

# **Introduction to numpy library in Python**

## **Lab report #06**



Fall 2022

CSE-408L Data Analytics lab

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_

Submitted to:

**Eng: Faizullah**

**Dec 22, 2022**

**Department of Computer Systems Engineering**  
**University of Engineering and Technology, Peshawar**

# lab06

December 22, 2022

## 1 Lab Tasks

**Task01** Replace all odd numbers in given array with -1.

```
task01=np.array([0,1,2,3,4,5,6,7,8,9])
```

```
[1]: import numpy as np
```

```
[2]: task01=np.array([0,1,2,3,4,5,6,7,8,9])
```

```
[3]: task01
```

```
[3]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
[4]: for i in task01:
      if i%2!=0:
          task01[i]=-1
```

```
[5]: task01
```

```
[5]: array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

### 1.0.1 Task02

Find the positions of 1) elements in x where its value is more than its corresponding element in y.  
2)elements in x where its value is equal to its corresponding element in y.

```
x=np.array([21,64,86,22,74,55,81,66,89])
```

```
y=np.array([21,7,3,45,10,29,55,44,89])
```

```
[6]: x=np.array([21,64,86,22,74,55,81,66,89])
```

```
[7]: x
```

```
[7]: array([21, 64, 86, 22, 74, 55, 81, 66, 89])
```

```
[8]: y=np.array([21,7,3,45,10,29,55,44,89])
```

```
[9]: y
```

```
[9]: array([21,  7,  3, 45, 10, 29, 55, 44, 89])
```

positions of elements in x where its value is more than its corresponding element in y.

```
[10]: for i in range(len(x)):
        if x[i]>y[i]:
            print(i,end=',')
```

1,2,4,5,6,7,

positions of elements in x where its value is equal to its corresponding element in y.

```
[11]: for i in range(len(x)):
        if x[i]==y[i]:
            print(i,end=',')
```

0,8,

### 1.0.2 Task03

Generate a 2D array of random numbers of size (1000,3). find the mean of 3 random numbers and store the mean of each row in fourth column of that row. save result in csv file by the name mean.csv on your local disk with headers num1, num2 num3 and avervage. also find which row has the highest mean of all.

```
[12]: arr2d=np.random.rand(1000,3)
```

```
[13]: arr2d
```

```
[13]: array([[0.39323667, 0.6394314 , 0.36252106],
             [0.55616985, 0.65306829, 0.884103  ],
             [0.42144908, 0.92911412, 0.96655347],
             ...,
             [0.373153  , 0.46836664, 0.26040695],
             [0.86446074, 0.82746778, 0.71771681],
             [0.69594843, 0.44118409, 0.64880824]])
```

each list inside array is the item of an array

```
[14]: mean_row=[]
```

```
[15]: for row in arr2d:    #row is not a keyword. we can also use any other variable.
        m=np.mean(row)
        mean_row.append(m)
```

```
[16]: darray=np.array(mean_row)    #convert to numpy array
```

```
[17]: n=darray.reshape(1000,1)
```

```
[18]: new_array=np.concatenate((arr2d,n),axis=1)    #concatinate two arrays to
        ↪specific axis. 0=row and 1=column
        new_array
```

```
[18]: array([[0.39323667, 0.6394314 , 0.36252106, 0.46506304],
            [0.55616985, 0.65306829, 0.884103  , 0.69778038],
            [0.42144908, 0.92911412, 0.96655347, 0.77237222],
            ...,
            [0.373153  , 0.46836664, 0.26040695, 0.36730886],
            [0.86446074, 0.82746778, 0.71771681, 0.80321511],
            [0.69594843, 0.44118409, 0.64880824, 0.59531359]])
```

```
[19]: np.savetxt('mean.csv',new_array, fmt='%.18e',header='num1,num2,num3,mean',
            ↪delimiter=' ') #this function save array to file
```

```
[20]: np.max(new_array[:,3])
```

```
[20]: 0.9339277299816967
```

```
[21]: np.min(new_array[:,3])
```

```
[21]: 0.04305083486581851
```

```
[ ]:
```