**PYTHON LAB**

**ASSIGNMENT 3**

NAME: SAVI CHANDWADKAR

MIS NO: 112315169

BRANCH: CSE

GROUP: G4

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

Code:

s = input("Enter a string: ")

freq = {}

for i in s:

    if i in freq:

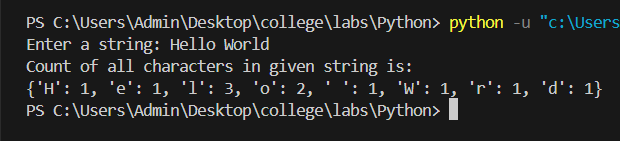
        freq[i]+=1

    else:

        freq[i]=1

print("Count of all characters in given string is: \n" + str(freq))

Output:



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

Code:

dict = {"Savi": [20, "Feb", 2004],

        "Swanandi": [29, "May", 2007],

        "Yajat": [24, "Sept", 2009]}

name = input("Enter name: ")

if name in dict:

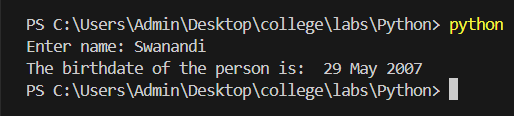
    date = " ".join(map(str,dict[name]))

    print("The birthdate of the person is: ", date)

else:

    print("The person's birthdate is unknown")

Output:



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

Code:

def gcd(a, b):

    if(b==0):

        return a

    return gcd(a, a%b)

def lcm(a, b):

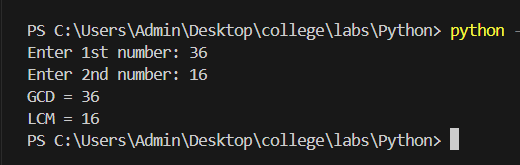
    return (a\*b)//gcd(a,b)

x = int(input("Enter 1st number: "))

y = int(input("Enter 2nd number: "))

print("GCD =", gcd(x,y), "\nLCM =", lcm(x,y))

Output:



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) <= (sum of their radii) then (they are colliding).

Code:

import math

def ball\_collide(b1, b2):

    dist = math.sqrt((b1[0]-b2[0])\*\*2 + (b1[1]-b2[1])\*\*2)

    if(dist<=(b1[2]+b2[2])):

        return True

    else:

        return False

ball1 = input("Enter x, y coordinate and radius of the ball: ")

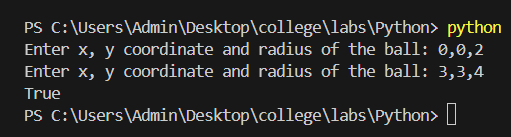
ball2 = input("Enter x, y coordinate and radius of the ball: ")

b1 = tuple(int(x) for x in ball1.split(","))

b2 = tuple(int(x) for x in ball2.split(","))

print(ball\_collide(b1, b2))

Output:



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

Code:

list = []

n = int(input("Enter number of elements: "))

for i in range(0,n):

    x = int(input())

    list.append(x)

s = sum(list)

mean = s/n

list.sort()

if n%2==0:

    median1 = list[n//2]

    median2 = list[n//2 - 1]

    median = (median1+median2)/2

else:

    median = list[n//2]

freq = []

i = 0

while i < n:

    freq.append(list.count(list[i]))

    i += 1

d = dict(zip(list, freq))

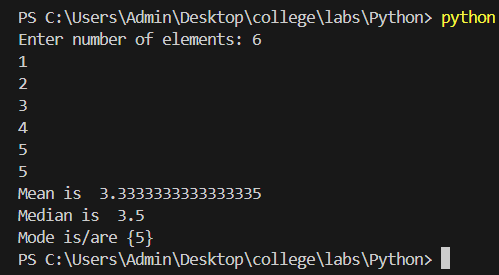
d2 = {k for (k,v) in d.items() if v==max(freq)}

print("Mean is ", mean)

print("Median is ", median)

print("Mode is/are " + str(d2))

Output:



6. Write a program to implement

a. Bubble sort,

b. Merge sort,

c. Selection sort and

d. Insertion sort.

Execute these sorting algorithms using switch case.

Code:

def main():

    print("Menu:\n1.Bubble Sort\n2.Merge Sort\n3.Selection Sort\n4.Insertion Sort")

    num = int(input("Which sorting algorithm would you like to choose? "))

    match num:

        case 1:

            bubblesort()

        case 2:

            mergesort()

        case 3:

            selectionsort()

        case 4:

            insertionsort()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

def errorMessage():

    print("Incorrect input. Please enter a number between 1 and 4.")

def createarr():

    arr = []

    elem = int(input("How many elements you want?\t "))

    for i in range(0, elem):

            arr.append(int(input("Enter number "+str(i+1)+":" )))

def bubblesort():

    newarr = createarr()

    print("Before sorting the array is: ", newarr)

    done = 0

    while(done!=1):

        done = 1

        for i in range(1, len(newarr)):

            if(newarr[i-1]>newarr[i]):

                done = 0

                tmp = newarr[i-1]

                newarr[i-1] = newarr[i]

                newarr[i] = tmp

    print("After sorting array is: ", newarr)

def mergesort():

    newarr=createarr()

    print("Before sorting the array is", newarr)

    print("After sorting the array is", divide(newarr))

def divide(arrToDivide):

    if (len(arrToDivide))<=1:

        return arrToDivide

    middle = len(arrToDivide)/2

    left = arrToDivide[:middle]

    right = arrToDivide[middle:]

    left = divide(left)

    right = divide(right)

    return merge(left,right)

def merge(left,right):

    result = []

    leftID=0

    rightID=0

    while leftID<len(left) and rightID<len(right):

        if left[leftID]<=right[rightID]:

            result.append(left[leftID])

            leftID+=1

        else:

            result.append(right[rightID])

            rightID+=1

    if left:

        result.extend(left[leftID:])

    if right:

        result.extend(right[rightID:])

    return result

def selectionsort():

    newarr=createarr()

    print("Before sorting the array is", newarr)

    for i in range(0,len(newarr)-1):

        m=i

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

            if(newarr[j]<newarr[m]):

                m=j

        temp = newarr[i]

        newarr[i]=newarr[m]

        newarr[m]=temp

    print("After sorting the array is", newarr)

def insertionsort():

    newarr=createarr()

    print ("Before sorting the array is", newarr)

    print ("After sorting the array is", insertion(newarr))

def insertion(iarr):

    for i in range(1,len(iarr)):

        value = iarr[i]

        j=i-1

        while (j>=0) and (iarr[j]>value):

            iarr[j+1] = iarr[j]

            j=j-1

        iarr[j+1] = value

    return iarr