 = np.array([[0, 1, 2], [3, 4, 5]])
#sum of diagonal elements
diag_sum = np.trace(s2)
#print the sum
print("sum of diagonal elements is", diag_sum)
```

```
sum of diagonal elements is 4
```

program to create new shape to array without changing its data. Reshape 3x2 :[[1 2][3 4][5 6]]

```
#array declaration
array3= np.array([[1,2,3],[4,5,6]])
reshaped_array = array3.reshape((3, 2))
print("Reshaped 3x2 array:\n ")
print (reshaped_array)
```

```
Reshaped 3x2 array:
 
[[1 2]
 [3 4]
 [5 6]]
```

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

```
#Reshape to 2X3
reshape2= reshaped_array.reshape((2, 3))
print("Reshaped 2X3 array:\n ")
print (reshape2)
```

```
Reshaped 2X3 array:
 
[[1 2 3]
 [4 5 6]]
```

Matplotlib

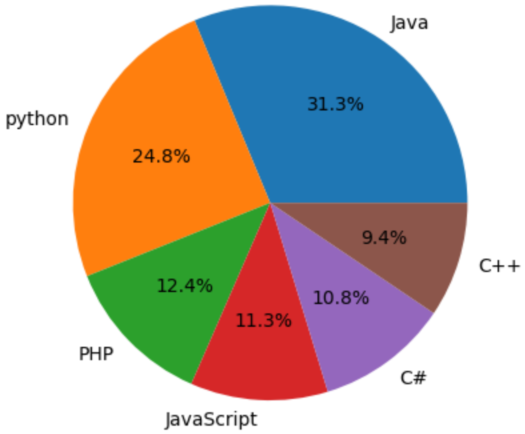
1. Write a Python programming to create a below chart of the popularity of programming Languages.
2. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

```
import matplotlib.pyplot as plt
#Given sample data

langs = ['Java','python','PHP','JavaScript','C#','C++' ]
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

# Creating a pie chart
plt.pie(popularity, labels=langs, autopct='%1.1f%%')


#To show the plot
plt.show()
```


Java
31.3%
python
24.8%
9.4%
C++
12.4%
10.8%
11.3%
PHP
C#
JavaScript