Q3.A)

import numpy as np

a=np.array([[1,2],[3,4],[5,6]])

b=np.array([[3,4],[1,2],[7,8]])

c=np.array([[7,8],[1,2],[4,5]])

ab=np.multiply(a,b)

print("AB")

print(ab)

m=np.square(b)

print("B2")

print(m)

f=np.multiply(4,m)

print("4B2")

print(f)

n=np.divide(c,4)

print("C/4")

print(n)

r=np.add(ab,f)

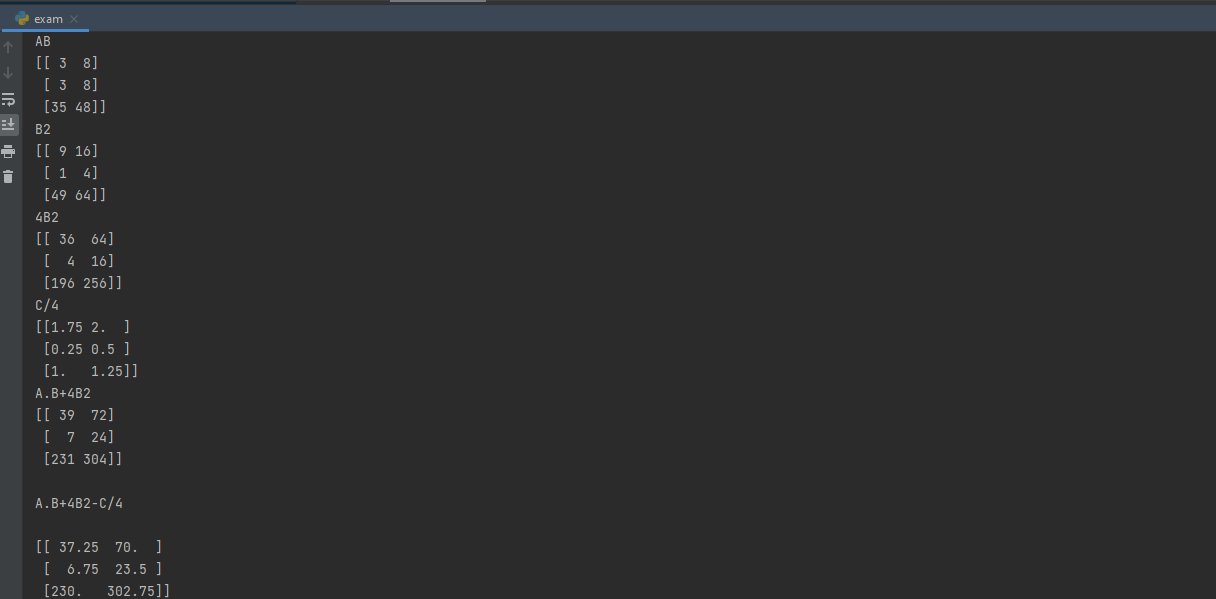
print("A.B+4B2")

print(r)

t=np.subtract(r,n)

print("\nA.B+4B2-C/4\n")

print(t)



B)

import dataset

import numpy as np

import matplotlib.pyplot as mlt

#x=dataset.iloc[:,-1]

#y=dataset.iloc[:,1]

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

x=np.array([[1,2],[3,4],[5,6]])

y=np.array([[3,4],[1,2],[7,8]])

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

x\_text,x\_train,y\_test,y\_train= train\_test\_split(x,y,test\_size=1/3,random\_state=3)

model=LinearRegression()

model.fit(x,y)

y\_predict=model.predict(x\_text)

y\_pred=model.predict(y\_test)

mlt.scatter(x\_train,y\_train,color="red")

mlt.plot(x\_text,y\_predict,color="blue")

mlt.scatter(x\_text,y\_test,color="red")

mlt.plot(x\_text,y\_predict,color="blue")

mlt.title("Regression")

mlt.xlabel("x\_label")

mlt.ylabel("y\_label")

mlt.show()

mae=mean\_absolute\_error(x,y)

mea=mean\_squared\_error(x,y)

print("absolute error",mea)

print("squared error",mae)

