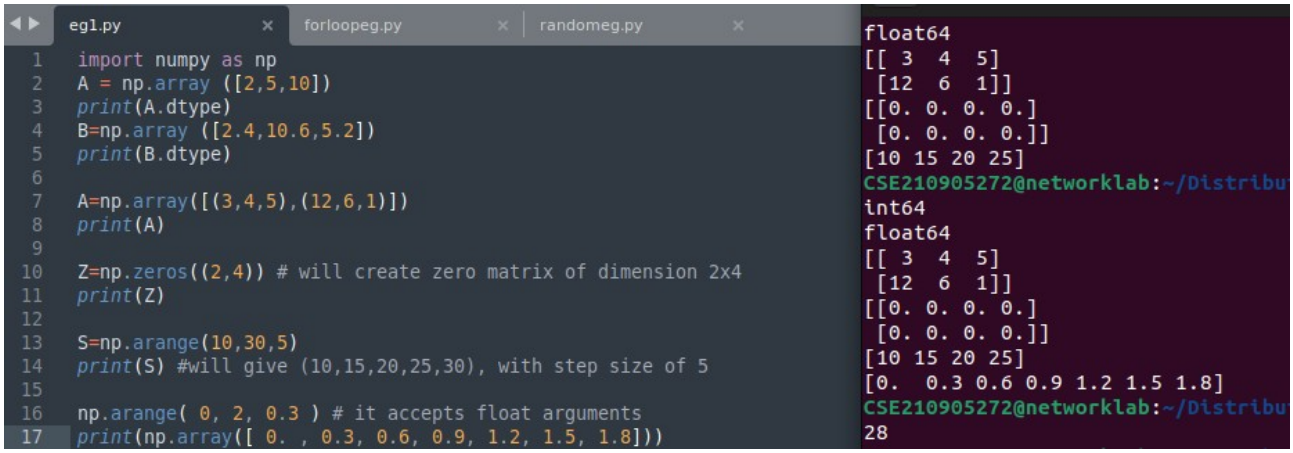
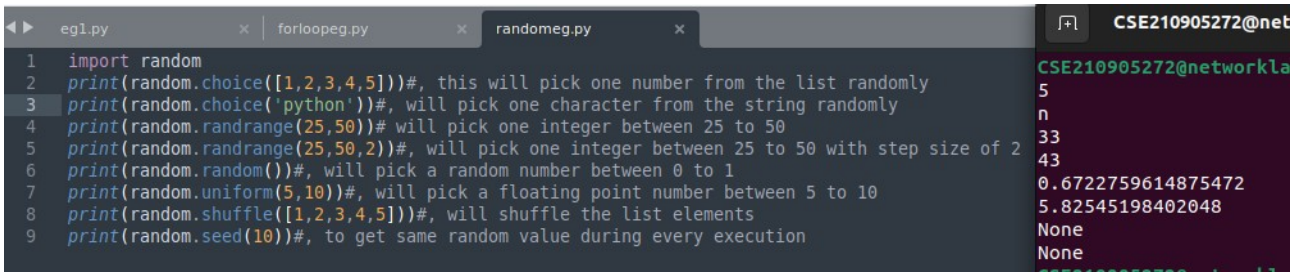
LAB – 3
PYTHON BASIC PRACTICE – III

```
import numpy as np

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

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

A=np.array([(3,4,5),(12,6,1)])
print(A)

Z=np.zeros((2,4)) # will create zero matrix of dimension 2x4
print(Z)
```

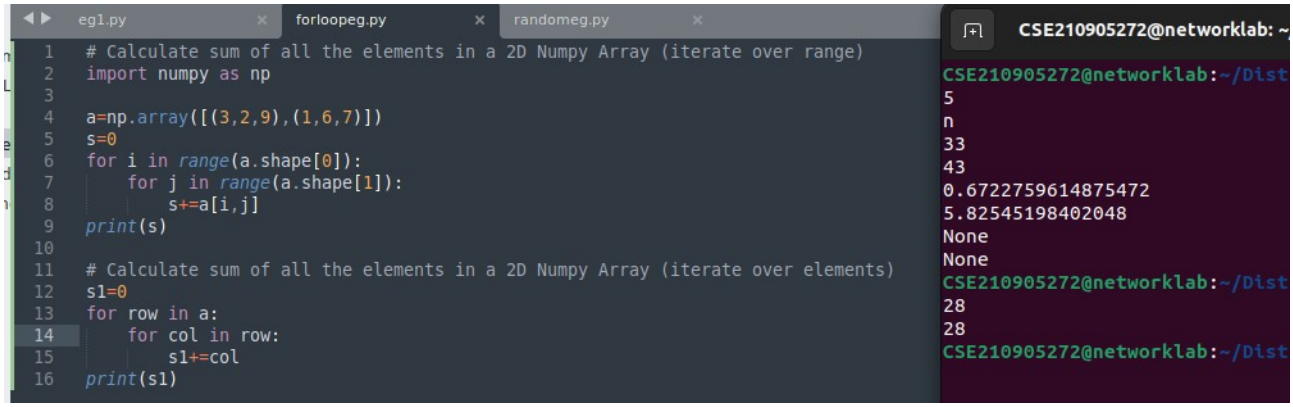



```
import random
print(random.choice([1,2,3,4,5]))#, this will pick one number from the list randomly
print(random.choice('python'))#, will pick one character from the string randomly
print(random.randrange(25,50))# will pick one integer between 25 to 50
print(random.randrange(25,50,2))#, will pick one integer between 25 to 50 with step size of 2
print(random.random())#, will pick a random number between 0 to 1
print(random.uniform(5,10))#, will pick a floating point number between 5 to 10
print(random.shuffle([1,2,3,4,5]))#, will shuffle the list elements
print(random.seed(10))#, to get same random value during every execution
```

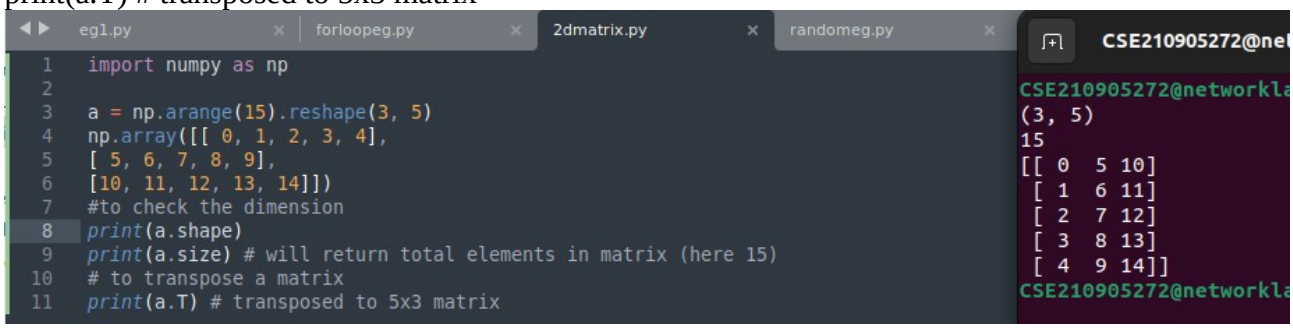


Calculate sum of all the elements in a 2D Numpy Array (iterate over range)

```
import numpy as np

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)

# Calculate sum of all the elements in a 2D Numpy Array (iterate over elements)
s1=0
for row in a:
        for col in row:
                s1+=col
print(s1)
```



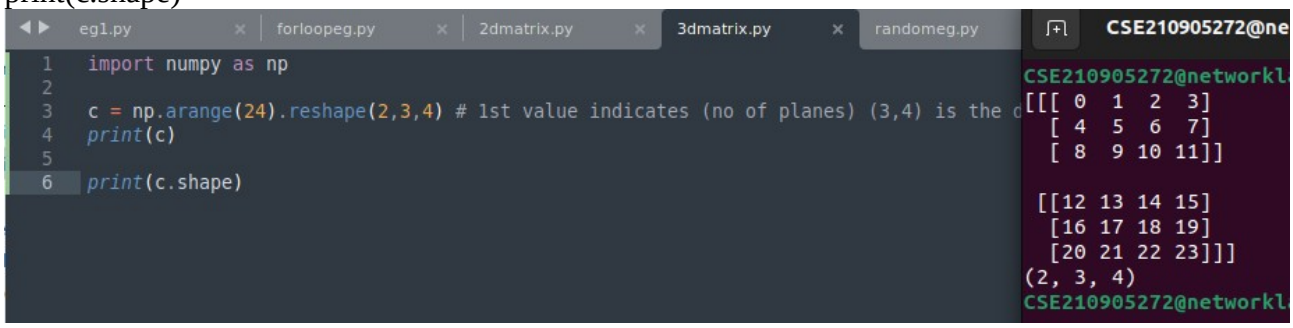

```
import numpy as np

a = np.arange(15).reshape(3, 5)
np.array([[ 0, 1, 2, 3, 4],
[ 5, 6, 7, 8, 9],
[10, 11, 12, 13, 14]])
#to check the dimension
print(a.shape)
print(a.size) # will return total elements in matrix (here 15)
# to transpose a matrix
print(a.T) # transposed to 5x3 matrix
```




```
import numpy as np
```

c = np.arange(24).reshape(2,3,4) # 1st value indicates (no of planes) (3,4) is the dimension
print(c)

print(c.shape)



import numpy as np
a = np.array([20,30,40,50])
b = np.arange(4)
print(b)
# array([0, 1, 2, 3])
c = a−b
print(c)
# array([20, 29, 38, 47])

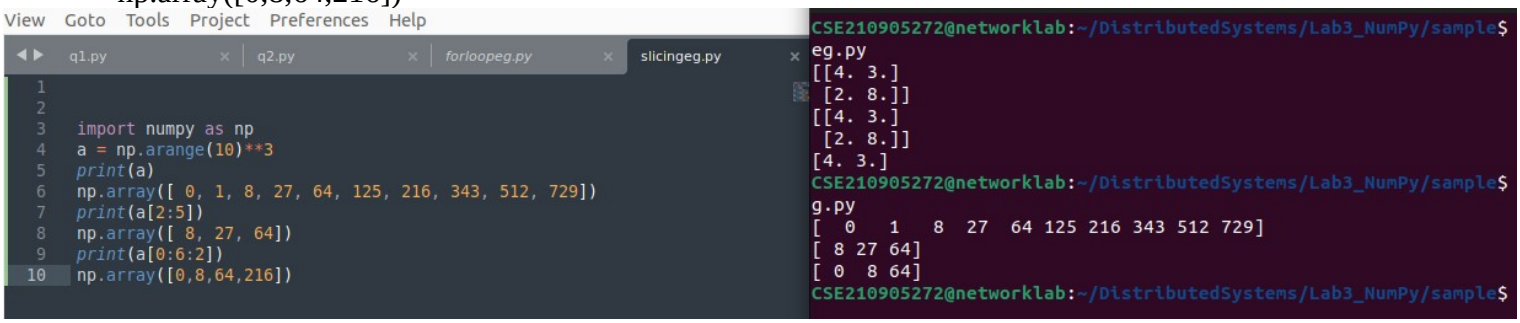
print(b**2)
array([0, 1, 4, 9])
10*np.sin(a)
array([9.12945251, −9.88031624, 7.4511316 , −2.62374854])
a<35
array([True, True, False, False], dtype=bool)

import numpy as np
a = np.array([4.,2.])
b = np.array([3.,8.])
print(np.column_stack((a,b))) # returns a 2D array
print(np.array([[4., 3.],
[2., 8.]]))

np.hstack((a,b))
# the result is different
np.array([4., 2., 3., 8.])
print(np.hstack((a[0],b[0]))) # the result is the same
np.array([[4., 3.],
[2., 8.]])

import numpy as np
a = np.arange(10)**3
print(a)
np.array([0, 1, 8, 27, 64, 125, 216, 343, 512, 729])

print(a[2:5])
np.array([8, 27, 64])
print(a[0:6:2])
np.array([0,8,64,216])



import numpy as np

A = np.array([[1,1],[0,1]])
B = np.array([[2,0],[3,4]])
print(A*B)
# elementwise product
np.array([[2, 0],
[0, 4]])
A.dot(B)
# matrix product
np.array([[5, 4],
[3, 4]])
# (OR)
np.dot(A, B)
np.array([[5, 4],
[3, 4]])
# another matrix product
b = np.arange(12).reshape(3,4)
print(b)
np.array([[0, 1, 2, 3],
[4, 5, 6, 7],
[8, 9, 10, 11]])
b.sum(axis=0)
np.array([12, 15, 18, 21])# sum of each column
b.sum(axis=1)
np.array([6, 22, 38])

```python
import numpy as np

A = np.array( [[1,1],[0,1]] )
B = np.array( [[2,0],[3,4]] )
print(A*B)
# elementwise product
np.array([[2, 0],
[0, 4]])
A.dot(B)
# matrix product
np.array([[5, 4],
[3, 4]])
# (OR)
np.dot(A, B)
np.array([[5, 4],
[3, 4]])
# another matrix product
b = np.arange(12).reshape(3,4)
print(b)
np.array([[ 0, 1, 2, 3],
[ 4, 5, 6, 7],
[ 8, 9, 10, 11]])
b.sum(axis=0)
np.array([12, 15, 18, 21])# sum of each column
b.sum(axis=1)
np.array([6, 22, 38])
```

```
CSE210905272@networklab:~/D
seg.py
[[2 0]
 [0 4]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
CSE210905272@networklab:~/D
```

```python
import numpy as np
b=np.array([[ 0, 1, 2, 3],
[10, 11, 12, 13],
[20, 21, 22, 23],
[30, 31, 32, 33],
[40, 41, 42, 43]])
b[2,3] #will fetch 23
b[0:5,1] #or b[:5,1] or b[:,1] #will fetch [1,11,21,31,41]
b[-1,:] # will fetch last row
b[:,-1] # will fetch last col
for row in b:
    print (row) # will print every rowfor element in b.flat:
# print (element) # will show all elements of b in 1-D array
# Changing the shape of a matrix
b.ravel() # returns the array flattened to (1x 20)
# Later, we can convert 5
```

```
CSE210905272@networklab:~/Distributed
[0 1 2 3]
[10 11 12 13]
[20 21 22 23]
[30 31 32 33]
[40 41 42 43]
CSE210905272@networklab:~/Distributed
```

LAB EXERCISES:

Q1)Write a program to find the factors of a given number (get input from user) using for loop.

->

```
import numpy as np
import pandas as pd

n=int(input("Enter number: "))
print("The factors are: \n")
for i in range(1,n+1):
        if (n%i==0):
                print(i)
```

```
# Write a program to find the factors of a given number (get input from user)

import numpy as np
import pandas as pd
from icecream import ic

n=int(input("Enter number: "))
print("The factors are: \n")
for i in range(1,n+1):
    if (n%i==0):
        # ic(i)
        print(i)
```

```
CSE210905272@networklab:~/DistributedSys
Enter number: 12
The factors are:

1
2
3
4
6
12
CSE210905272@networklab:~/DistributedSys
```

Q2)Find the sum of columns and rows using axis.

->

```
import numpy as np
import pandas as pd
from icecream import ic

a=np.array([[ 0, 1, 2, 3, 4],
[ 5, 6, 7, 8, 9],
[10, 11, 12, 13, 14]])
ic("ROW sum:")
print(a.sum(axis=1))
ic("COLUMN sum:")
print(a.sum(axis=0))
```

```
# Find the sum of columns and rows using axis

import numpy as np
import pandas as pd
from icecream import ic

a=np.array([[ 0, 1, 2, 3, 4],
[ 5, 6, 7, 8, 9],
[10, 11, 12, 13, 14]])
ic("ROW sum:")
print(a.sum(axis=1))
ic("COLUMN sum:")
print(a.sum(axis=0))
```

```
CSE210905272@networklab:~/D
ic| 'ROW sum:'
[10 35 60]
ic| 'COLUMN sum:'
[15 18 21 24 27]
CSE210905272@networklab:~/D
```

Q3)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
import pandas as pd
from icecream import ic

#a
list1=[1,2,3,4]
arr1=np.array(list1, dtype="float")
print(arr1)

#b
tuple1=(7,6,5,4)
arr2=np.array(tuple1, dtype="float")
print(arr2)

#c
print("Zeroes matrix:")
zeroes1=np.zeros((3,4))
print(zeroes1_

#d
print(np.arange(0,20,5))

#e
e=zeroes1.reshape(2,2,3)
print(e)

#f
f=np.array([[1,11,2],[3,55,6]])
print("The array: ")
print(f)
print(f.max(axis=0))
print(f.min(axis=0))
print(f.max(axis=1))
print(f.min(axis=1))
print("Sum: ",f.sum())
```

q1.py q2.py q3.py q4.py q5.py q6.py

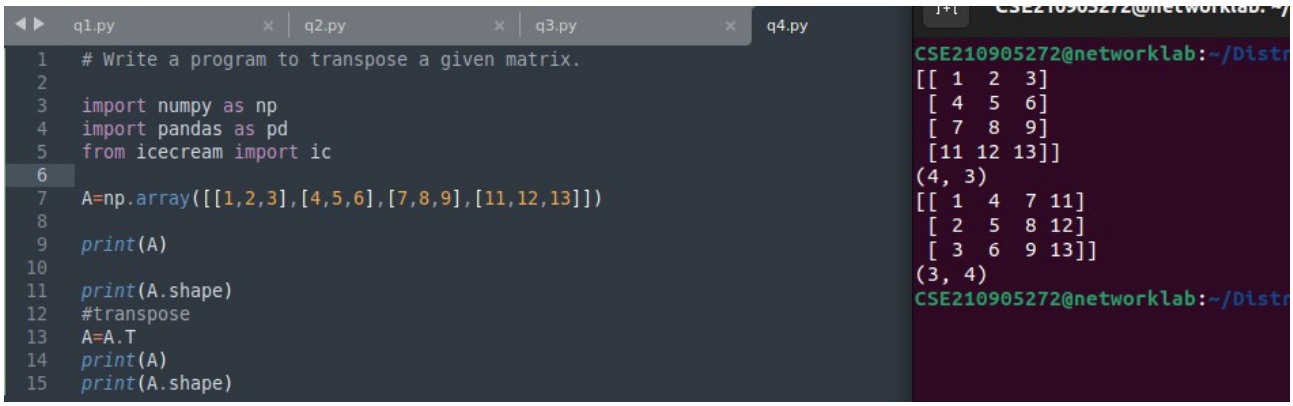
```python
# d) Create a sequence of integers from 0 to 20 with steps of 5
# e) Reshape 3X4 array to 2X2X3 array
# f) Find maximum and minimum element of array, Row wise max and min, column wise max
# and min and sum of elements. (Use functions max(), min(), sum())

import numpy as np
import pandas as pd
from icecream import ic
#a
list1=[1,2,3,4]
arr1=np.array(list1, dtype="float")
print(arr1)
#b
tuple1=(7,6,5,4)
arr2=np.array(tuple1, dtype="float")
print(arr2)
#c
print("Zeroes matrix:")
zeroes1=np.zeros((3,4))
print(zeroes1)
#d
print(np.arange(0,20,5))

#e
e=zeroes1.reshape(2,2,3)
print(e)

#f

f=np.array([[1,11,2],[3,55,6]])
print("The array: ")
print(f)
print(f.max(axis=0))
print(f.min(axis=0))
print(f.max(axis=1))
print(f.min(axis=1))
```

CSE210905272@networklab: ~/DistributedSystems/L

```
[1 3]
78
CSE210905272@networklab:~/DistributedSystems/Lab
[1. 2. 3. 4.]
[7. 6. 5. 4.]
Zeroes matrix:
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[ 0  5 10 15]
[[[0. 0. 0.]
  [0. 0. 0.]]

 [[0. 0. 0.]
  [0. 0. 0.]]]
The array:
[[ 1 11  2]
 [ 3 55  6]]
[ 3 55  6]
[ 1 11  2]
[11 55]
[1 3]
Sum:  78
CSE210905272@networklab:~/DistributedSystems/Lab
```

Q4)Write a program to transpose a given matrix.

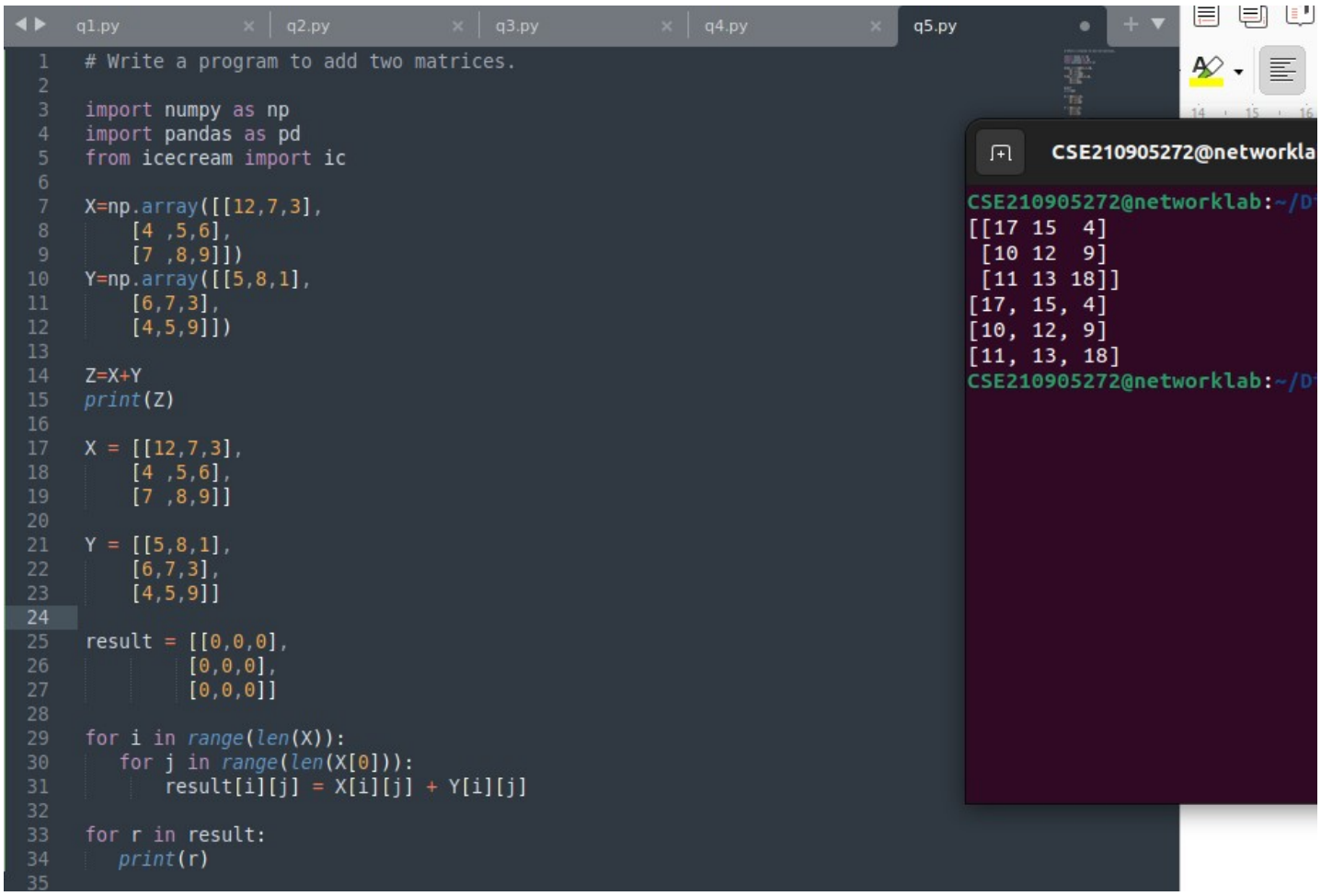
```
import numpy as np
import pandas as pd
from icecream import ic

A=np.array([[1,2,3],[4,5,6],[7,8,9],[11,12,13]])

print(A)
print(A.shape)

#transpose
A=A.T
print(A)
print(A.shape)
```



Q5)Write a program to add two matrices.

->

```
import numpy as np
import pandas as pd
from icecream import ic

X=np.array([[12,7,3],
    [4 ,5,6],
    [7 ,8,9]])
Y=np.array([[5,8,1],
    [6,7,3],
    [4,5,9]])

Z=X+Y
print(Z)
```



Q6)Write a program to find element wise product between two matrices.

->

```
import numpy as np
import pandas as pd
from icecream import ic

X=np.array([[12,7,3],
   [4 ,5,6],
   [7 ,8,9]])
Y=np.array([[5,8,1],
   [6,7,3],
   [4,5,9]])

Z=X*Y
print(Z)
```

q1.py | q2.py | q3.py | q4.py | q5.py | q6.py

```python
# Write a program to find element wise product between two matrices.

import numpy as np
import pandas as pd
from icecream import ic

X=np.array([[12,7,3],
    [4 ,5,6],
    [7 ,8,9]])
Y=np.array([[5,8,1],
    [6,7,3],
    [4,5,9]])

Z=X*Y
print(Z)

X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]

Y = [[5,8,1],
    [6,7,3],
    [4,5,9]]

Z = [[0,0,0],
         [0,0,0],
         [0,0,0]]

for i in range(len(X)):
   for j in range(len(X[0])):
       Z[i][j] = X[i][j] * Y[i][j]

for r in Z:
   print(r)
```

CSE210905272@networklab: ~/DistributedSystems/Lab

```
CSE210905272@networklab:~/DistributedSystems/Lab3_
[[60 56  3]
 [24 35 18]
 [28 40 81]]
[60, 56, 3]
[24, 35, 18]
[28, 40, 81]
CSE210905272@networklab:~/DistributedSystems/Lab3_
```