**#Problem # 1**import numpy as np

def mysum(x):

    num = 0

    for n in x:

        num = num + n

    return num

def mymax(x):

    maxNumber = 1

#Yourcodegoesfromheretocalculateandreturnthemaximumvalue

    for a in x:

        if a > maxNumber:

            maxNumber = a

    return maxNumber

def mymin(x):

    minNumber = 1

#Yourcodegoesfromheretocalculateandreturntheminimumvalue

    for a in x:

            if a < minNumber:

                minNumber = a

    return minNumber

# x = [0,1,2,3,4,5,6,7,8,9]

x = range(10)

#   expected   result:sum= 45   max=9   min= 0

print("sum=",mysum(x)," max=",mymax(x),"    min=",mymin(x))# x =[1,3,5,7,9]

x = range(1,10,2)

#  expected  result:sum=25  max=9   min=1

print("sum=",mysum(x)," max=",mymax(x),"    min=",mymin(x))

# x = 100 random numbers, rand() is a function to generate random numbers from

#  standard uniform distribution. Use  help(rand) or ?rand or rand? to see  help page  on   #randfunctioninCanopy.Rememberkeyinqandpressreturntoquithelppage.

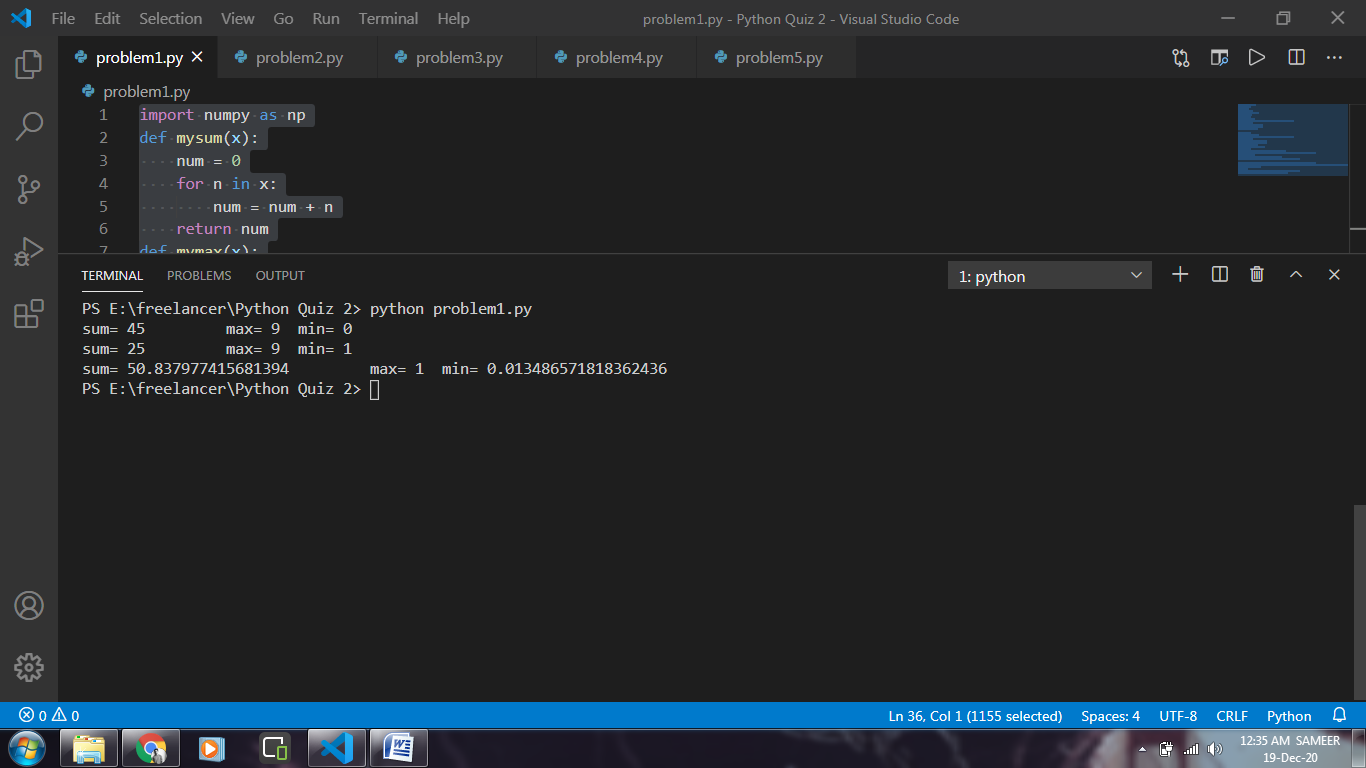
x = np.random.rand(100)

# print(x)

print("sum=",mysum(x)," max=",mymax(x),"    min=",mymin(x))

# expected result: depending on the random values.

**Output:**

****

#Problem 2

import numpy as np

import math

def getNorm(x):

    #Yourcodegoesfromheretocalculateandreturnthenorm

    num = 0

    for i in x:

        num = (i\*i) + num

    norm = math.sqrt(num)

    return norm

def getInnerProd(x, y):

# Your code goes from here to calculate and return the inner product

    return np.dot(x, y)

def getDist(x, y):

    # Your code goes from here to calculate and return the distance

    distance = math.sqrt( ((x[0]-y[0])\*\*2)+((x[1]-y[1])\*\*2) +((x[2]-y[2])\*\*2)+((x[3]-y[3])\*\*2)+((x[4]-y[4])\*\*2) )

    return distance

# xVec= [0,1,2,3,4]

xVec =range(5)

# yVec= [4,3,2,1,0]

yVec=range(4,-1,-1)

#  expected  result:  norm(x)=  5.4772  norm(y) =5.4772

print("norm(x)=",getNorm(xVec),"norm(y) =",getNorm(yVec))

#expectedresult:innerprod(x,y)=10

print("innerprod(x,y) =" ,  getInnerProd(xVec,   yVec))

#expected result: distance(x,y) = 6.3246

# print("distance(x,y) =", getDist(xVec,yVec))

getDist(xVec,yVec)

# Generate  two  random  vectors  of  length  10  and  calculate  their  distance.  # Your code goes from here. Use rand() to generate randomvectors.

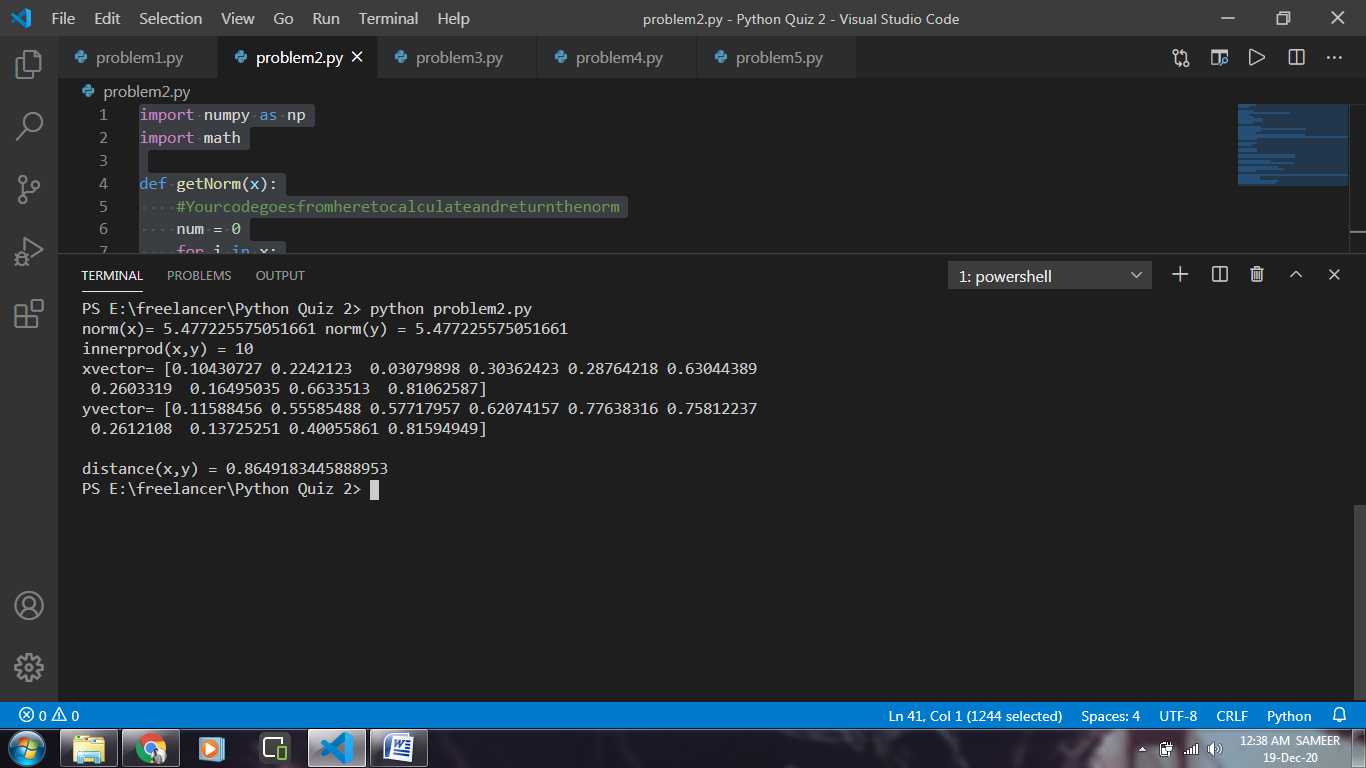
x = np.random.rand(10)

y = np.random.rand(10)

print("xvector=", x,"\nyvector=",y,"\n")

print("distance(x,y) =", getDist(x,y))

**Output:**

****

#Problem 3

def countVotes(votes):

    #Yourcodegoesfromheretocounttheinputvotesandoutputvotingresult

    list1 = votes.split(',')

    count = 0

    reject = 0

    for x in list1:

        if (x.strip(' ') == 'y') or (x.strip(' ') == 'Y'):

            count = count + 1

        elif (x.strip(' ') == 'n') or (x.strip(' ') == 'N'):

            reject = reject + 1

    print(f"Reject: 20 votes in total, {count} accept and {reject} reject")

votes = "N ,Y, Y,N,n, N , N , N ,n ,y, n,N,Y, y,Y,N, N , n ,y, N"

# the function call below should display the message

# Reject: 20 votes in total, 7 accept and 13 reject

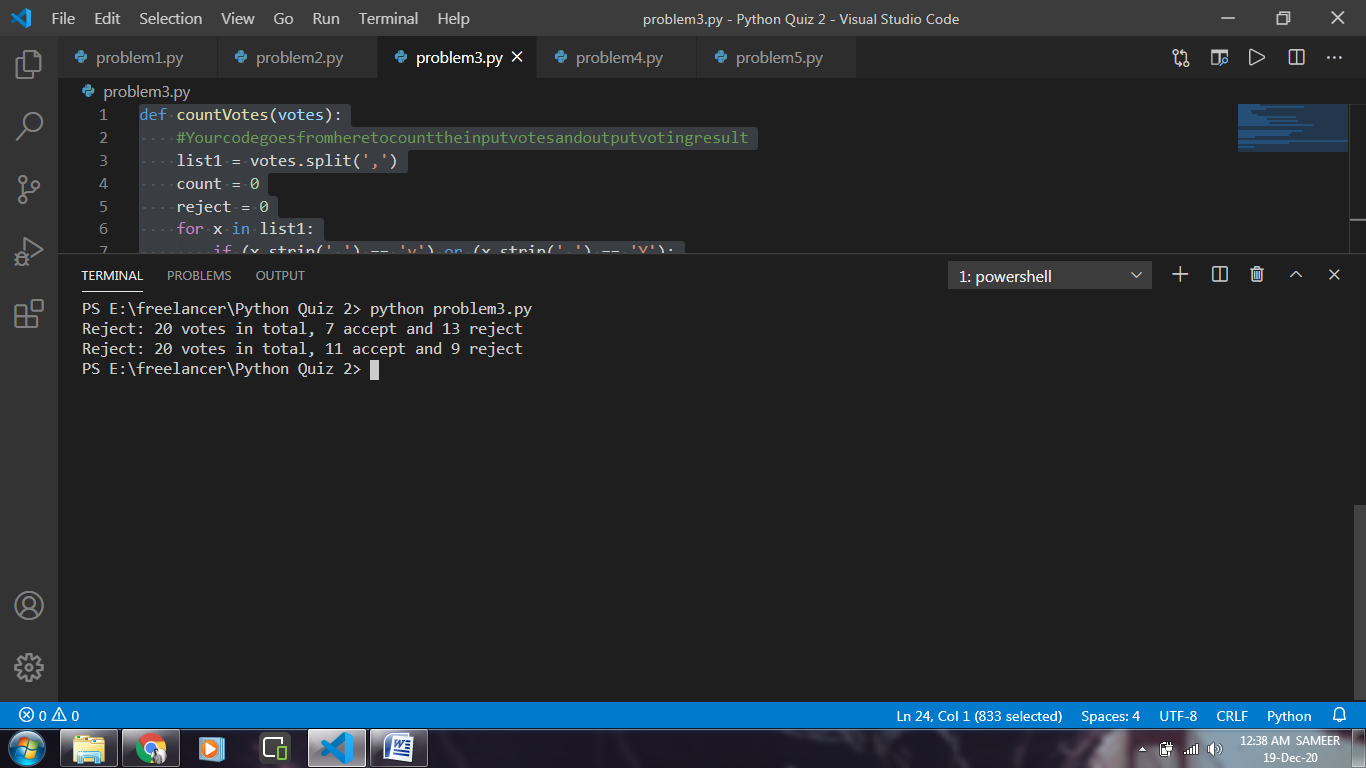
countVotes(votes)

votes =  "y,n,Y,y,y ,n, N,  Y,N  , N,y, n,  Y,Y,y,  N  ,y, n,  n  , Y  "  #thefunctioncallbelowshoulddisplaythemessage

# Accept: 20 votes in total, 11 accept and 9 reject

countVotes(votes)

**Output:**

****

#Problem 4

def capitalise(phrase):

    lowerWordList = ["a", "an", "the", "am", "is", "are", "and", "of", "in", "on", "with", "from", "to"]

    arr = ""

    #writeyourcodegoesfromheretoperformwordcapitalisation

    wordsList = phrase.split(" ")

#     print(wordsList)

    for word in wordsList:

        if word in lowerWordList:

            w = word.lower()

            arr = arr + w + " "

        else:

            w = word[0: 1].upper()

            wl = word[1:].lower()

            arr = arr + w + wl + " "

    print(arr)

# Output: "I am an Educator and a Researcher"

capitalise("I am an educator and a researcher")

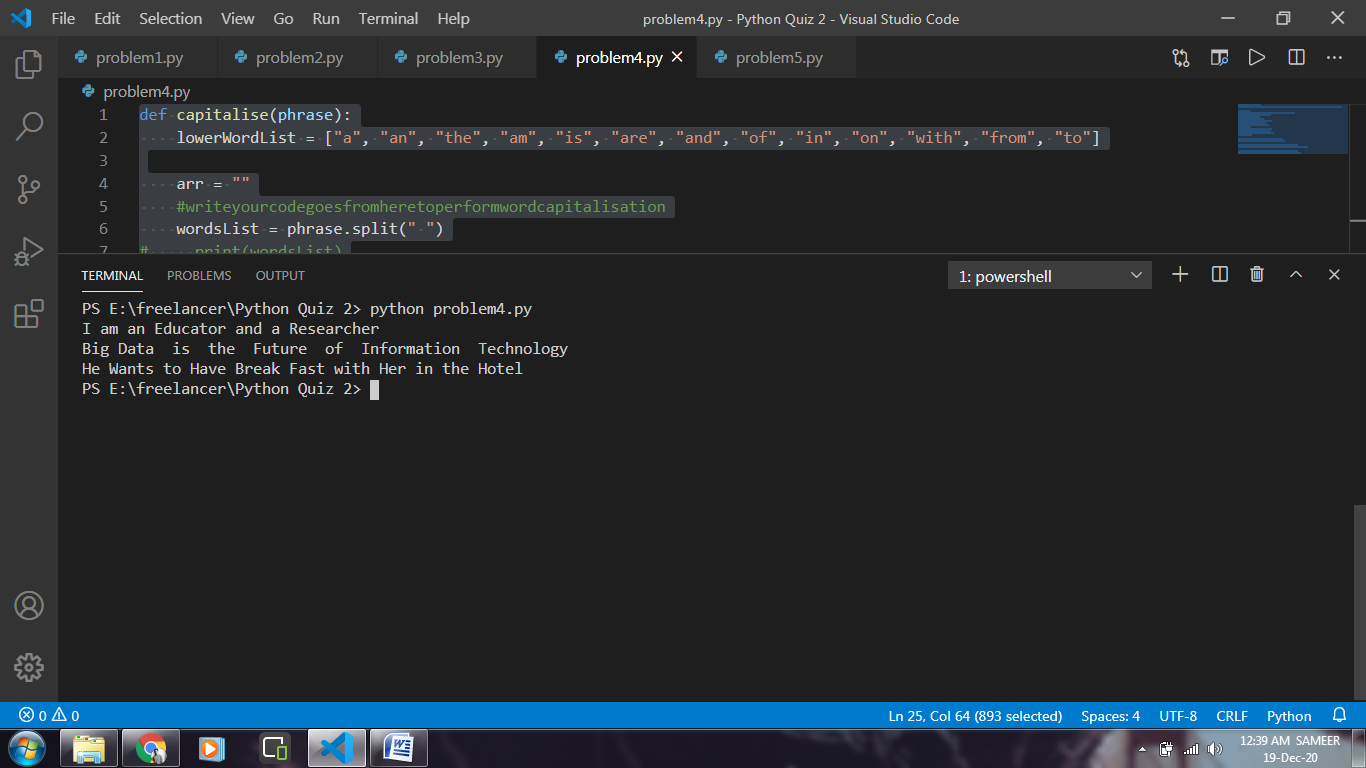
# Output: "Big Data is the Future of Information Technology"

capitalise("big data  is  the  future  of  information  technology")

# Output: "He Wants to Have Breakfast with Her in the Hotel"

capitalise("He wants to have break fast with her in the hotel")

**Output:**

****

#Problem 5

# itemgetter is used by the sorted() system function

from operator import itemgetter

def createDict(filename):

    # Your code goes from here to create and return a dictionary

    with open(filename ,'r') as f:

        para = f.read()

    words = para.split()

    counts = {}

    for word in words:

        if word in counts:

            counts[word] += 1

        else:

            counts[word] = 1

    return counts

myDict =createDict("bigdata.txt")

# sort the items of the dictionary by descending order of the second entry of each pair (i.e.value in the key-valuepair)

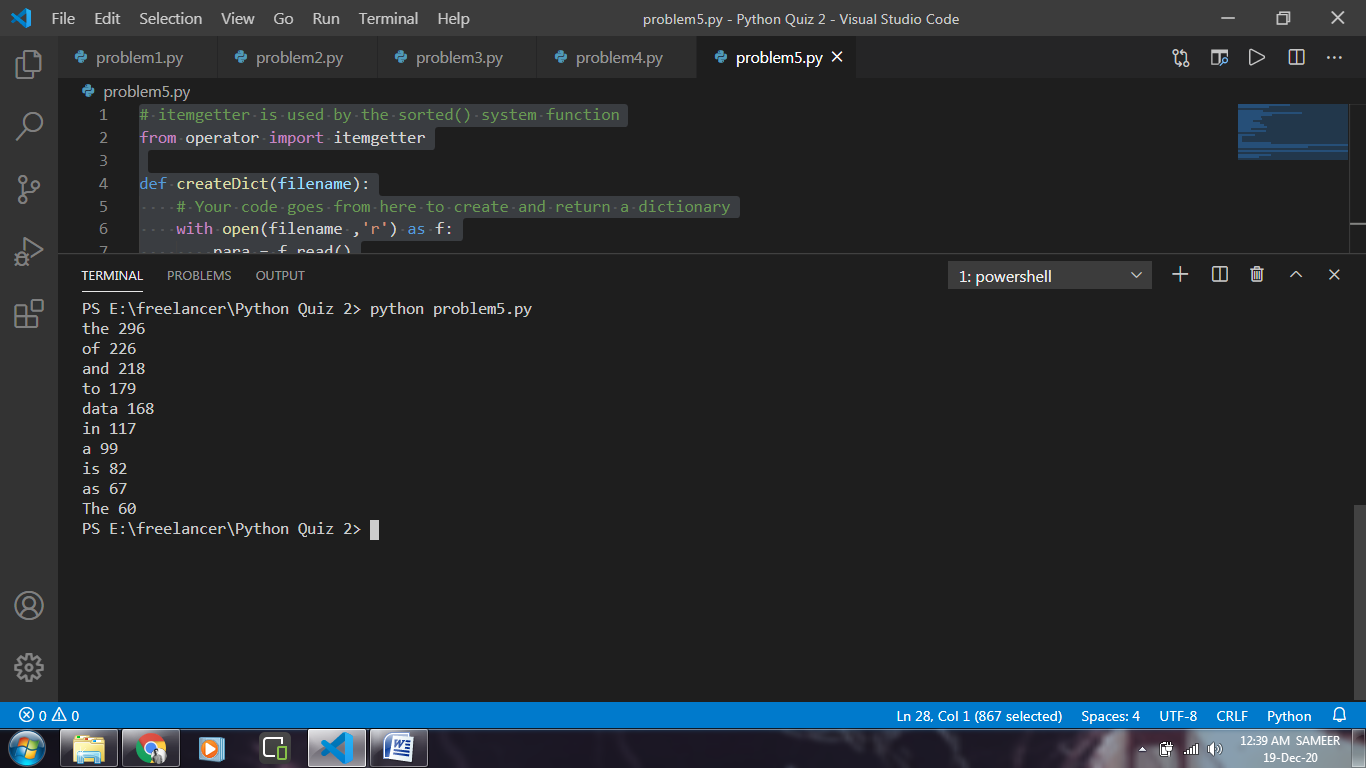
# note that the return value is a list which is assigned to sortedList

sortedList  =  sorted(myDict.items(),  key=itemgetter(1),  reverse  =  True) # print the top 10 entries in the sortedlist

for key,value in sortedList[:10]:

    print(key, value)

**Output:**

****