**PYTHON LAB EXPERIMENTS FILE**

* Name : Deepali gupta  
  Sap ID: 500096950
* **EXPERIMENT- 1**Q1. Given an integer n, perform the following conditional actions:  
   ∙ If n is odd, print Weird   
  ∙ If n is even and in the inclusive range of 2 to 5 , print Not Weird   
  ∙ If n is even and in the inclusive range of 6 to 20, print Weird  
   ∙ If n is even and greater than 20, print Not Weird   
  Test cases  
  4-Not Weird   
  18- Weird   
  29- Weird   
  5- Weird  
     
  **CODE:**  
   num=int(input("enter the no. of lopping:"))

for i in range(0,num):

n=int(input("enter an integer:"))  
 if(n%2!=0):

print("weird")

elif(n%2==0):

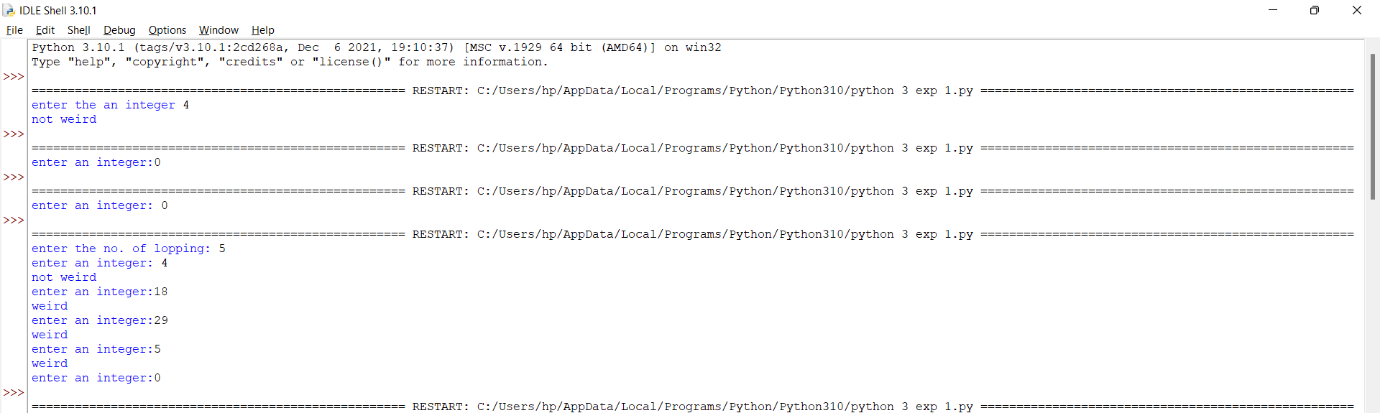
if(n>2 and n<6):

print("not weird")

if(n>6 and n<20):

print("weird")

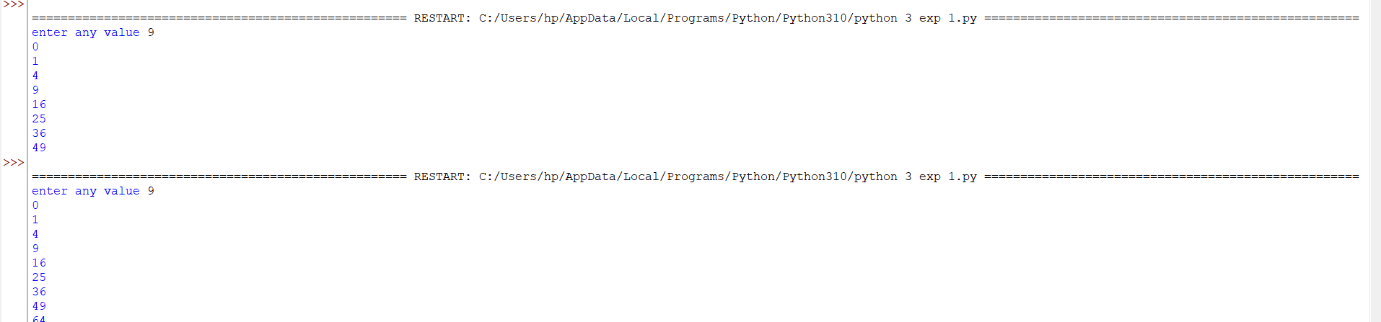
if(n>20):

print("not weird")  
  
  
**OUTPUT:**Q2. WAP to read an integer ‘n’ from STDIN. For all non-negative integers i<n, print i2 on a separate line.   
sample input  
3  
sample output  
0   
1  
4  
 Example :  
The list of non-negative integers that are less than 3 is [0, 1, 2]. The squares of each number is given below: 0 1 4

**CODE:** num=int(input("enter the no of lopping:"))

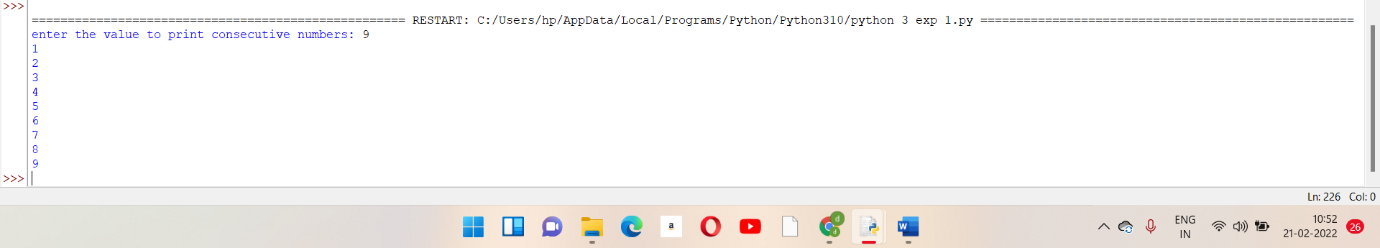
for i in range(0,num):  
 n=int(input("enter any value:"))

for i in range(0,n):

print(i\*i)  
  
**OUTPUT:  
**

Q3. WAP to read an integer from STDIN. Without using any string methods, print the following on a single line: 123…n .  
n=5 Output- 12345  
  
**CODE:** n=int(input("enter the value to print consecutive numbers:"))

for i in range(1,n+1):

print(i)  
  
  
  
**OUTPUT:  
**

**EXPERIMENT -5&6**Q1. Using functions, re-write and execute Python program to:   
1. Add natural numbers upto n where n is taken as an input from user.   
2. Print Fibonacci series till nth term (Take input from user). **CODE 1.** def add():

n=int(input("enter the value of n:"))

sum=0

for i in range(1,n+1):

sum=sum+i

i=i+1  
 return(sum)

a=add()

print(a)  
  
USING FUNCTIONS WITH RECURSION   
  
def recur\_sum(n):

if(n<=1):

return n

else :

return n+recur\_sum(n-1)

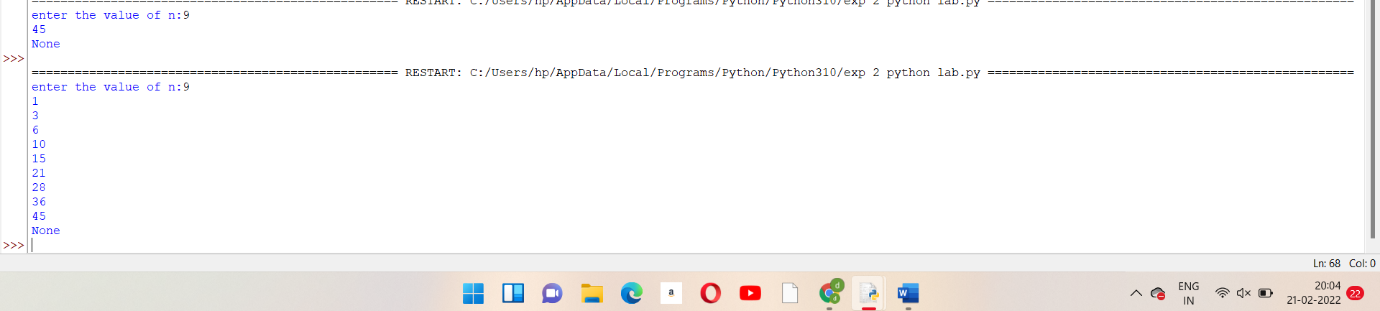
num=int(input("enter n:"))

if(num<0):

print("enter a positive number")

else:

print("sum of numbers :",recur\_sum(num))

**OUTPUT:  
**

**CODE 2**.  
 def fibo(n):

a=0

b=1

for i in range(1,n+1):

third=a+b

a=b

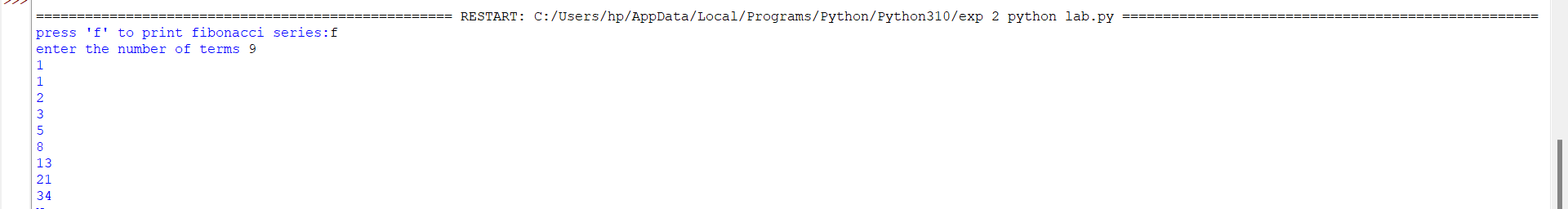
b=third

print(a)

fib=str(input("press 'f' to print fibonacci series:"))

if(fib=='f'):

num=int(input("enter the number of terms"))  
 print(fibo(num))  
 else:

print("wrong call")  
  
  
  
  
  
  
  
  
  
**OUTPUT:**

Q2. At an airport, a traveler is allowed entry into the flight only if he clears the following checks:  
 1. Baggage Check  
 2. Immigration Check   
3. Security Check   
The logic for the check methods are given below:   
check\_baggage (baggage\_weight)   
∙ returns True if baggage\_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.   
check\_immigration (expiry\_year)  
 ∙ returns True if expiry\_year is greater than or equal to 2030 and less than or equal to 2050. Otherwise returns False.  
 check\_security(noc\_status)  
 ∙ returns True if noc\_status is 'valid' or 'VALID', for all other values return False. traveler()   
∙ Initialize the traveler Id and traveler name and invoke the functions check\_baggage(), check\_immigration() and check\_security() by passing required arguments.  
 Refer the table below for values of arguments.  
  
 ∙ If all values of check\_baggage(), check\_immigration() and check\_security() are true, - display traveler\_id and traveler\_name - display "Allow Traveler to fly!"   
Otherwise, - display traveler\_id and traveler\_name - display "Detain Traveler for Re-checking! Invoke the traveler() function.  
 Modify the values of different variables in traveler() function and observe the output.  
  
**CODE :**def traveller(Name,id):

Name=str(input("enter the name :"))

t\_id=int(input("Enter the traveller if:"))

return Name,id

def baggage\_check(weight):

w=int(input("enter the weight of bag:"))

print(w)

if(weight>0 and weight<40):

return true

else:

return false

def check\_immigration(year):

year=int(input("enter the immigration year:"))

print(year)

if(year>=2030 and year<=2050):

return true

else:

return false

def check\_security(noc):

noc\_c=str(input("enter the validity:"))

print(noc\_c)

if(noc=='VALID' or noc=='valid'):

return true

else:

false

print("\n1.check traveller\n2.check baggage weight\n3.check immigration\n4.check security\n")

choice=int(input("enter your choice"))

if(choice==1):

traveller(Name,id)

elif(choice==2):

baggage\_check(weight)

elif(choice==3):

check\_immigration(year)

elif(choice==4):

check\_security(noc)

else:

print("check for credentials")

if(baggage\_check(w)==true and check\_migration(yer)==true and check\_security(noc\_c)==true):

print("name is: ",Name)

print("Travelling id is:",t\_id)

print("allowed to borad plane and fly!!")

else:

print("name is: ",Name)

print("Travelling id is:",t\_id)

print("detained for travelling!!")

**EXPERIMENT -2&3**Ques1. WAP to read the record of n students in a dictionary containing key/value pairs of name: [marks]. Print the average of the marks obtained by the particular student correct to 2 decimal places. Input Format  
 The first line contains the integer n, the number of student’s records. The next n lines contain the names and marks obtained by a student, each value separated by a space.   
Sample Input  
 3  
 Krishna 67 68 69   
Arjun 70 98 63  
 Malika 52 56 60   
Sample Output  
 56.00  
  
**CODE :**dict={}

n=int(input("enter the no. of records"))

for i in range(0,n):

name=str(input("\nenter the name:"))

mark1=int(input("\nenter the marks of subject 1:"))

mark2=int(input("\nenter the marks of subject 2:"))

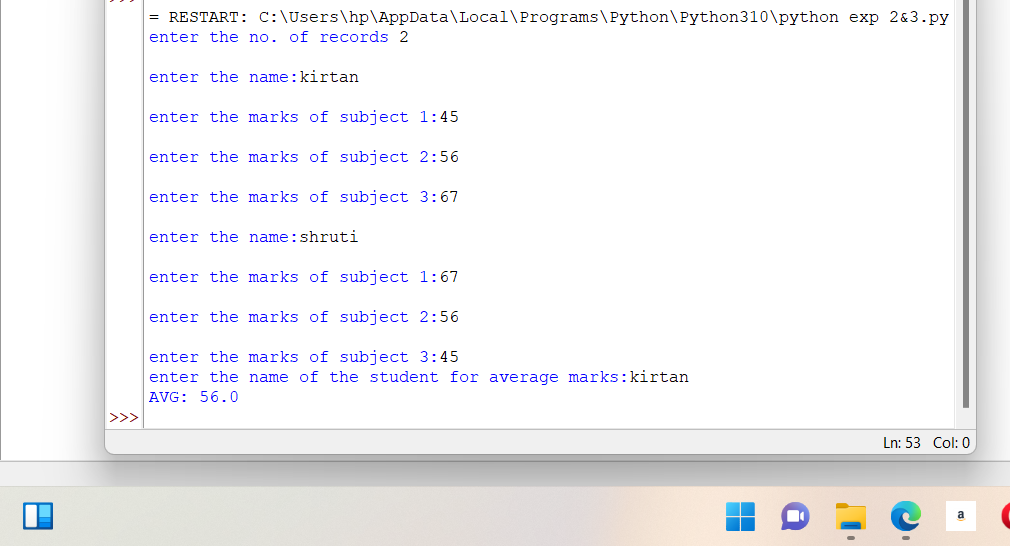
mark3=int(input("\nenter the marks of subject 3:"))

dict[name]=[int(mark1),int(mark2),int(mark3)]

name1=str(input("enter the name of the student for average marks:"))

if name1 in dict.keys():

print("AVG:",sum(dict[name1])/3)

**OUTPUT:  
**Ques2. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.  
Sample Input   
N = 5 Scores= 2 3 6 6 5   
Sample output 5   
  
**CODE :**n=int(input("enter the no. of inputs:"))

list=[]

import math

for i in range(0,n):

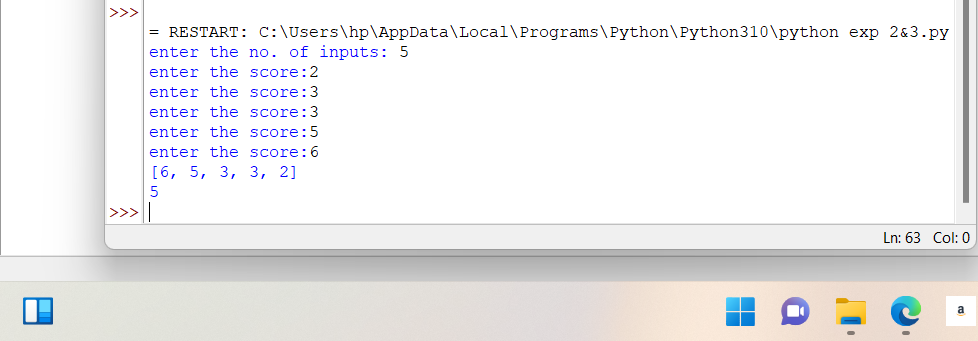
score=int(input("enter the score:"))

list.append(score)

list.sort(reverse=True)

print(list)

print(list[1])

**OUTPUT:  
**Ques3.Q3. Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of country stamps. Find the total number of distinct country stamps using a suitable data type.  
Note: Apply your knowledge of the .add() operation from set to help your friend Rupal.  
  
**CODE :**import random

stamps=['America','Iran','Iraq','India','Africa','Australia','Jordan','Indonesia','Istanbul']

x=random.choices(stamps,k=50)

l=len(stamps)

count\_America=count\_Iran=count\_Iraq=count\_India=count\_Africa=count\_Australia=count\_Jordan=count\_Indonesia=count\_Istanbul=0

l1=len(x)

for i in range(0,9):

if(x[i]=='America'):

count\_America+=1

if(x[i]=='Iran'):

count\_Iran+=1

if(x[i]=='Iraq'):

count\_Iraq+=1

if(x[i]=='India'):

count\_India+=1

if(x[i]=='Africa'):

count\_Africa+=1

if(x[i]=='Australia'):

count\_Australia+=1

if(x[i]=='Jordan'):

count\_Jordan+=1

if(x[i]=='Indonesia'):

count\_Indonesia+=1

if(x[i]=='Istanbul'):

count\_Istanbul+=1

print("America:",count\_America)

print("Iran:",count\_Iran)

print("Iraq:",count\_Iraq)

print("India:",count\_India)

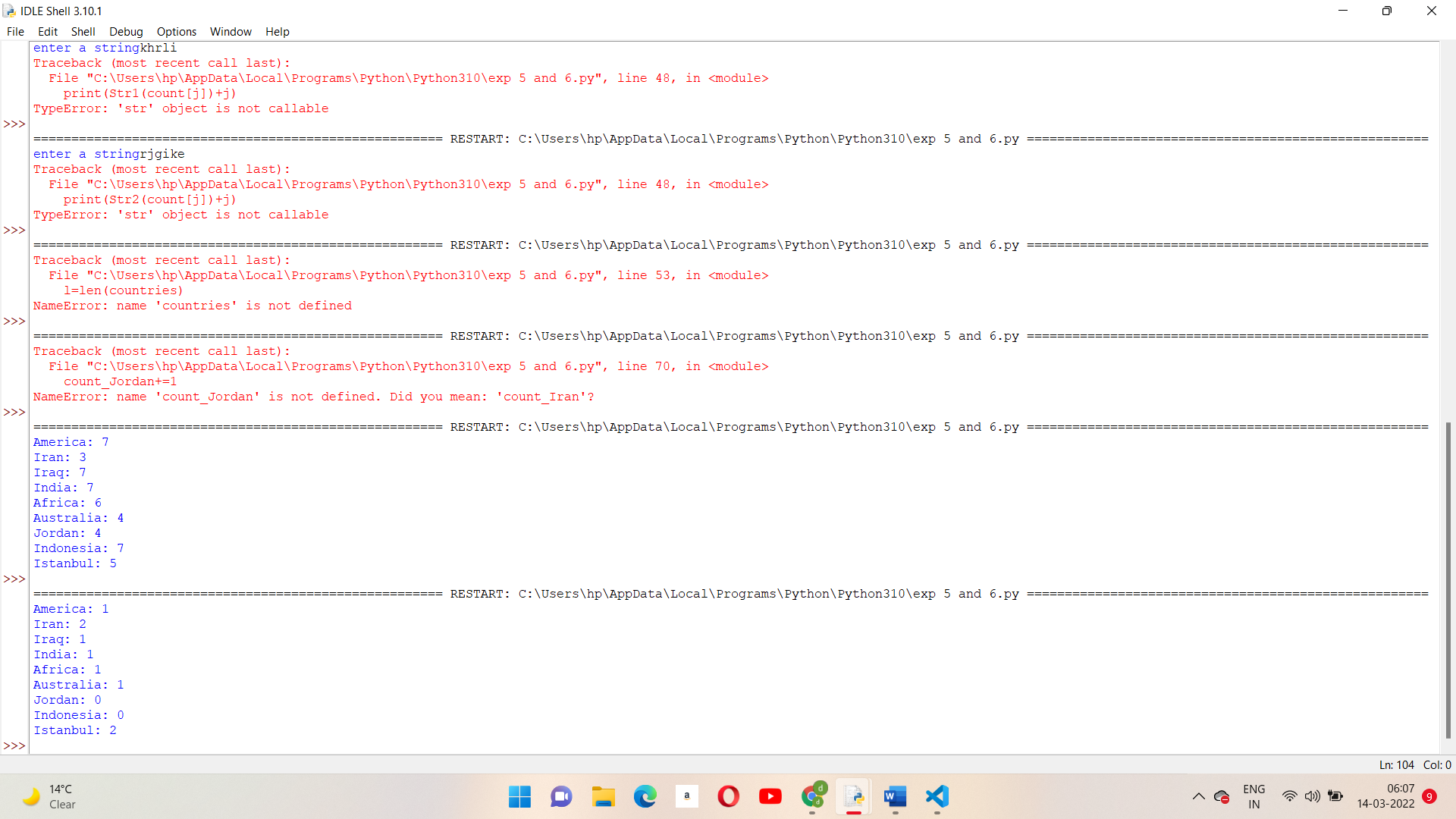
print("Africa:",count\_Africa)

print("Australia:",count\_Australia)

print("Jordan:",count\_Jordan)

print("Indonesia:",count\_Indonesia)

print("Istanbul:",count\_Istanbul)

**OUTPUT:  
**

**EXPERIMENT -4**Ques1. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.   
Sample Input  
 ABCDCDC   
CDC

**CODE :  
def count\_string(string,substring):**

**count=0**

**for i in range(len(string)-len(substring)-1):**

**if(string[i:i+len(substring)]==substring):**

**count+=1**

**return count**

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

**string=raw\_input().strip()**

**count=count\_string(string,substring)**

**print count**

Ques2.WAP to input the first name, middle and last name of a person. Your task is to print the initials of the first and middle name separated by a dot (.) The last name should be followed by a dot and a space where the first letter is capital.   
  
**CODE :**name=input("\nenter the first name:")

mid\_name=input("\nenter the middle name:")

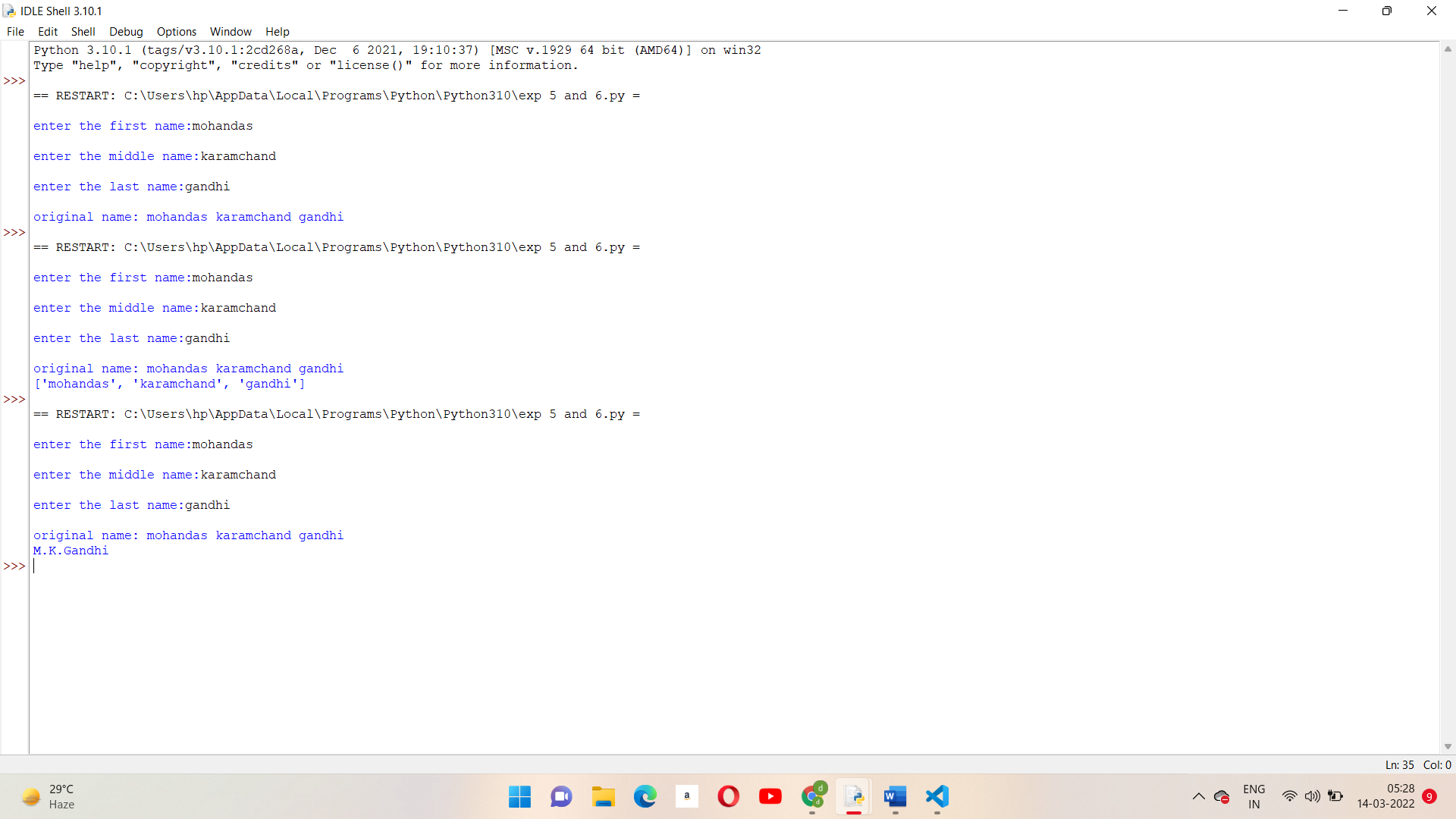
las\_name=input("\nenter the last name:")

print("\noriginal name:",fir\_name,mid\_name,las\_name)

fir\_name\_i=fir\_name[0]

mid\_name\_i=mid\_name[0]

print(fir\_name\_i.capitalize()+"."+mid\_name\_i.capitalize()+"."+las\_name.capitalize())

**OUTPUT:**Ques3.Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.  
Sample Input   
ABaBCbGc  
 Sample Output  
 2A   
3B  
 2C  
1G  
**CODE :**  
**EXPERIMENT -7 and 8**Ques1.Write a Python program to:   
1. read a file.  
 2. add backslash (\) before every double quote in the file contents.  
 3. write it to another file in the same folder.   
4. print the contents of both the files.  
 For example:   
If the first file is 'TestFile1.txt' with text as:  
 Jack said, "Hello Pune".  
 The output of the file 'TestFile2.txt' should be:  
 Jack said,\"Hello Pune\".  
  
**CODE :**f = open("TestFile1.txt", "w")

f.write('Jack said, "Hello jane"')

f.close()

f = open("TestFile1.txt", "r")

data = f.read()

print(data)

lst = []

for i in range(len(data)):

lst.append(data[i])

if (lst[i] == '"'):

lst[i] = ' \\" '

# print(lst)

var = "".join(lst)

f2 = open("TestFile2.txt", "w")

f2.write(var)

f2.close()

f2 = open("TestFile2.txt", "r")

x = f2.read()

print(x)

f2.close()

Ques2.Consider a file 'rhyme.txt' in D Drive with following text:

**CODE :  
with open("rhyme.txt","w")as poemfile:**

**poemfile.write("jingle bells jingle bells\njingle all the day\n oh what it is to ride\n in a horse open sleigh\njingle bells jingle bells\n jingle all the way")**

**with open("rhyme.txt","r") as poemfile:**

**print(poemfile.read())**

**poemfile=open("rhyme.txt","r")**

**count=0**

**for line in poemfile:**

**words=line.split()**

**count+=len(words)**

**print("number of words in file:",count)**

**poemfile=open("rhyme.txt","r")**

**read\_data=poemfile.read()**

**words=set(read\_data.split())**

**for word in words:**

**count=count+1**

**print("total unique words:",count)**

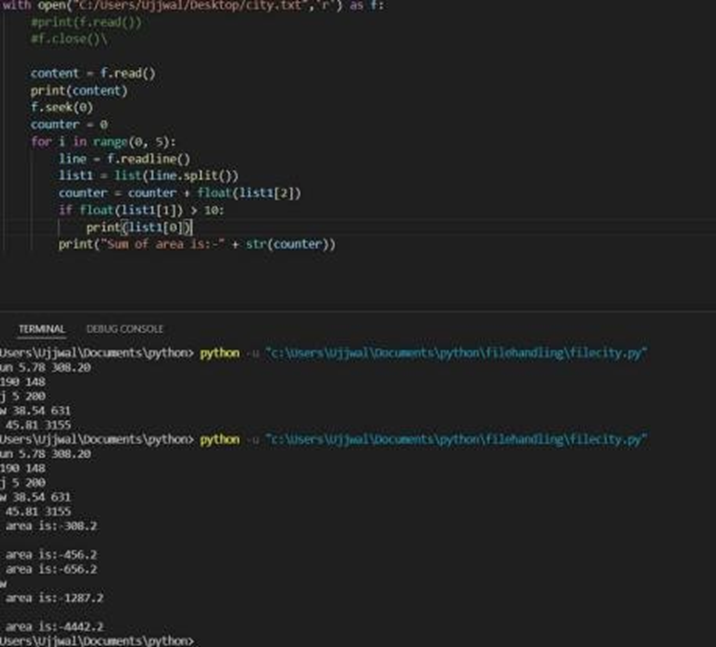
**with open("rhyme.txt","w") as result:**

**add\_text=count**

**print(add\_text,file=result)**

**result.close()**

**with open("rhyme.txt","r") as result:**

**print(result.read())**Ques3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM) ):  
 Example:   
Dehradun 5.78 308.20  
 Delhi 190 1484 ……………  
 Open file city.txt and read to:   
a. Display details of all cities  
 b. Display city names with population more than 10Lakhs   
c. Display sum of areas of all cities   
  
**CODE :  
**

**EXPERIMENT -9**Ques1.Input two values from user where the first line contains N, the number of test cases. The next N lines contain the space separated values of a and b. Perform integer division and print a/b. Print the error code in the case of ZeroDivisionError or ValueError.  
 Sample input  
 1 0  
 2 $  
 3 1   
Sample Output  
 Error Code: integer division or modulo by zero   
Error Code: invalid literal for int() with base 10: '$' 3   
  
**CODE :  
n=int(input("enter the no. of test cases:"))**

**try:**

**N1=int(input("enter the value of a:"))**

**N2=int(input("enter the value of b:"))**

**Divi=N1/N2**

**except zerodivision Error:**

**print("integer division or modulo by zero")**

Ques2. Assume the following Python codeRewrite the code to handle the exceptions raised. Print appropriate error messages wherever applicable.  
  
**CODE :  
sum=0**

**mylist=[1,2,3,"4",5]**

**try:**

**print(mylist[5])**

**for i in mylist:**

**sum=sum+i**

**except IndexError:**

**print("Dont' worry list index out of range")**

**except TypeError:**

**print("INVALID INPUT=Enter String")**

**print(sum)**

Ques3. You have already created a Python program to implement the following in file handling section:   
1. read a file.  
 2. add backslash (\) before every double quote in the file contents.   
3. write it to another file in the same folder.   
4. print the contents of both the files.   
For example: If the first file is 'TestFile1.txt' with text as:   
Jack said, "Hello Pune".   
The output of the file 'TestFile2.txt' should be:  
 Jack said,\"Hello Pune\".   
Modify your code to implement Exception handling. Print appropriate error messages wherever applicable.   
  
**CODE :  
 f = open("TestFile1.txt", "w")**

**f.write('Jack said, "Hello jane"')**

**f.close()**

**f = open("TestFile1.txt", "r")**

**data = f.read()**

**print(data)**

**lst = []**

**for i in range(len(data)):**

**lst.append(data[i])**

**if (lst[i] == '"'):**

**lst[i] = ' \\" '**

**# print(lst)**

**var = "".join(lst)**

**f2 = open("TestFile2.txt", "w")**

**f2.write(var)**

**f2.close()**

**f2 = open("TestFile2.txt", "r")**

**x = f2.read()**

**print(x)**

**f2.close()**

**EXPERIMENT -10 and 11**QUES1: create a class employee with following properties :  
 **CODE:  
class EMPLOYEE():**

**def \_\_init\_\_(self,fname,lname,pay):**

**self.fname=fname**

**print("fisrt name is:",fname)**

**self.lname=lname**

**print("last name is:",lname)**

**self.pay=pay**

**print("pay is :",pay)**

**print("Email is:",self.fname,".",self.lname,"@compan.com")**

**A=EMPLOYEE("Mohandas","Gandhi",5000)**

QUES2: Perform the following instructions:

a) Create a Vehicle class with max\_speed and mileage as instance attributes. Additionally, create a method named seating\_capacity() using the below syntax: def seating\_capacity(self, capacity): return f"The seating capacity of a {self.name} is {capacity} passengers"   
b) Create child class ‘Bus’ that will inherit all of the variables and methods of the Vehicle class. Set the seating capacity of the bus to 50 using super().  
 c) Create a Bus object that will inherit all of the variables and methods of the Vehicle class and display it.   
d) Define a class attribute “color” with a default value white. I.e., Every Vehicle should be white.  
  
CODE:  
class Vehicle:  
 def \_\_init\_\_(self, name, max\_speed, mileage):

self.name = name

self.max\_speed = max\_speed

self.mileage = mileage

def seating\_capacity(self, capacity):

return f"The seating capacity of a {self.name} is {capacity} passengers"

class Bus(Vehicle):

pass

class Car(Vehicle):

pass

QUES 3:list the risk associated with………………….  
Ans: risk associated with it all the bank details will be revealed , transactions will get mixed up and every detail is just gone .

solu : risk associated is that suppose in bank someone did a transaction into his or her account of 5000 and when some other wants to make it out who has only 2000 in his account can get 7000  
due to this . All the transactions need to be safe for this we have to avoid this at all .  
  
  
 **EXPERIMENT -12 and 13**QUES1:.  
 a) import Tkinter package and create a window and set its title  
 b) set the default window size using geometry function  
 c) Create a label with “Hello” text in it and set its position on the form.  
 d) Add a button to the window with “CLICK ME” written on it.   
e) change the foreground and background color for the button created above  
 f) Create a function that will be executed when the button is clicked and print “Button was clicked” on clicking the button   
  
**CODE:  
from tkinter import \***

**root = Tk()**

**root.geometry("500x500")**

**def click():**

**mylabel = Label(root,text="Button was clicked").pack()**

**btn = Button(root,text="Click Me",fg="blue",bg="green",command=click)**

**label = Label(root,text="Hello")**

**label.pack()**

**btn.pack()**

**root.mainloop()  
  
**QUES2: This is the continuation of Question1, add the given below features in the above program:   
a) Take user name as input using the Tkinter Entry class   
b) Print the entered text (username) on clicking the button.   
c) Create three RadioButtons as displayed below   
d) Print the currently selected radio button or the radio button value.  
  
  
**CODE :   
from tkinter import \***

**root = Tk()**

**entry = Entry(root)**

**def show\_name():**

**username = entry.get()**

**label = Label(root,text=username)**

**label.grid(row=6,column=1)**

**def func():**

**rdo = Label(root,text="You have clicked"+" "+str(var.get())).grid(row=6,column=2)**

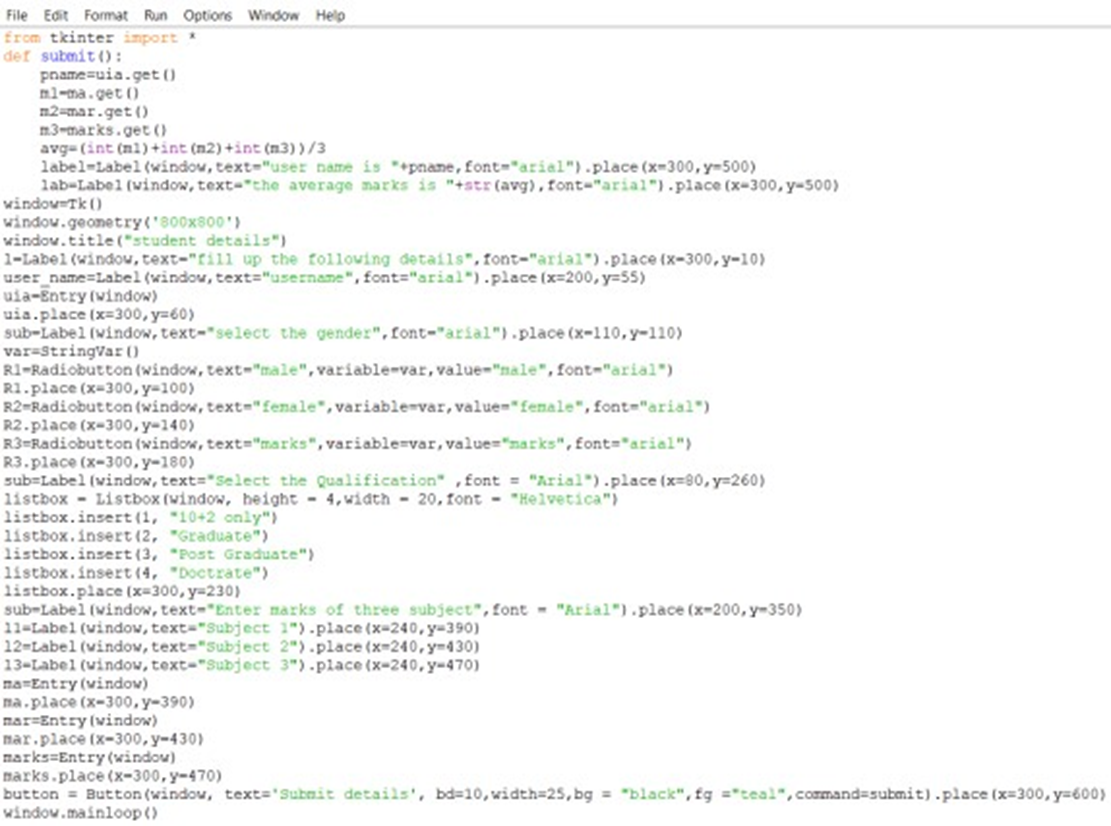
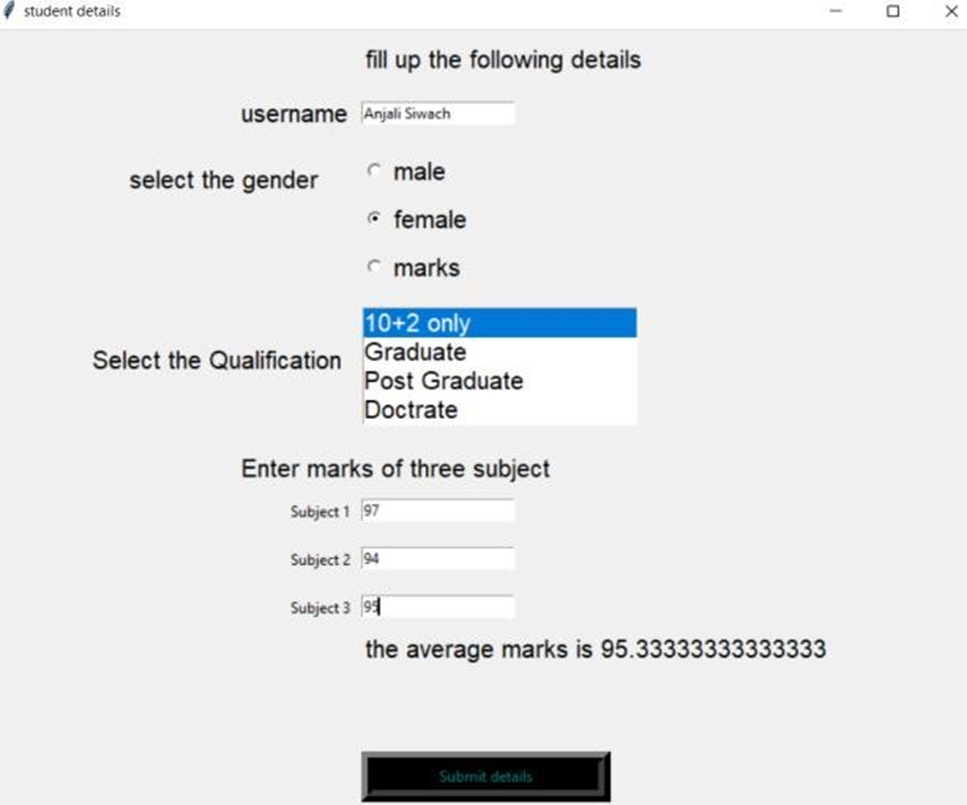
**var=IntVar()**

**f= Radiobutton(root, text="first choice", value=1, command=func, variable=var)**

**f.grid(row=1, column=0)  
s= Radiobutton(root, text="second choice", value=2, command=func, variable=var)  
s.grid(row=1, column=1)  
th= Radiobutton(root, text="third choice", value=3, command=func, variable=var)  
th.grid(row=1, column=2)**

**btn =Button(root,text="Show",command=show\_name)  
btn.grid()  
entry.grid(row=2,column=1)**

**root.mainloop()**

QUES3: Write a program to accept following details from a student using GUI   
1. Name of the student (using Textbox)   
2. Gender (Using radio button)  
 3. Qualification (Using List)  
 4. Marks of three subjects (using Textbox)  
 Compute the percentage of the student and display it in a textbox.   
**CODE :  
  
  
  
  
  
OUTPUT :  
**

**EXPERIMENT -14**QUES 1:   
a. Convert numbers =[1, 2.0, 3] to numpy array and convert all elements to string type.   
b. Create a 2 D array through list and set dtype as int32   
c. Find the rows and columns of the 2d array created in part b  
d. Print 10 random numbers between 1 and 100.  
**CODE** :   
a.import np

numbers = [1,2.0,3]

arr = np.array(numbers)

string = arr.tobytes()

print(np.frombuffer(string,dtype = int))  
  
b.import numpy

import numpy as np

A=[1,2,3,4,5],[1,2,3,4,5]

arr = np.array(A,dtype=np.int32)

print(arr.dtype)

print(arr)  
  
c. import numpy

import numpy as np

A=[1,2,3,4,5],[1,2,3,4,5]

arr = np.array(A,dtype=np.int32)

print(arr.dtype)

print(arr)

rows = len(arr)

cols = len(arr[0])

print('Length of rows is',rows)

print('Length colums is ',cols)  
  
d.import random

randomlist = random.sample(range(1,100),10)

print(randomlist)

QUES 2:  
a) Write a NumPy program to get help on the add function   
b) Write a NumPy program to test whether none of the elements of a given array is zero  
c) Write a NumPy program to test whether any of the elements of a given array is non-zero  
d) Write a NumPy program to generate an array of 15 random numbers from a standard normal distribution   
  
**CODE** :   
a.import numpy as np

print(np.info(np.add))

b.,c.import numpy as np

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

print(x)

print("Testing whether none of the elements of a given array is zero")

print(np.all(x))

print("Testing whether none of the elements of a given array is non-zero")

print(np.any(x))  
  
d.import numpy as np

random\_num = np.random.normal(0,2,15)

print(random\_num)