Lab 3
Python Basic Practice-III
Date - 25 Jan 2024

Aditi Shrivastava – 210905244
Section – CSE-A
Roll number – 44

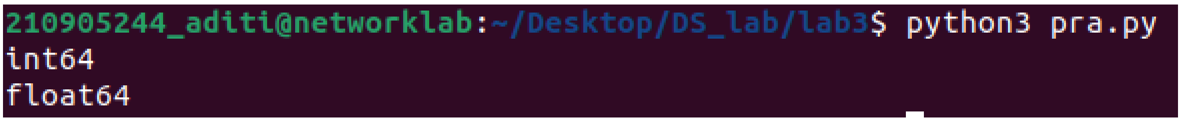
Practice

1. Array Creation

A = np.array([2, 5, 10])
print(A.dtype)

B = np.array([2.4, 10.6, 5.2])
print(B.dtype)

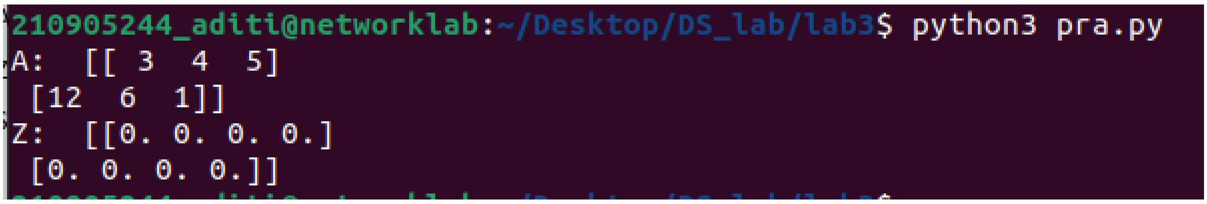
OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
int64
float64
```

2. Creating sequence of sequence will create 2-dimensional array.
A=np.array([(3,4,5),(12,6,1)])
Z=np.zeros((2,4))
print("A: ", A)
print("Z: ", Z)

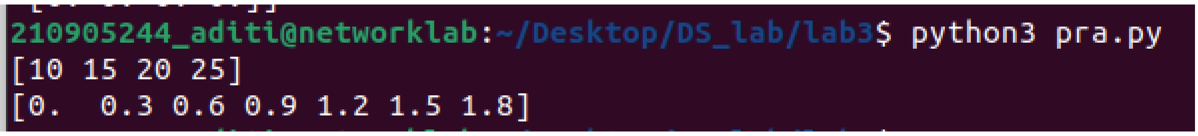
OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
A:  [[ 3  4  5]
 [12  6  1]]
Z:  [[0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

3. To create a sequence of data
S = np.arange(10, 30, 5)
print(S)
B= np.arange(0, 2, 0.3)
print(B)

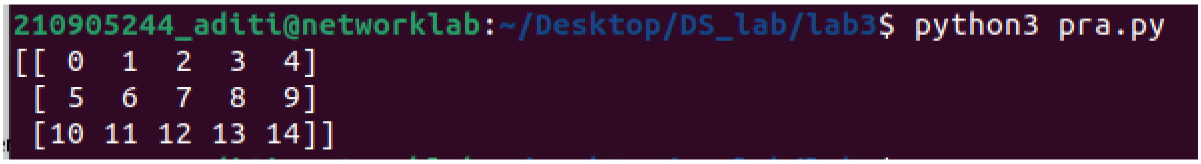
OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[10 15 20 25]
[0.  0.3 0.6 0.9 1.2 1.5 1.8]
```

2D Matrix
a = np.arange(15).reshape(3, 5)
print(a)

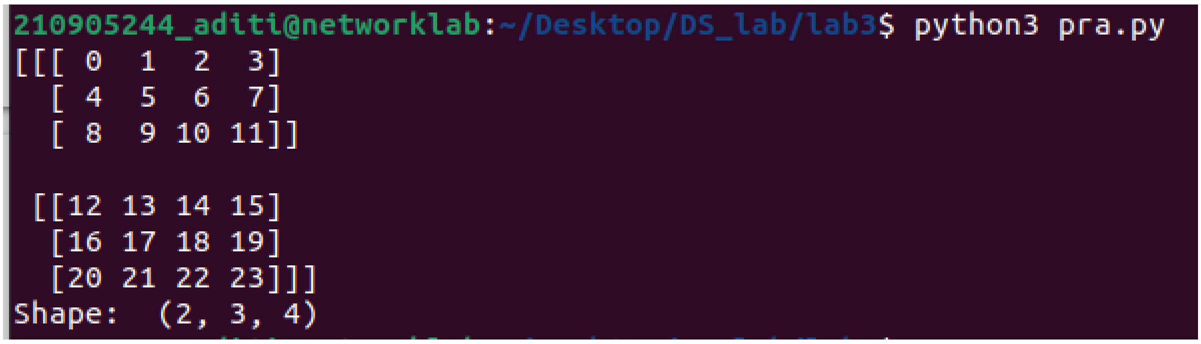
OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]]
```

3D Matrix
c = np.arange(24).reshape(2, 3, 4)
print(c)
print("Shape: ", c.shape)

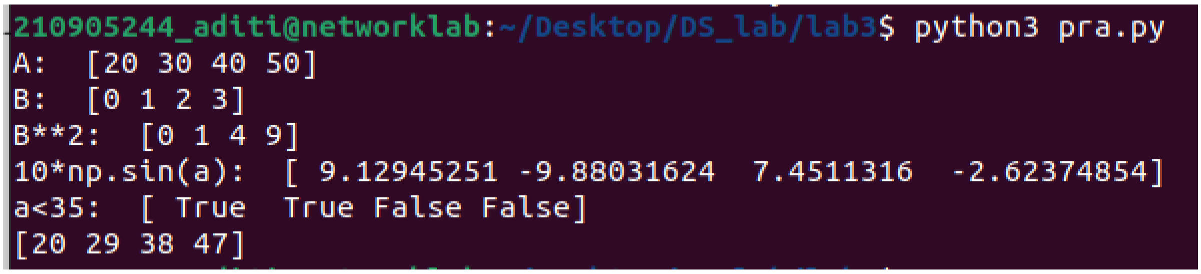
OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
Shape:  (2, 3, 4)
```

Array Operations
a= np.array([20, 30, 40, 50])
b = np.arange(4)
print("A: ", a)
print("B: ", b)
print("B**2: ", b**2)
print("10*np.sin(a): ", 10*np.sin(a))
print("a<35: ", a<35)
c = a-b
print(c)

OUTPUT:

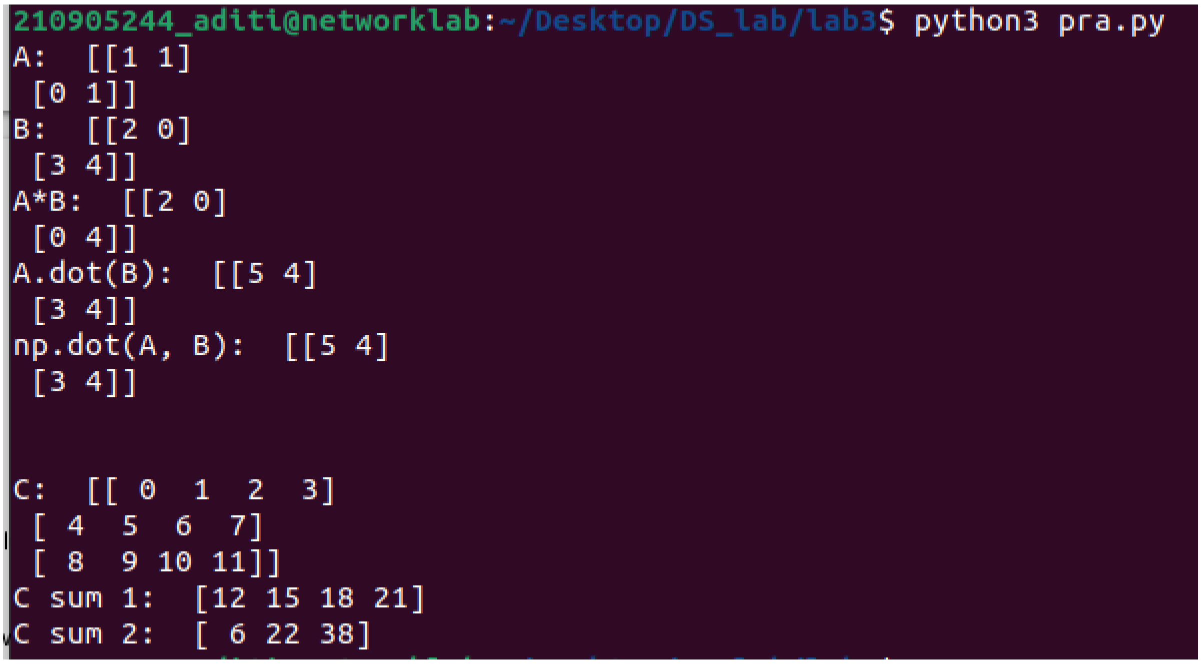


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
A:  [20 30 40 50]
B:  [0 1 2 3]
B**2:  [0 1 4 9]
10*np.sin(a):  [ 9.12945251 -9.88031624  7.4511316  -2.62374854]
a<35:  [ True  True False False]
[20 29 38 47]
```

Matrix Operations

```
A = np.array([[1, 1], [0, 1]])
B = np.array([[2, 0], [3, 4]])
print("A: ", A)
print("B: ", B)
print("A*B: ", A*B)
print("A.dot(B): ", A.dot(B))
print("np.dot(A, B): ", np.dot(A, B))
C= np.arange(12).reshape(3, 4)
print("\n\nC: ", C)
print("C sum 1: ",C.sum(axis = 0))
print("C sum 2: ", C.sum(axis = 1))
```

OUTPUT:

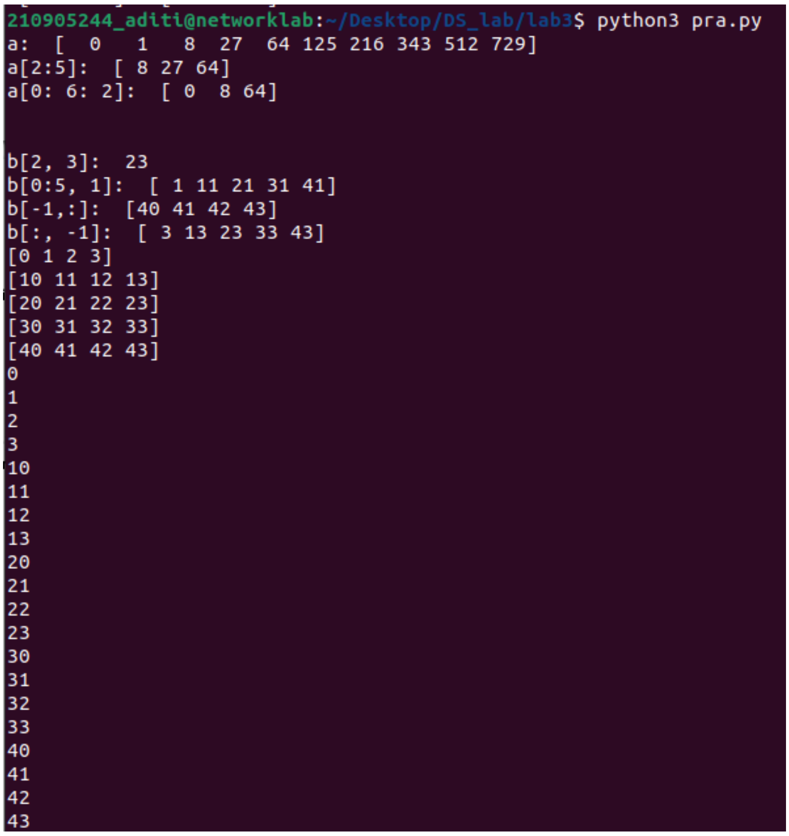


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
A:  [[1 1]
 [0 1]]
B:  [[2 0]
 [3 4]]
A*B:  [[2 0]
 [0 4]]
A.dot(B):  [[5 4]
 [3 4]]
np.dot(A, B):  [[5 4]
 [3 4]]


C:  [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
C sum 1:  [12 15 18 21]
C sum 2:  [ 6 22 38]
```

Indexing, Slicing and Iterating Array

```
a = np.arange(10)**3
print("a: ", a)
print("a[2:5]: ", a[2: 5])
print("a[0: 6: 2]: ", a[0: 6: 2])

b = np.array([[ 0, 1, 2, 3],
[10, 11, 12, 13],
[20, 21, 22, 23],
[30, 31, 32, 33],
[40, 41, 42, 43]])

print("\n\nb[2, 3]: ", b[2,3])
print("b[0:5, 1]: ", b[0: 5,1])
print("b[-1,:]: ", b[-1, :])
print("b[:, -1]: ", b[:, -1])
for row in b:
    print(row)
for element in b.flat:
    print(element)
```

OUTPUT:

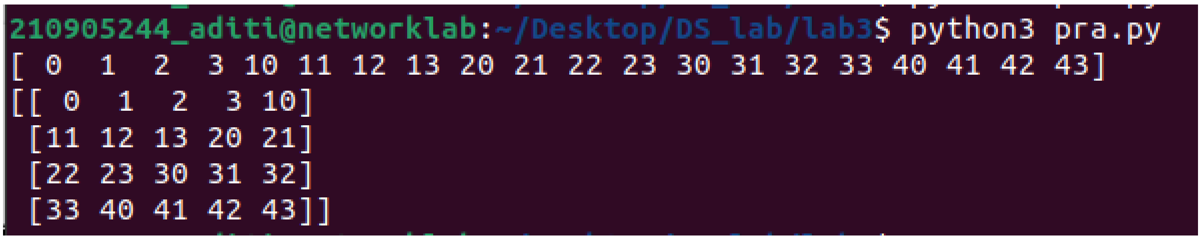


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
a:  [  0   1   8  27  64 125 216 343 512 729]
a[2:5]:  [ 8 27 64]
a[0: 6: 2]:  [ 0  8 64]


b[2, 3]:  23
b[0:5, 1]:  [ 1 11 21 31 41]
b[-1,:]:  [40 41 42 43]
b[:, -1]:  [ 3 13 23 33 43]
[0 1 2 3]
[10 11 12 13]
[20 21 22 23]
[30 31 32 33]
[40 41 42 43]
0
1
2
3
10
11
12
13
20
21
22
23
30
31
32
33
40
41
42
43
```

Changing shape of a matrix

```
b = np.array([[ 0, 1, 2, 3],
[10, 11, 12, 13],
[20, 21, 22, 23],
[30, 31, 32, 33],
[40, 41, 42, 43]])

print(b.ravel())
B1 = b.reshape(4, 5)
print(B1)
```

OUTPUT:

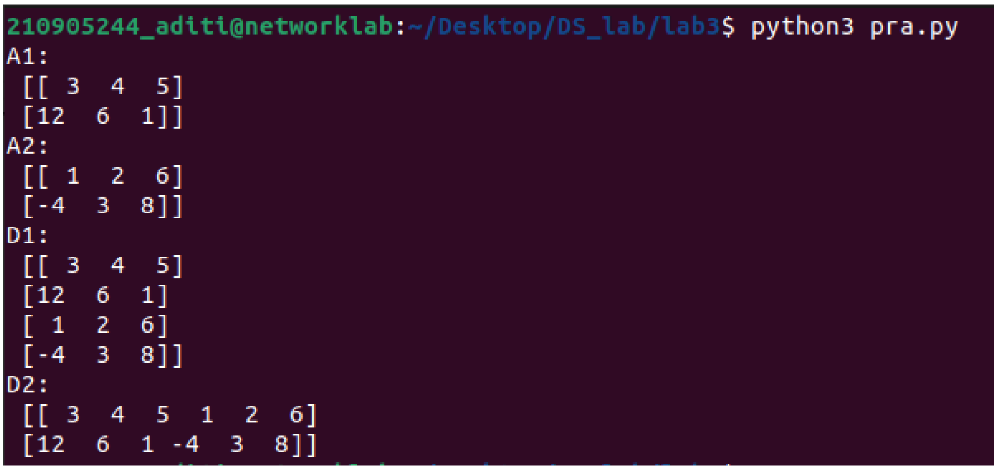


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
[[ 0  1  2  3 10]
 [11 12 13 20 21]
 [22 23 30 31 32]
 [33 40 41 42 43]]
```

Stacking together different arrays

```
A1 = np.array([(3, 4, 5), (12, 6, 1)])
print("A1: \n",A1)
A2=np.array([(1,2,6),(-4,3,8)])
print("A2: \n", A2)

D1=np.vstack((A1,A2))
print("D1: \n", D1)

D2=np.hstack((A1,A2))
print("D2: \n", D2)
```

OUTPUT:

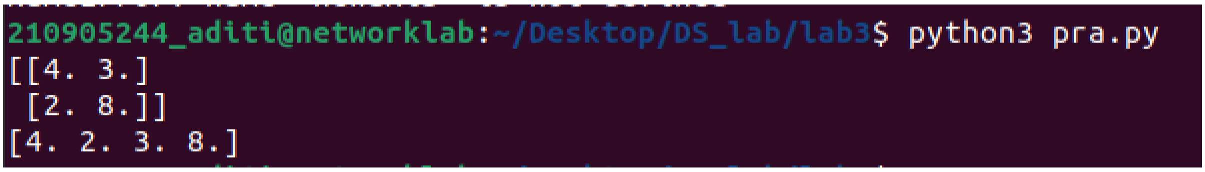


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
A1:
 [[ 3  4  5]
 [12  6  1]]
A2:
 [[ 1  2  6]
 [-4  3  8]]
D1:
 [[ 3  4  5]
 [12  6  1]
 [ 1  2  6]
 [-4  3  8]]
D2:
 [[ 3  4  5  1  2  6]
 [12  6  1 -4  3  8]]
```

Stacking 1D array into 2D array (column wise)

```
a = np.array([4., 2.])
b = np.array([3., 8.])
print(np.column_stack((a, b)))
print(np.hstack((a, b)))
```

OUTPUT:

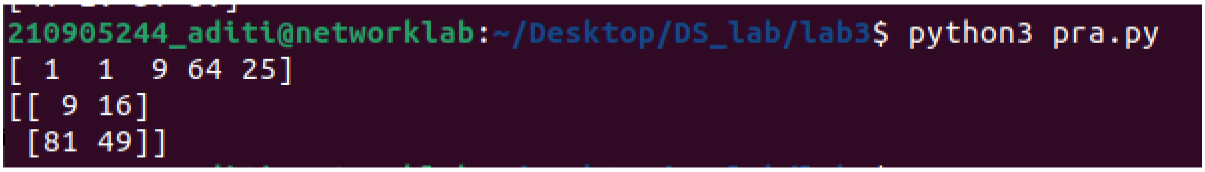


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[[4. 3.]
 [2. 8.]]
[4. 2. 3. 8.]
```

Indexing with array of indices

```
a = np.arange(12)**2
i = np.array([1, 1, 3, 8, 5])
print(a[i])

j= np.array([[3, 4], [9, 7]])
print(a[j])
```

OUTPUT:

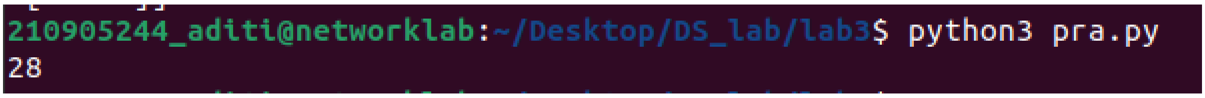


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
[ 1  1  9 64 25]
[[ 9 16]
 [81 49]]
```

Usage of for-loop(Mapping by value)

```
a=np.array([(3,2,9),(1,6,7)])
s1=0
for row in a:
    for col in row:
            s1+=col
print(s1)
```

OUTPUT:

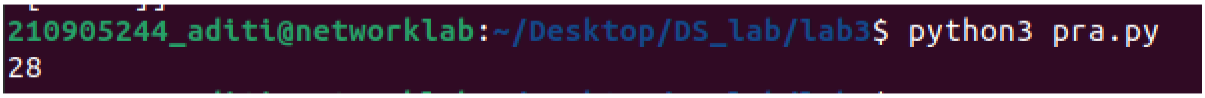


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
28
```

Usage of for-loop (Mapping by index)

```
a = np.array([(3, 2, 9), (1, 6, 7)])
s=0
for i in range(a.shape[0]):
    for j in range(a.shape[1]):
        s+=a[i, j]
print(s)
```

OUTPUT:



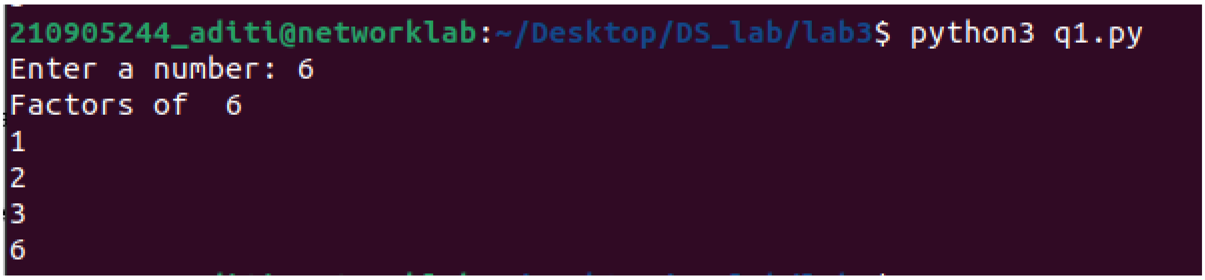
```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 pra.py
28
```

Lab Exercise Programs

1. Write a program to find the factors of a given number (get input from user) using for loop

```
num = int(input("Enter a number: "))
print("Factors of ", num)
for i in range(1, num+1):
   if num%i ==0:
        print(i)
```

OUTPUT:

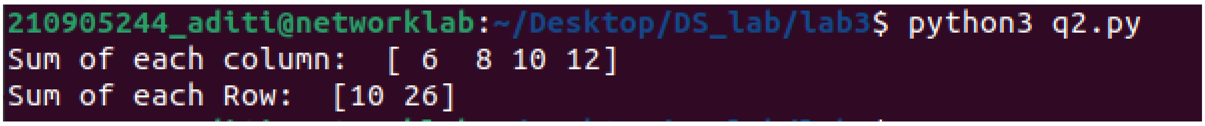


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q1.py
Enter a number: 6
Factors of  6
1
2
3
6
```

2.Find the sum of columns and rows using axis

```
import numpy as np

A = np.array([[1, 2, 3, 4],
   [5, 6, 7, 8]])
print("Sum of each column: ", A.sum(axis=0))
print("Sum of each Row: ", A.sum(axis=1))
```

OUTPUT:



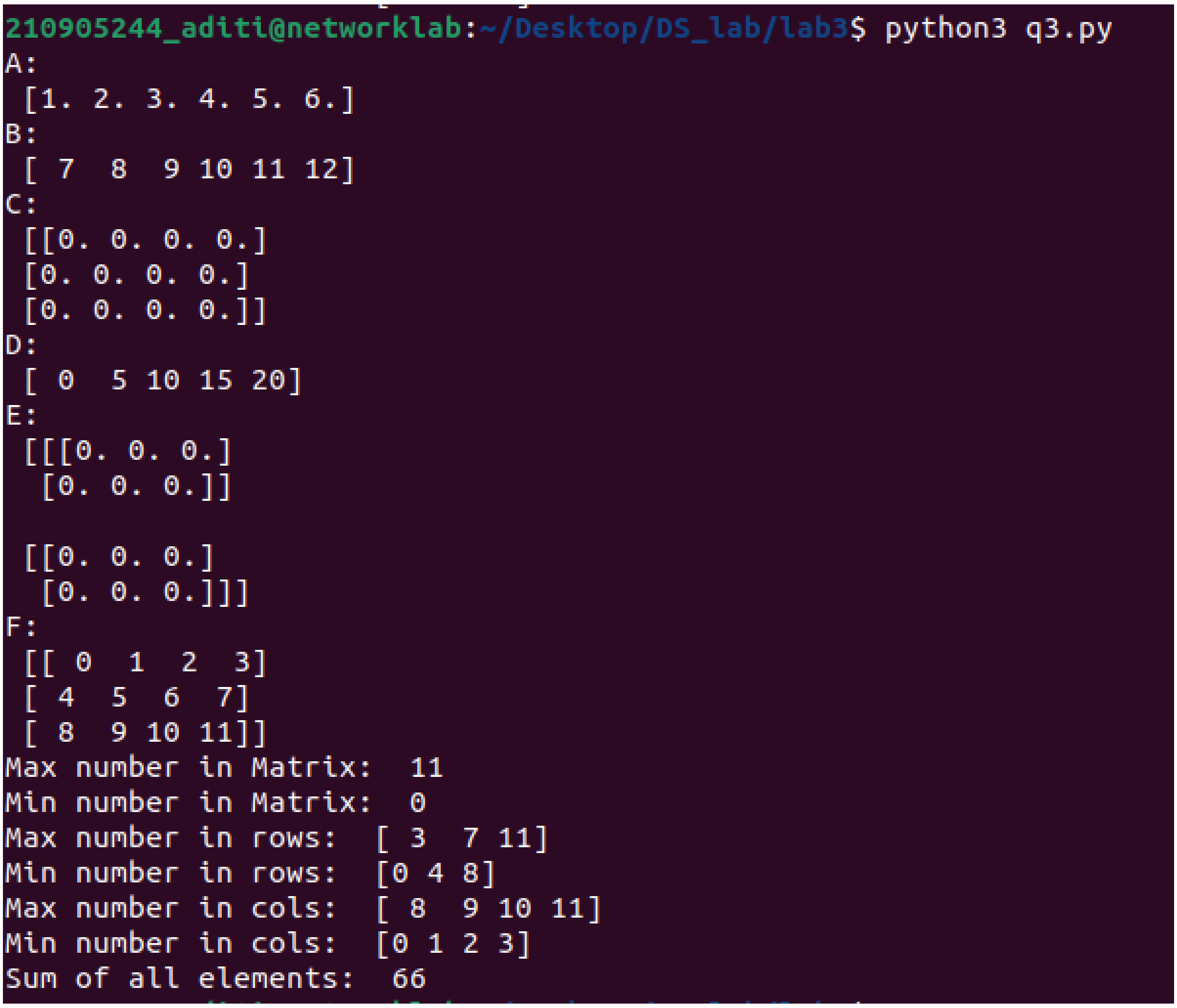
```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q2.py
Sum of each column:  [ 6  8 10 12]
Sum of each Row:  [10 26]
```

3. Operations on Arrays (use numpy wherever required):
a) Create array from list with type float
b) Create array from tuple
c) Creating a 3X4 array with all zeros
d) Create a sequence of integers from 0 to 20 with steps of 5
e) Reshape 3X4 array to 2X2X3 array
f) Find maximum and minimum element of array, Row wise max and min, column wise max
and min and sum of elements. (Use functions max(), min(), sum())

```
import numpy as np

list = [1, 2, 3 ,4 ,5, 6]
A = np.array(list, dtype=float)
print("A: \n", A)

tuple = (7, 8, 9, 10, 11, 12)
B = np.array(tuple)
print("B: \n", B)

C = np.zeros((3, 4))
print("C: \n", C)

D = np.arange(0, 21, 5)
print("D: \n", D)

E = C.reshape(2, 2, 3)
print("E: \n", E)

F = np.arange(12).reshape(3, 4)
print("F: \n", F)
F_max = np.max(F)
print("Max number in Matrix: ", F_max)
F_min = np.min(F)
print("Min number in Matrix: ", F_min)
F_max_row = np.max(F, axis = 0)
print("Max number in rows: ", F_max_row)
F_min_row = np.min(F, axis = 0)
print("Min number in rows: ", F_min_row)
F_max_col = np.max(F, axis = 1)
print("Max number in cols: ", F_max_col)
F_min_col = np.min(F, axis = 1)
print("Min number in cols: ", F_min_col)
```

F_sum = np.sum(F)
print("Sum of all elements: ", F_sum)

OUTPUT

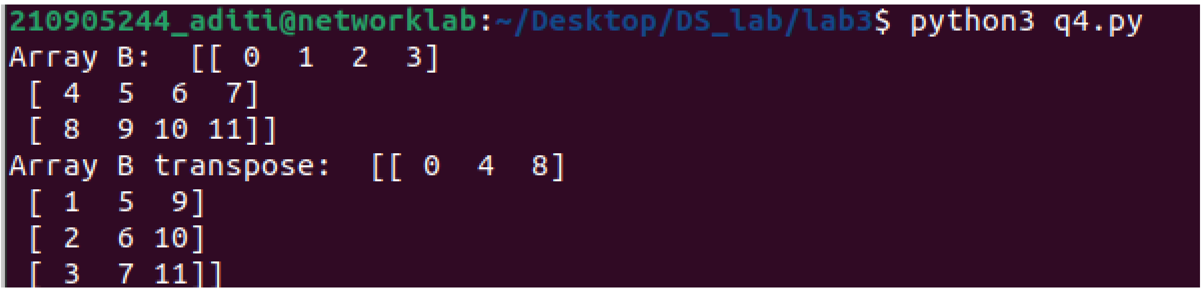


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q3.py
A:
 [1. 2. 3. 4. 5. 6.]
B:
 [ 7  8  9 10 11 12]
C:
 [[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
D:
 [ 0  5 10 15 20]
E:
 [[[0. 0. 0.]
  [0. 0. 0.]]

 [[0. 0. 0.]
  [0. 0. 0.]]]
F:
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
Max number in Matrix:  11
Min number in Matrix:  0
Max number in rows:  [ 3  7 11]
Min number in rows:  [0 4 8]
Max number in cols:  [ 8  9 10 11]
Min number in cols:  [0 1 2 3]
Sum of all elements:  66
```

4. Write a program to transpose a given matrix

```
import numpy as np
B = np.arange(12).reshape(3, 4)
print("Array B: ", B)
B_T = B.transpose()
print("Array B transpose: ", B_T)
```

OUTPUT:

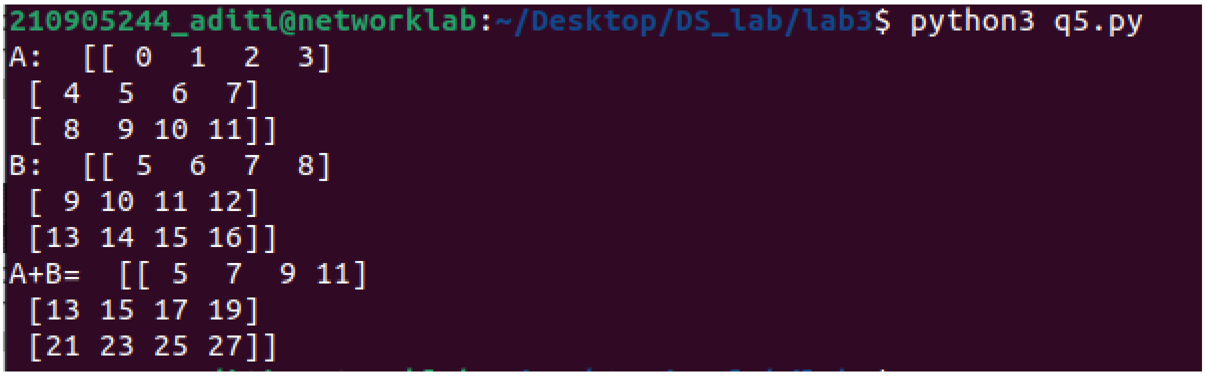


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q4.py
Array B:  [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
Array B transpose:  [[ 0  4  8]
 [ 1  5  9]
 [ 2  6 10]
 [ 3  7 11]]
```

5. Write a program to add two matrices

```
import numpy as np
A = np.arange(12).reshape(3, 4)
B = np.arange(5, 17).reshape(3, 4)
print("A: ", A)
print("B: ", B)
print("A+B= ", A+B)
```

OUTPUT:

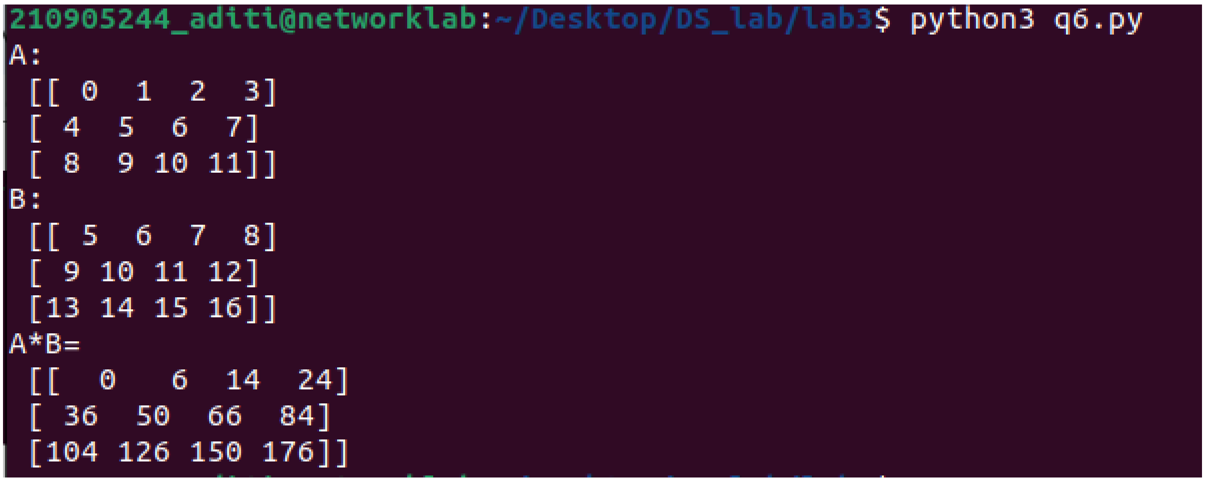


```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q5.py
A:  [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
B:  [[ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
A+B=  [[ 5  7  9 11]
 [13 15 17 19]
 [21 23 25 27]]
```

6. Write a program to find element wise product between two matrices

```
import numpy as np
A = np.arange(12).reshape(3, 4)
B = np.arange(5, 17).reshape(3, 4)
print("A: \n", A)
print("B: \n", B)
print("A*B= \n", np.multiply(A, B))
```

OUTPUT:



```
210905244_aditi@networklab:~/Desktop/DS_lab/lab3$ python3 q6.py
A:
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
B:
 [[ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
A*B=
 [[  0   6  14  24]
 [ 36  50  66  84]
 [104 126 150 176]]
```