**Name : Aman Jakhetiya**

**Enroll : 9918103209**

**Batch : F8**

**Week 6**

**Github Link: <https://github.com/Amanjakhetiya/OSS_lab_6>**

**Q1.**

import numpy as np

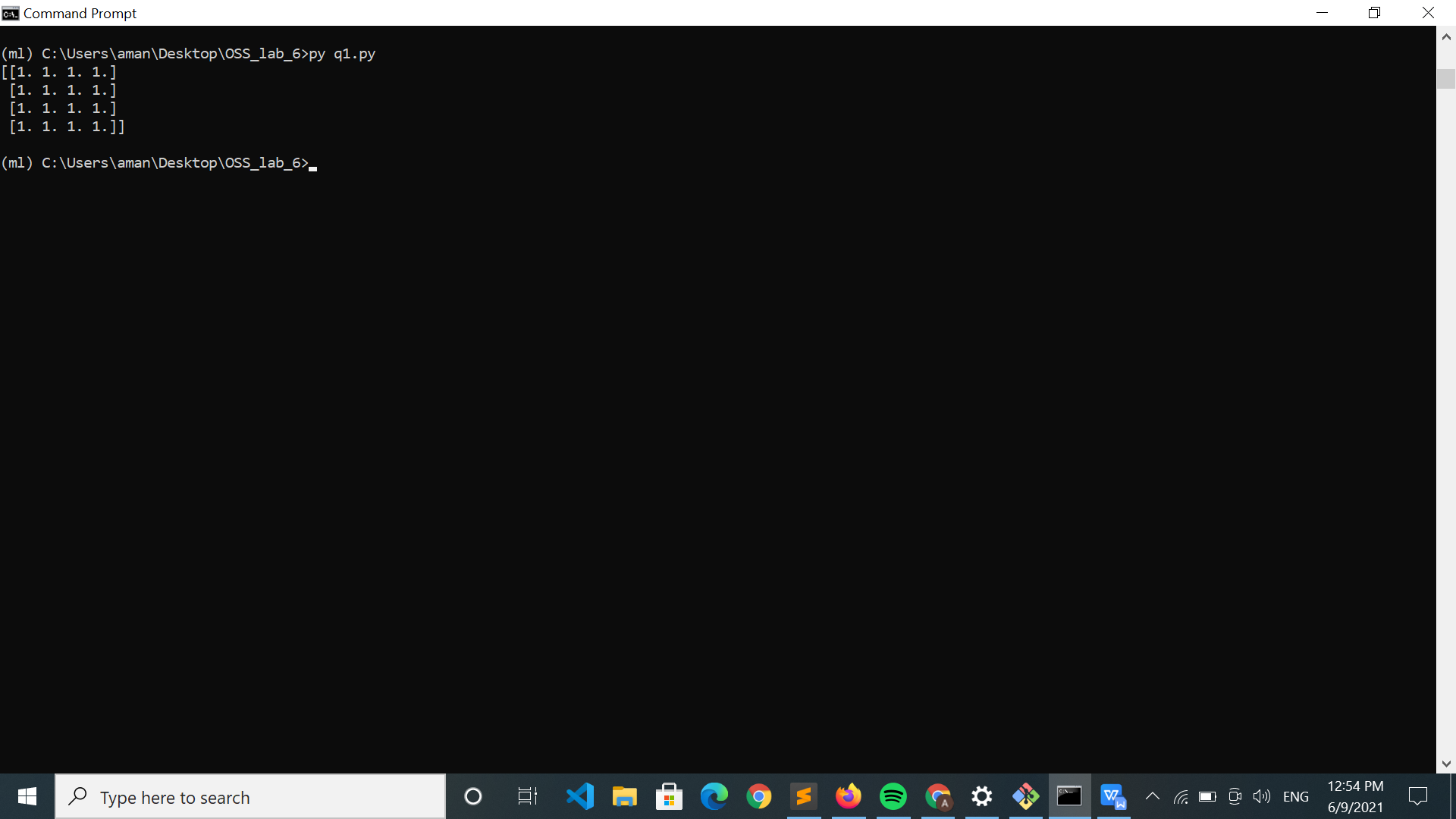
from scipy import io

a = np.ones((4,4))

io.savemat('test.txt',{'ar':a})

data = io.loadmat('test.txt',struct\_as\_record=True)

print(data['ar'])

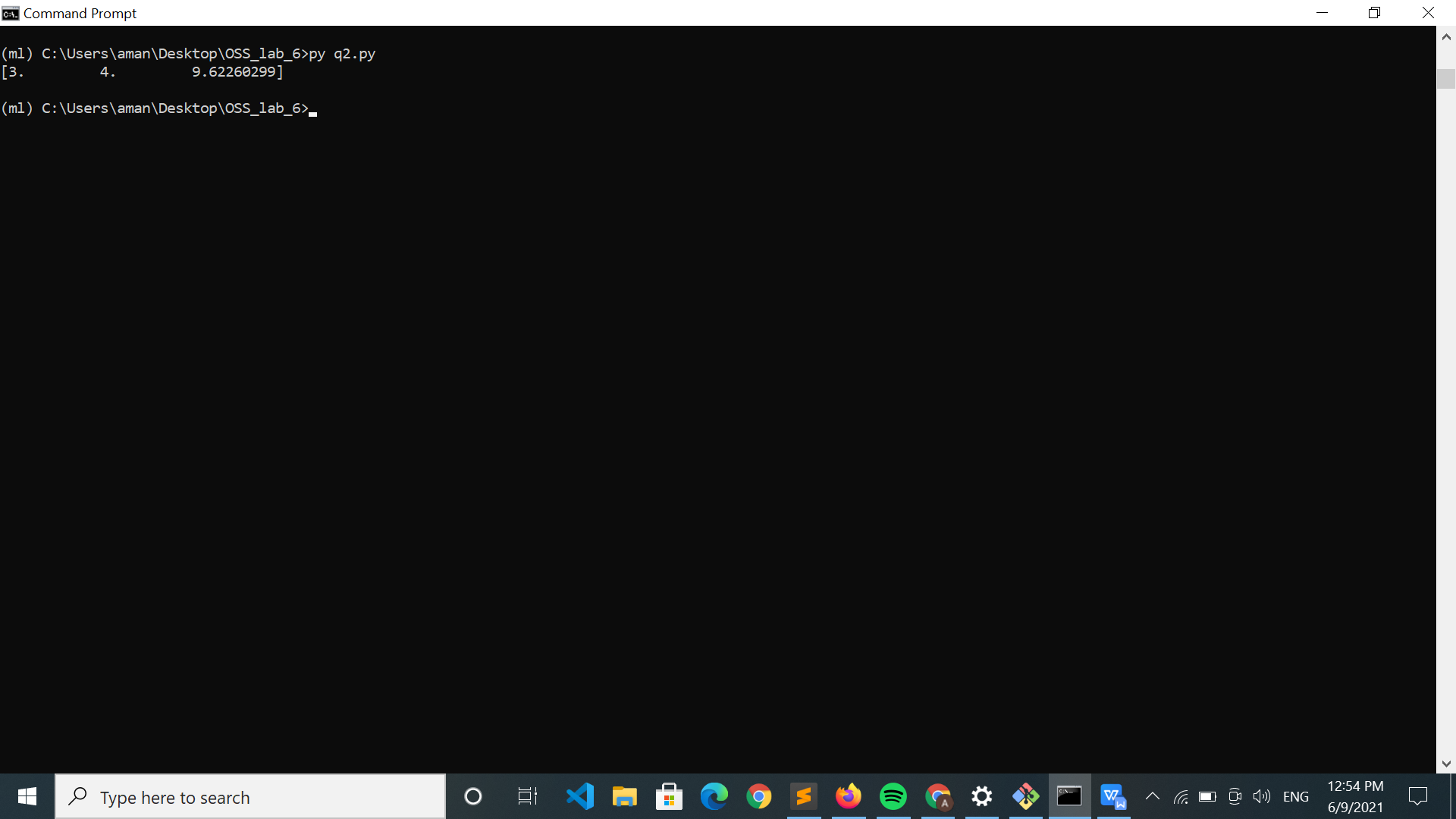


**Q2.**

from scipy.special import cbrt

a = [27,64,891]

print(cbrt(a))



**Q3.**

import numpy as np

a = np.array([[4,5],[3,2]])

b=a.copy()

from scipy.linalg import det

#determinant

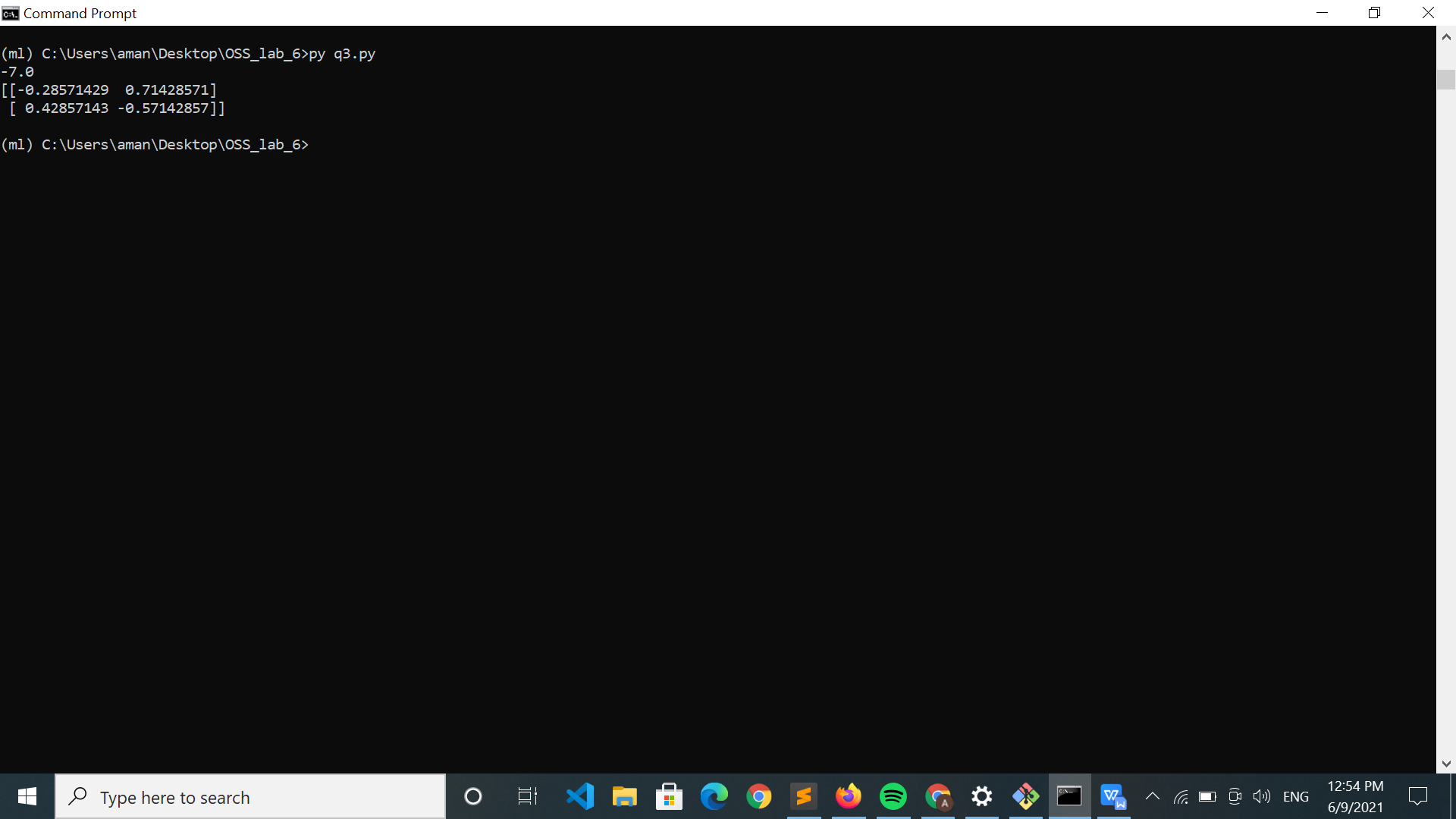
print(det(a))

#inverse

from scipy.linalg import inv

print(inv(b))

# print(a.ndim,a.shape)



**Q4.**

from scipy.linalg import eig

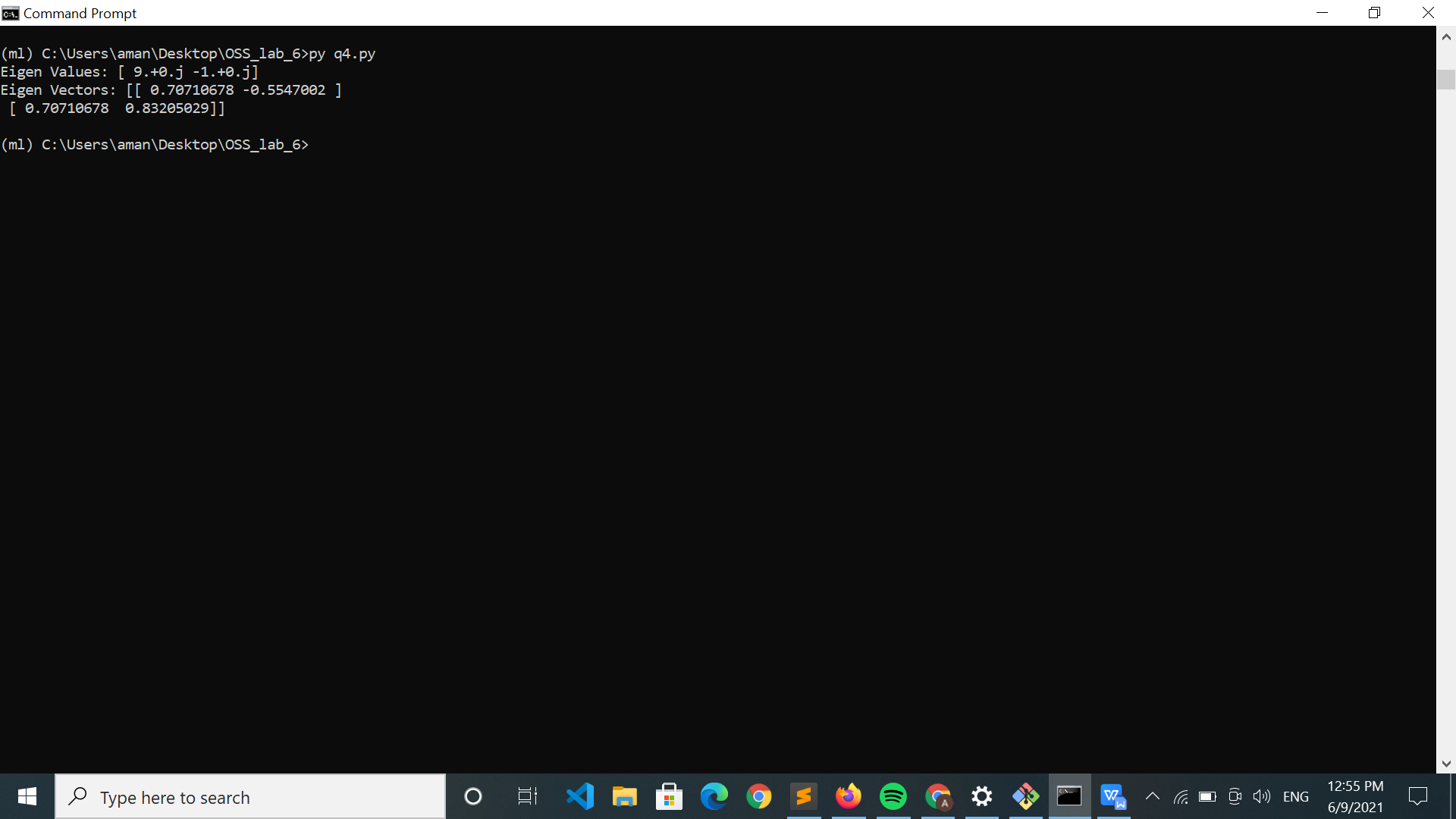
import numpy as np

a = np.array([[5,4],[6,3]])

eigen\_values,eigen\_vector = eig(a)

print("Eigen Values:",eigen\_values,end='\n')

print("Eigen Vectors:",eigen\_vector,end='\n')



**Q5.**

from scipy.sparse import random,hstack,vstack,tril,triu,issparse

S1 = random(3,4,density=0.75)

S2 = random(3,4,density=0.25)

print(S1.A,end='\n\n')

print(S2.A,end='\n\n')

# horizonatal stack

print(hstack([S1,S2]).toarray(),end='\n\n')

# vertical stack

print(vstack([S1,S2]).toarray(),end='\n\n')

# lower triangular portion

print(tril(S1).toarray(),end='\n\n')

# upper triangular portion

print(triu(S1).toarray(),end='\n\n')

# sparse matrix check

print(issparse(S1)," ",issparse(S2))

