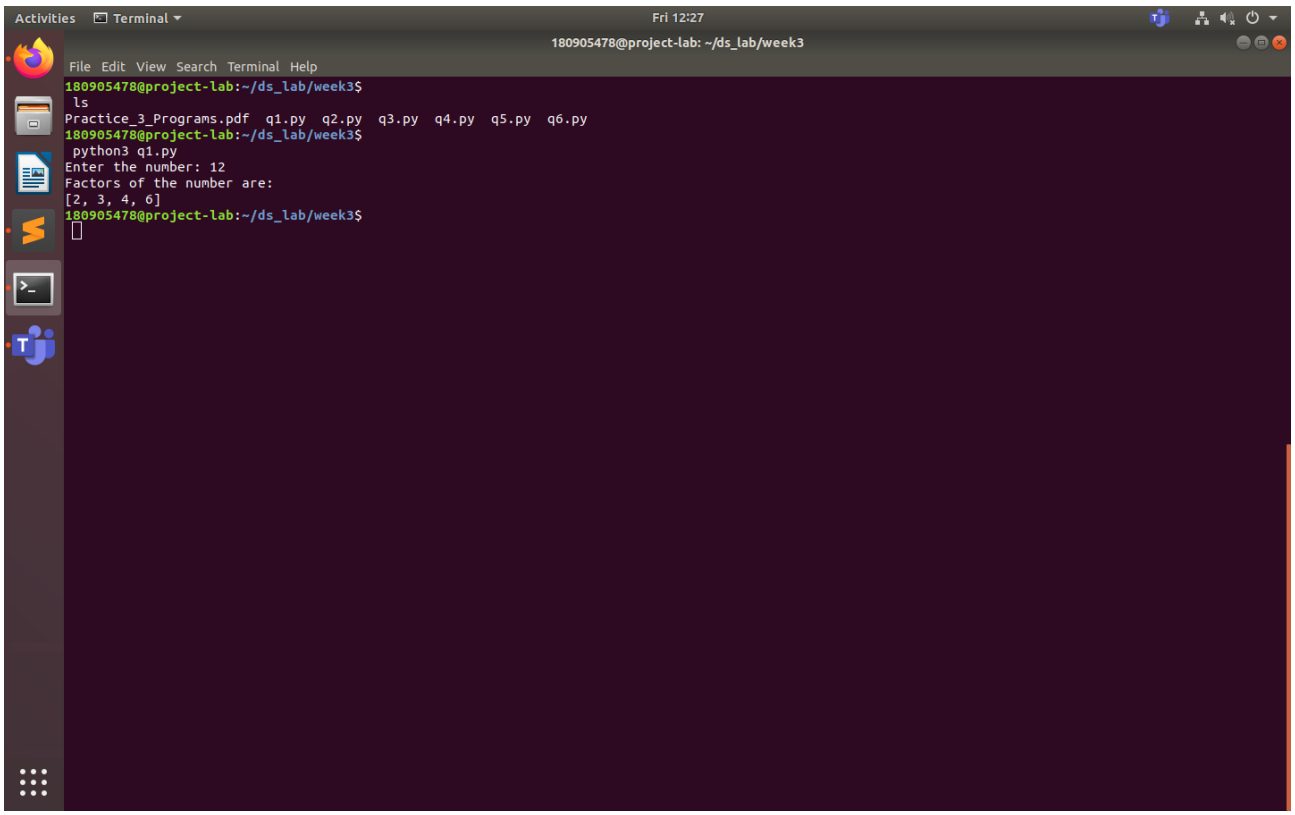
LAB3 – Distirbuted Systems Lab
Name: Sagnik Chatterjee
Reg:- 180905478
Roll No : 61
Batch : B2
SEC : B

Q1

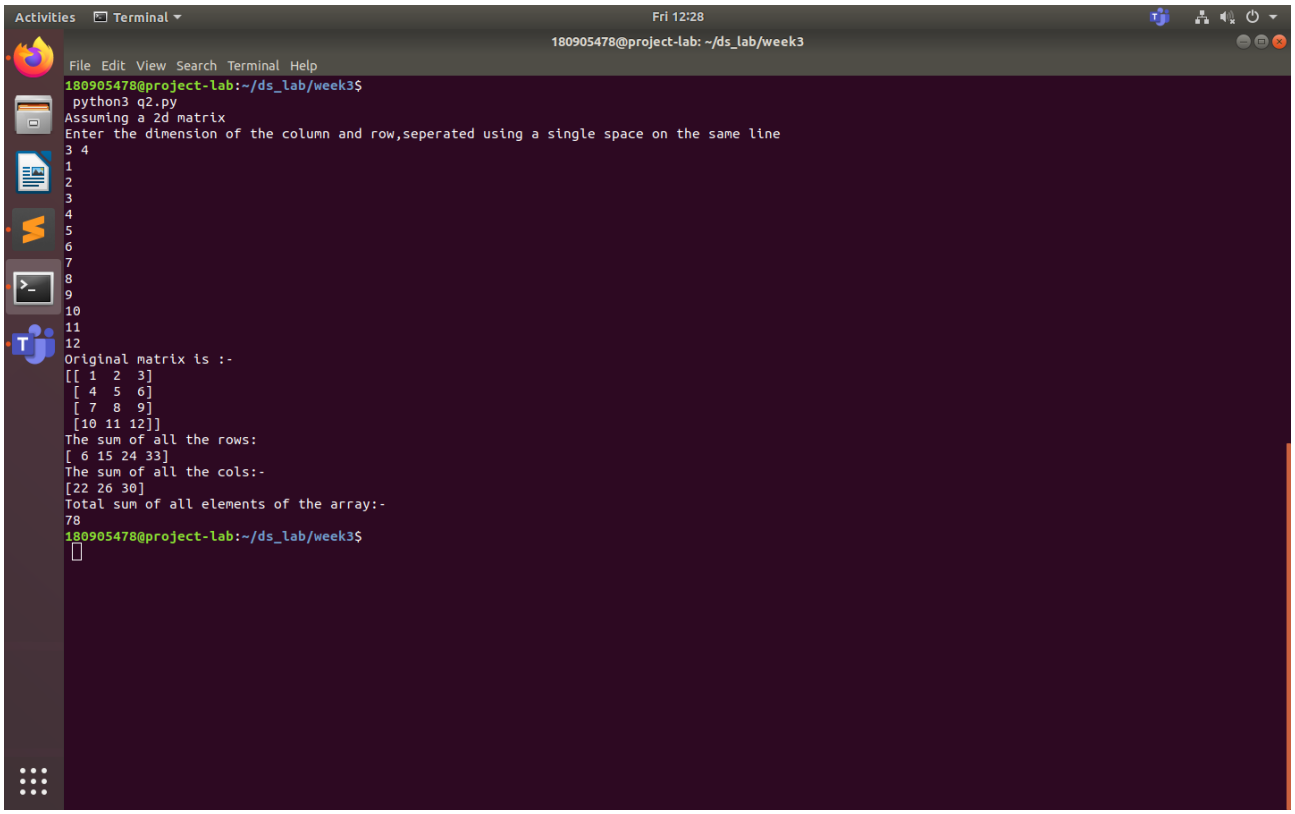
```
'''
Author: Sagnik Chatterjee
Program to find the factos of a given number (input from user)
using for loop.
'''

def factors(n:int):
    list1=[]
    for i in range(2,n):
        if n%i==0:
            list1.append(i)
    return list1

def main():
    n=int(input("Enter the number: "))
    if n >=1:
        print("Factors of the number are: ")
        print(factors(n))

    else:
        print("Please enter a positive natural number for checking tis factors.")

if __name__=='__main__':
        main()
```



Q2

```
'''
Author : Sagnik Chatterjee
Program to find the sum of columns and rows using axis.

'''

import numpy as np


def main():
        print("Assuming a 2d matrix")
        print('Enter the dimension of the column and row,seperated using a single space on the same line')
        n,m=map(int,input().split())
        arr=[[int(input()) for x in range(n)] for y in range(m)]
        numpy_arr=np.array(arr)

        #original matrix
        print("Original matrix is :- ")
        print(numpy_arr)

        ##summing along all the rows
        print("The sum of all the rows:")
        print(np.sum(numpy_arr,axis=1))

        #summing along all the cols
```

```
        print("The sum of all the cols:-")
        print(np.sum(numpy_arr,axis=0))

        #sum of matrix
        print("Total sum of all elements of the array:- ")
        print(np.sum(numpy_arr))

if __name__=='__main__':
        main()
```

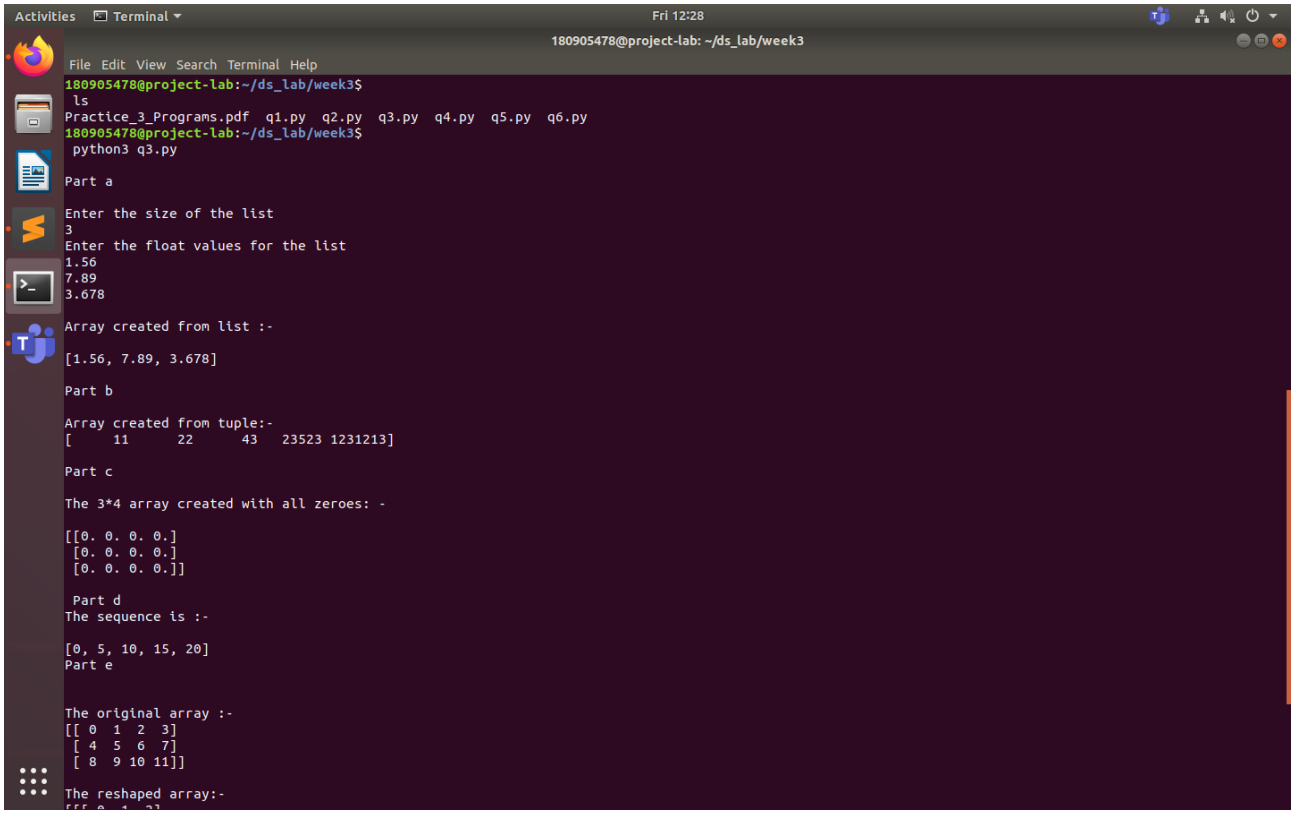


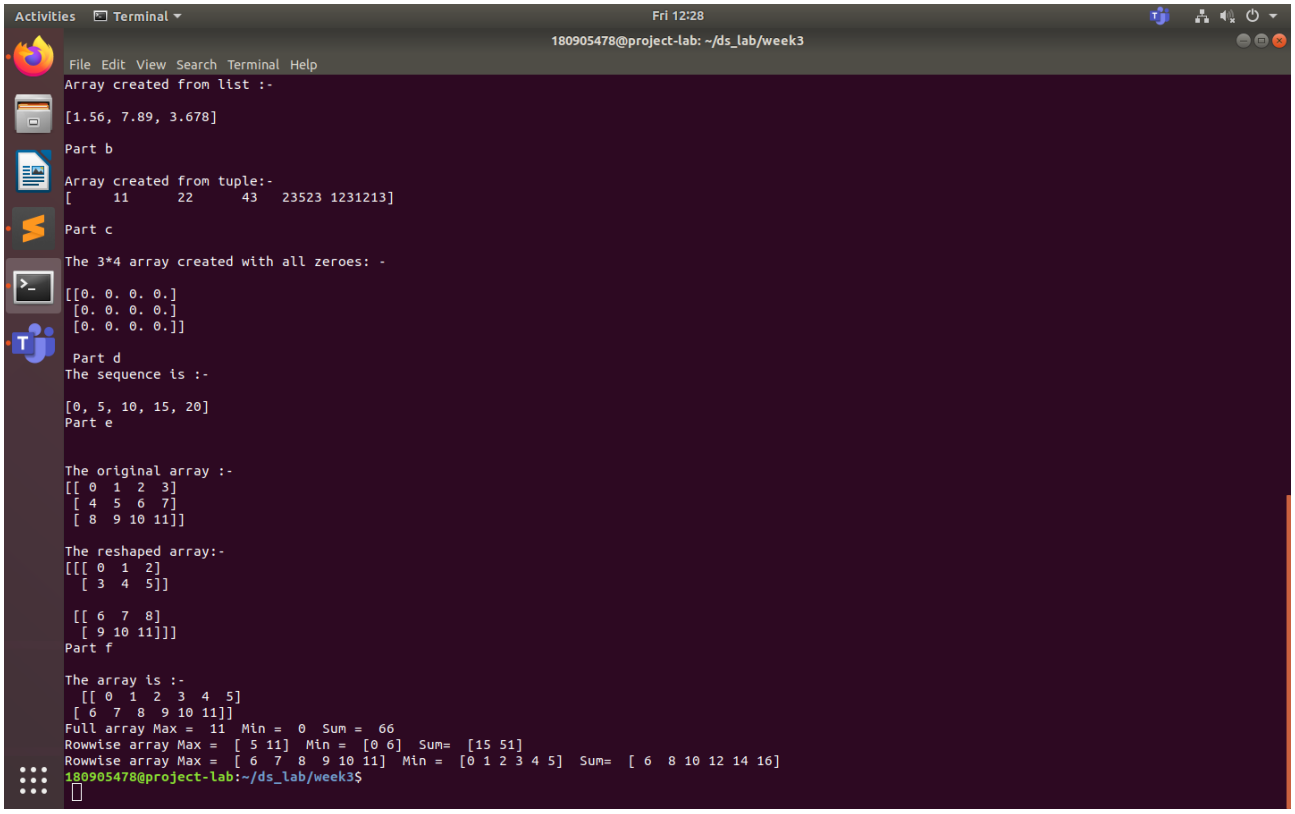
Q3

```
'''
Author: Sagnik Chatterjee
Operations on arrays
'''
import numpy as np

#part a : Create array from list with type float

print('\nPart a')
print("\nEnter the size of the list ")
p=int(input())
arr1=[]
print('Enter the float values for the list')
while p>0:
        q=float(input())
```

```
        arr1.append(q)
        p=p-1
arr =np.array(arr1)
print("\nArray created from list :- \n")
print(arr1)

#part b: Create array from tuple
print('\nPart b')
tup1=(11,22,43,23523,1231213)
arr2 =np.array(tup1)
print("\nArray created from tuple:- ")
print(arr2)

#part c:- Creating a 3*4 array with all zeroes
print('\nPart c ')
a=(3,4)
print("\nThe 3*4 array created with all zeroes: - \n")
print(np.zeros(a))

#part d:- Creae a sequence of integers fom 0 to 20 with steps of 5
print('\n Part d ')
temp=0
sequence=[]
while temp<=20:
        sequence.append(temp)
        temp+=5

print('The sequence is :- \n')
print(sequence)


#part e:- Reshape 3X4 array to 2X2X3 array
print('Part e\n')
arr3=np.arange(12).reshape(3,4)
print("\nThe original array :- ")
print(arr3)
print('\nThe reshaped array:- ')
print(arr3.reshape(2,2,3))


''' part 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())
'''
print('Part f\n')
a=np.arange(12).reshape(2,6)
print('The array is :-\n ',a)
print("Full array Max = " , a.max() , " Min = " , a.min() , " Sum = " , a.sum())
print("Rowwise array Max = " , a.max(axis = 1) , " Min = " , a.min(axis = 1) , " Sum= " ,
a.sum(axis = 1))
```

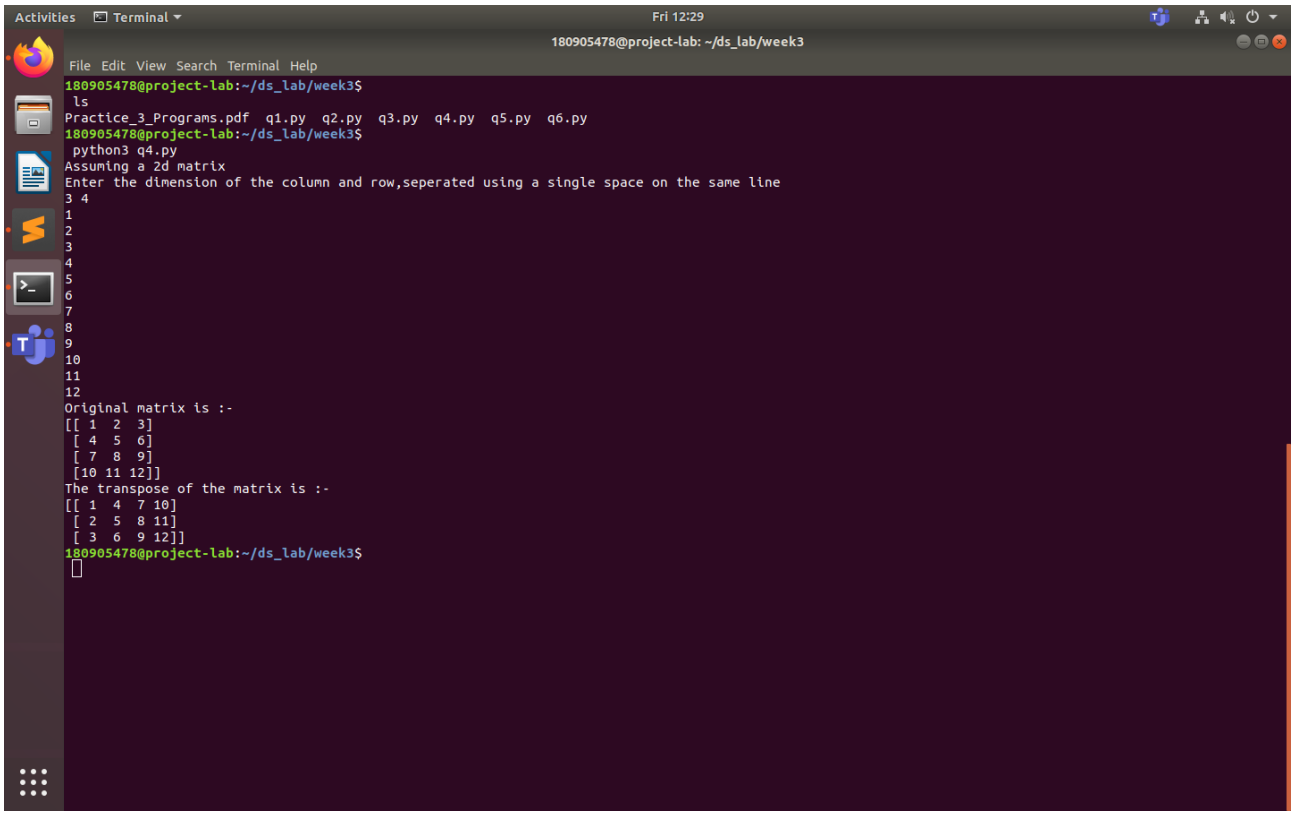
print("Rowwise array Max = " , a.max(axis = 0) , " Min = " , a.min(axis = 0) , " Sum= " , a.sum(axis = 0))





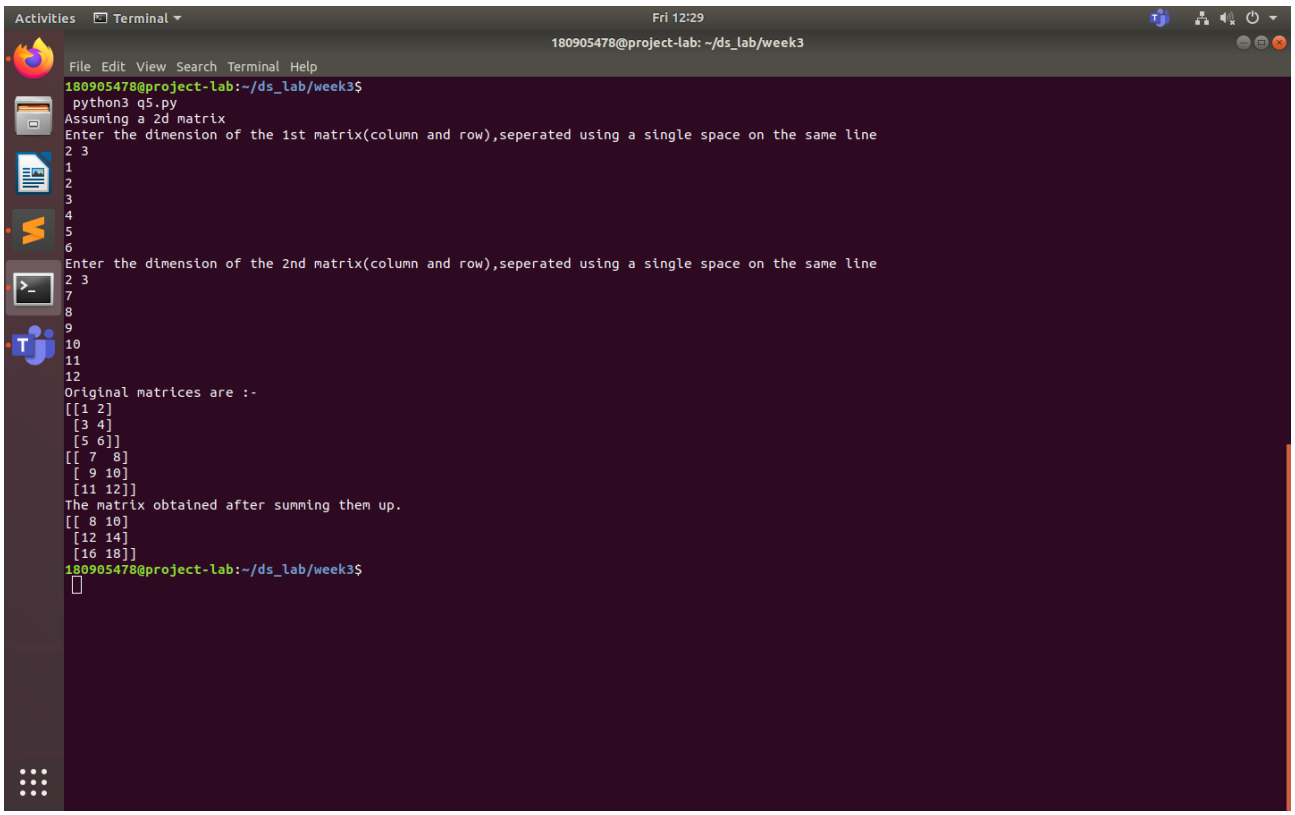
Q4
'''
Author : Sagnik Chatterjee
Program to transpose a given matrix.
'''

```
import numpy as np


def main():
        print("Assuming a 2d matrix")
        print('Enter the dimension of the column and row,seperated using a single space on the same line')
        n,m=map(int,input().split())
        arr=[[int(input()) for x in range(n)] for y in range(m)]
        numpy_arr=np.array(arr)

        #original matrix
        print("Original matrix is :- ")
        print(numpy_arr)

        # the transpose of the matrix is
        print("The transpose of the matrix is :-")
        print(numpy_arr.transpose())

if __name__=='__main__':
        main()
```



Q5

```
'''
Author : Sagnik Chatterjee
Program to add two matrices.
'''

import numpy as np


def main():
	try:
		print("Assuming a 2d matrix")
		print('Enter the dimension of the 1st matrix(column and row),seperated using a single space on the same line')
		n,m=map(int,input().split())
		arr1=[[int(input()) for x in range(n)] for y in range(m)]

		print('Enter the dimension of the 2nd matrix(column and row),seperated using a single space on the same line')
		a,b=map(int,input().split())
		arr2=[[int(input()) for x in range(a)] for y in range(b)]

		if(n!=a or m!=b ):
			raise Exception("Dimension do not match for the two matrices;please check your values.")

		numpy_arr1=np.array(arr1)
		numpy_arr2=np.array(arr2)

		#original matrix
		print("Original matrices are :- ")
		print(numpy_arr1)
		print(numpy_arr2)


		## the matrix after summing them up
		print("The matrix obtained after summing them up.")
		result_matrix= np.add(numpy_arr1,numpy_arr2)

		print(result_matrix)

	except Exception as e:
		print(e)

if __name__=='__main__':
	main()
```



Q6

```
'''
Author : Sagnik Chatterjee
Program to find element wise product between 2 matrices
'''

import numpy as np


def main():
        try:
                print("Assuming a 2d matrix")
                print('Enter the dimension of the 1st matrix(column and row),seperated using
a single space on the same line')
                n,m=map(int,input().split())
                arr1=[[int(input()) for x in range(n)] for y in range(m)]

                print('Enter the dimension of the 2nd matrix(column and row),seperated
using a single space on the same line')
                a,b=map(int,input().split())
                arr2=[[int(input()) for x in range(a)] for y in range(b)]

                if(n!=a or m!=b ):
                        raise Exception("Dimension do not match for the two matrices;please
check your values.")

                numpy_arr1=np.array(arr1)
```

```
            numpy_arr2=np.array(arr2)

            ##the original matrices
            print("The first matrix is :- ")
            print(numpy_arr1)
            print("The second matrix is :- ")
            print(numpy_arr2)

            ## the element wise product of the 2 matrices
            print("The element wise product of the two matrices are:- ")
            print(np.multiply(numpy_arr1,numpy_arr2))

        except Exception as e:
            print(e)


if __name__=='__main__':
        main()
```