Python Lab Assignment 4

Name: Shravani V. Patwardhan

MIS: 1122315179(G4)

Q.1 Write a program to count the numbers of characters in the given string and store them in a dictionary data structure.

Code:

string = input("Enter a string:")

dict = {}

for char in string:

    if char not in dict:

        dict[char]=1

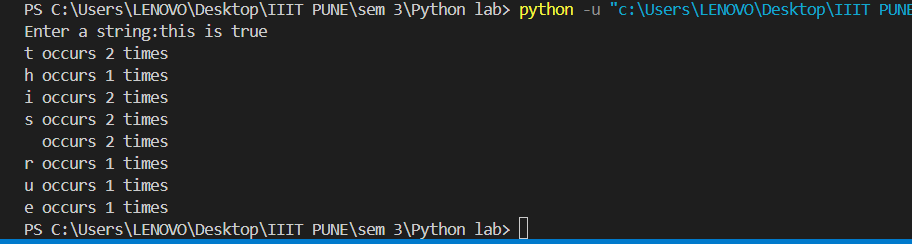
    else:

        dict[char]=dict[char]+1

for key,value in dict.items():

    print(f"{key} occurs {value} times")

Output:



Q.2 Write a program to use split and join methods in the given string and trace a birthday with a dictionary data structure.

Ans. dict = {"Shrav": [20,'Aug',2004],

        "Shre": [10,'Aug',2004],

        "Mok": [14,'June',2005],

        "Musk": [2,'April',2004]

        }

inp\_name = input("Enter the name whose bday should be searched:").strip()

if inp\_name in dict:

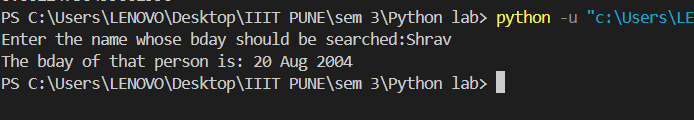
    date = " ".join(map(str,dict[inp\_name]))

    print("The bday of that person is:",date)

else:

    print("This person's bday is unknown")

Output:



Q.3 Write function to compute gcd and lcm of two numbers.

Ans. #to find gcd and lcm of two numbers

a = int(input("Enter the first number:"))

b = int(input("Enter the second number:"))

def gcd(a,b):

    for i in range(min(a,b),0,-1):

        if(a%i==0 and b%i==0):

            return i

    return 1

def lcm(a,b):

    for i in range(max(a,b), a\*b + 1):

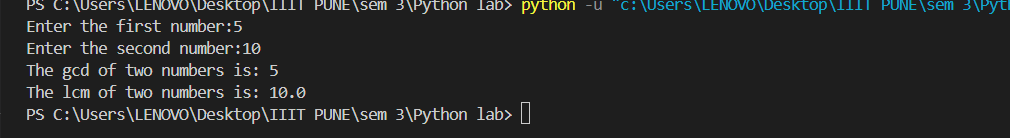
        if(i%a==0 and i%b==0):

            return i

print("The gcd of two numbers is:",gcd(a,b))

print("The lcm of two numbers is:",lcm(a,b))

Output:



Q.4 Write a function ball \_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. Represent a ball on a plane as a tuple of (x, y, r), r being the radius. If (distance between two balls centers) &if(sum of their radii) then (they are colliding)

Ans. import math

def parse\_input(input\_str):

    return tuple(map(float, input\_str.split()))

a = input("Enter the x, y coordinate and radius of 1st ball (separated by spaces): ")

b = input("Enter the x, y coordinate and radius of 2nd ball (separated by spaces): ")

a = parse\_input(a)

b = parse\_input(b)

def collision(a, b):

    distance = math.sqrt((b[0] - a[0]) \*\* 2 + (b[1] - a[1]) \*\* 2)

    radiisum = a[2] + b[2]

    if distance <= radiisum:

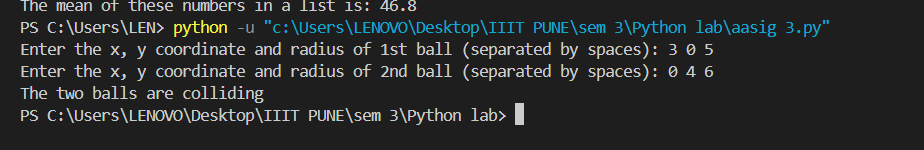
        print("The two balls are colliding")

    else:

        print("No, the two balls are not colliding")

collision(a, b)

Output:



Q.5 Find mean, median, mode for the given set of numbers in a list.

Code:

lis1 = [10,23,45,89,67]

sum=0

max=0

mean=0

dict={}

for i in range(0,len(lis1)):

       sum+=lis1[i]

       mean=sum/len(lis1)

def median(lis1):

   n=len(lis1)

   l1 = sorted(lis1)

   if(n%2==1):

      return (l1[n//2])

   else:

      return (l1[n//2]-1 + l1[n//2])/2

def mode(l1):

    dict={}

    for i in l1:

        if i not in dict:

            dict[i]=1

        else:

            dict[i]+=1

    max=0

    maxnum=0

    for key in dict:

        if dict[key]>max:

            max=dict[key]

            maxnum=key

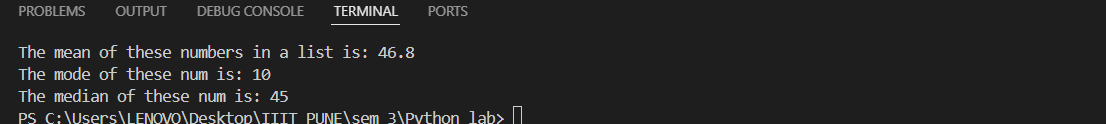
    return maxnum

print("The mean of these numbers in a list is:",mean)

print("The mode of these num is:",mode(lis1))

print("The median of these num is:",median(lis1))

Output:



Q.6 Write a program to implement

a. Bubble sort,

b. Merge sort,

c. Selection sort and

d. Insertion sort.

Code:

list1 = [10, 24, 78, 45, 90, 8]

length = len(list1)

def bubble\_sort(list1):

    n = len(list1)

    for i in range(n):

        for j in range(0, n-i-1):

            if list1[j] > list1[j+1]:

                list1[j], list1[j+1] = list1[j+1], list1[j]

    return list1

def merge\_sort(list1,low,high):

    if low<high:

        mid=(low+high)//2

        merge\_sort(list1,low,mid)

        merge\_sort(list1,mid+1,high)

        merge(list1,low,mid,high)

def merge(list1,low,mid,high):

    n1=mid-low+1

    n2=high-mid

    L=list1[low:mid+1]

    R=list1[mid+1:high+1]

    i,j,k=0,0,low

    while(i<n1 and j<n2):

        if(L[i]<R[j]):

            list1[k]=L[i]

            i+=1

        else:

            list1[k]=R[j]

            j+=1

        k+=1

    while i<n1:

        list1[k]=L[i]

        i+=1

        k+=1

    while j<n2:

        list1[k]=R[j]

        j+=1

        k+=1

def selection\_sort(list1):

    for i in range(len(list1)-1):

        mini=1

        for j in range(i+1,len(list1)):

            if(list1[j]<list1[mini]):

                mini=j

        list1[mini],list1[i]=list1[i],list1[mini]

def insertion\_sort(list1):

    for i in range(1,len(list1)):

        key=list1[i]

        j=i-1

        while j>=0 and list1[j]>key:

            list1[j+1]+=list1[j]

            j-=1

        list1[j+1]=key

def printlist(list1):

    for i in list1:

        print(i,end=" ")

num=int(input("Which sorting do you want to apply?\n1.Bubble\n2.Merge\n3.Selection\n4.Insertion"))

match num:

    case 1:

        bubble\_sort(list1)

    case 2:

        merge\_sort(list1,0,len(list1)-1)

    case 3:

        selection\_sort(list1)

    case 4:

        insertion\_sort(list1)

printlist(list1)

