

**BHARATI VIDYAPEETH’S**

**INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT**

(Affiliated to Guru Gobind Singh Indraprastha University, Approved by AICTE, New Delhi)

**Python**

**Programming**

**(MCA-106)**

**Practical File**

**Submitted To: Submitted By:**

Dr. Rakhee Sharma Sanchit Ahuja (01911604422)

(Assistant Professor) MCA 2ndSem, Sec 1

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Problem Description | Date of Execution | Date of Submisssion |
| 1 | Implement Python Script to generate first N natural numbers. | 24-03-2023 | 31-03-2023 |
| 2 | By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms. | 24-03-2023 | 31-03-2023 |
| 3 | Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000. | 24-03-2023 | 31-03-2023 |
| 4 | Write a function cumulative\_product to compute cumulative product of a list of numbers. | 24-03-2023 | 31-03-2023 |
| 5 | Write a function reverse to reverse a list. Without using the reverse function. | 29-03-2023 | 31-03-2023 |
| 6 | Define a function which generates Fibonacci series up to n numbers using RECURSION. | 29-03-2023 | 31-03-2023 |
| 7 | With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line. | 29-03-2023 | 31-03-2023 |
| 8 | Write a program to count the numbers of characters in the string and store them in a dictionary data structure. | 29-03-2023 | 31-03-2023 |
| 9 | Remove spaces from a string using recursion. | 29-03-2023 | 31-03-2023 |
| 10 | Write a program to compute the number of characters, words and lines in a file. | 29-03-2023 | 31-03-2023 |
| 11 | Write a Python class which has two methods get\_String and print\_String. get\_String accept a string from the user and print\_String print the string in upper case. | 25-05-2023 | 28-05-2023 |
| 12 | Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle. | 25-05-2023 | 28-05-2023 |
| 13 | Write a Python class to reverse a string word by word  Input string : 'hello Python'  Expected Output : 'Python hello' | 25-05-2023 | 28-05-2023 |
| 14 | Write python functions to produce following output patterns  a)    b) | 25-05-2023 | 28-05-2023 |
| 15 | Write a function that finds the sum of first n terms of following series: 1 – x2 / 2! + x4 / 4! – x6 / 6! + …… xn / n! | 25-05-2023 | 28-05-2023 |
| 16 | Write a program to compare two files and display total number of lines in a file. | 09-06-2023 | 10-06-2023 |
| 17 | Every time a vote is cast the name of the candidate is appended to the data structure. Print the names of candidates who received maximum vote in lexicographical order and if there is a tie print lexicographically smaller name. | 09-06-2023 | 10-06-2023 |
| 18 | Write a program to determine whether a given string has balanced parenthesis or not. | 09-06-2023 | 10-06-2023 |
| 19 | Implement a python script to check the element is in the list or not by using Linear search & Binary search. | 09-06-2023 | 10-06-2023 |
| 20 | Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques | 09-06-2023 | 10-06-2023 |
| 21 | Create a menu driven program to perform various matrices operations. | 09-06-2023 | 10-06-2023 |
| 22 | Draw different graphs and plots in Python using Matplotlib Library. | 09-06-2023 | 10-06-2023 |
| 23 | Write a python code for simple GUI calculator using Tk. | 09-06-2023 | 10-06-2023 |
| 24 | Write a python program to perform various database operations (create, insert, delete, update) | 09-06-2023 | 10-06-2023 |

**Question 1 Implement Python Script to generate first N natural numbers**

**Solution**

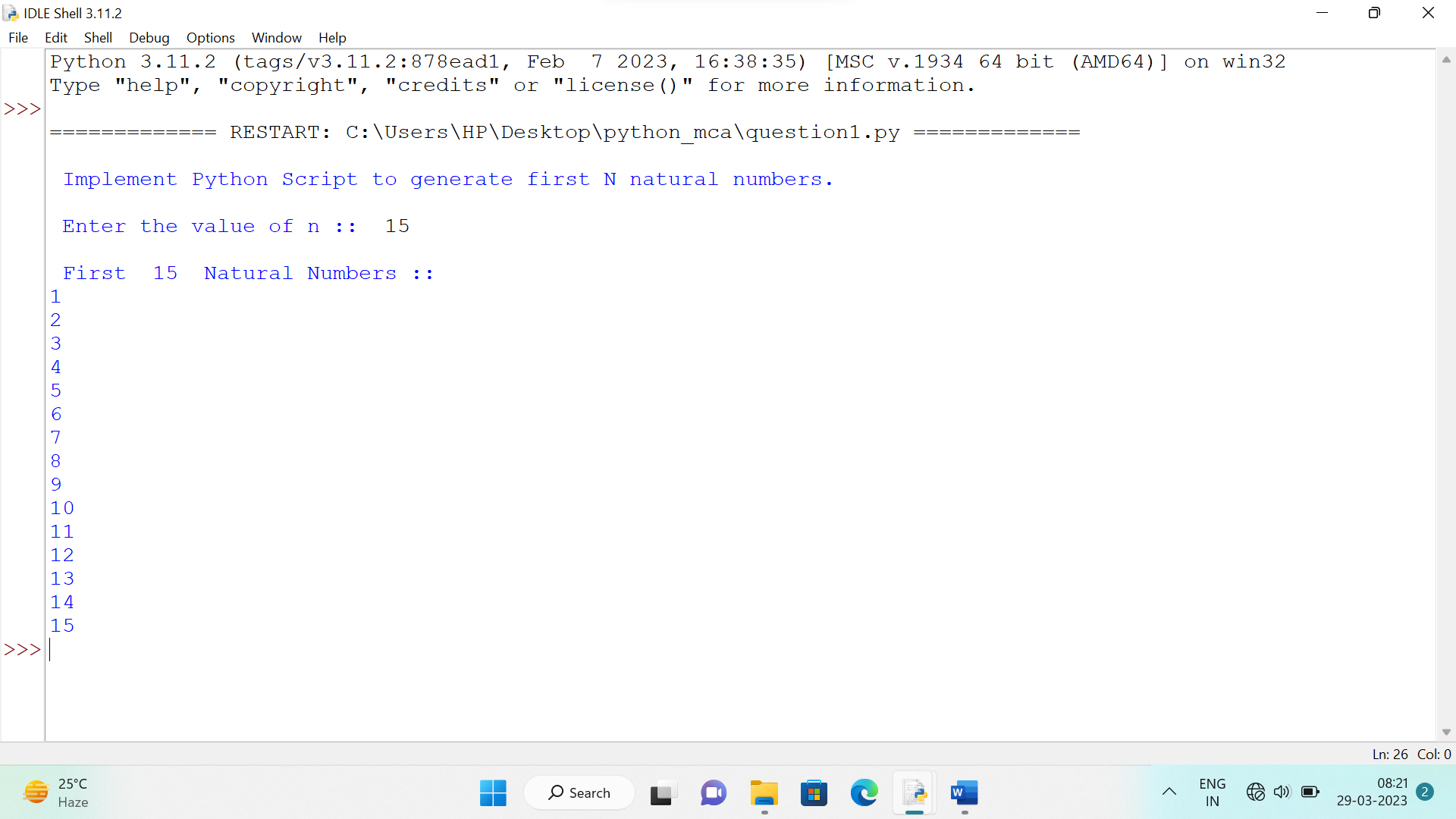
print("\n Enter the value of n :: ",end=' ')

n=int(input())

print("\n First ",n," Natural Numbers :: ")

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

print(i)



**Question2 By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms**

**Solution**

term1=0

term2=1

sum=0

print("\n Enter n :: ",end=" ")

n=int(input())

print("\n Fibonacci Series :: ",end=" ")

print(term1,"",term2,end=" ")

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

term3=term1+term2

if term3 <=1000 and term3%2==0:

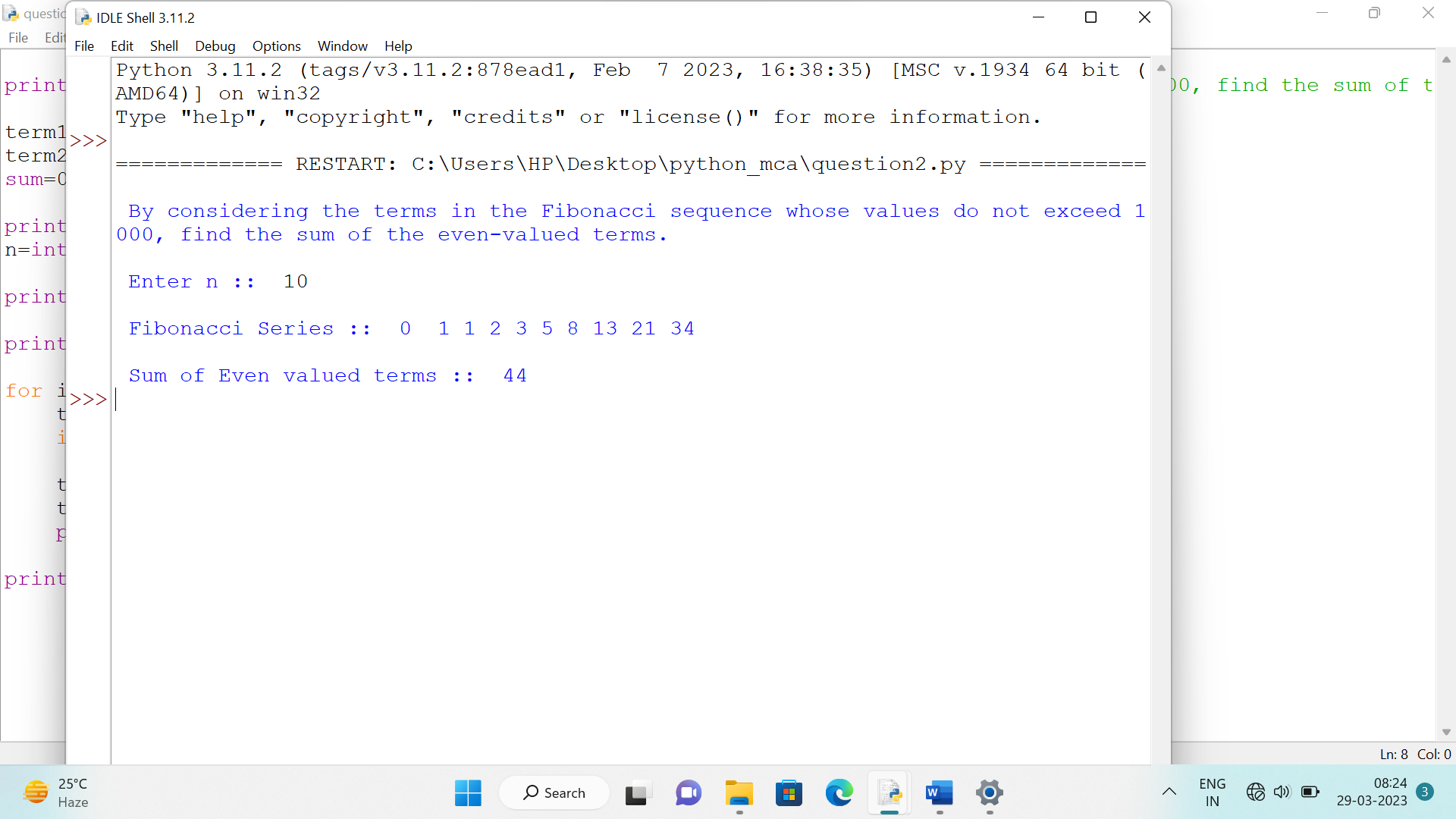
sum=sum+term3

term1=term2

term2=term3

print(term3,end=" ")

print("\n\n Sum of Even valued terms :: ",sum)



**Question 3 Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.**

**Solution**

def func(start,end):

print("\n\n Numbers divisible by 7 but are not multiple of 5 \n")

for i in range(start,end):

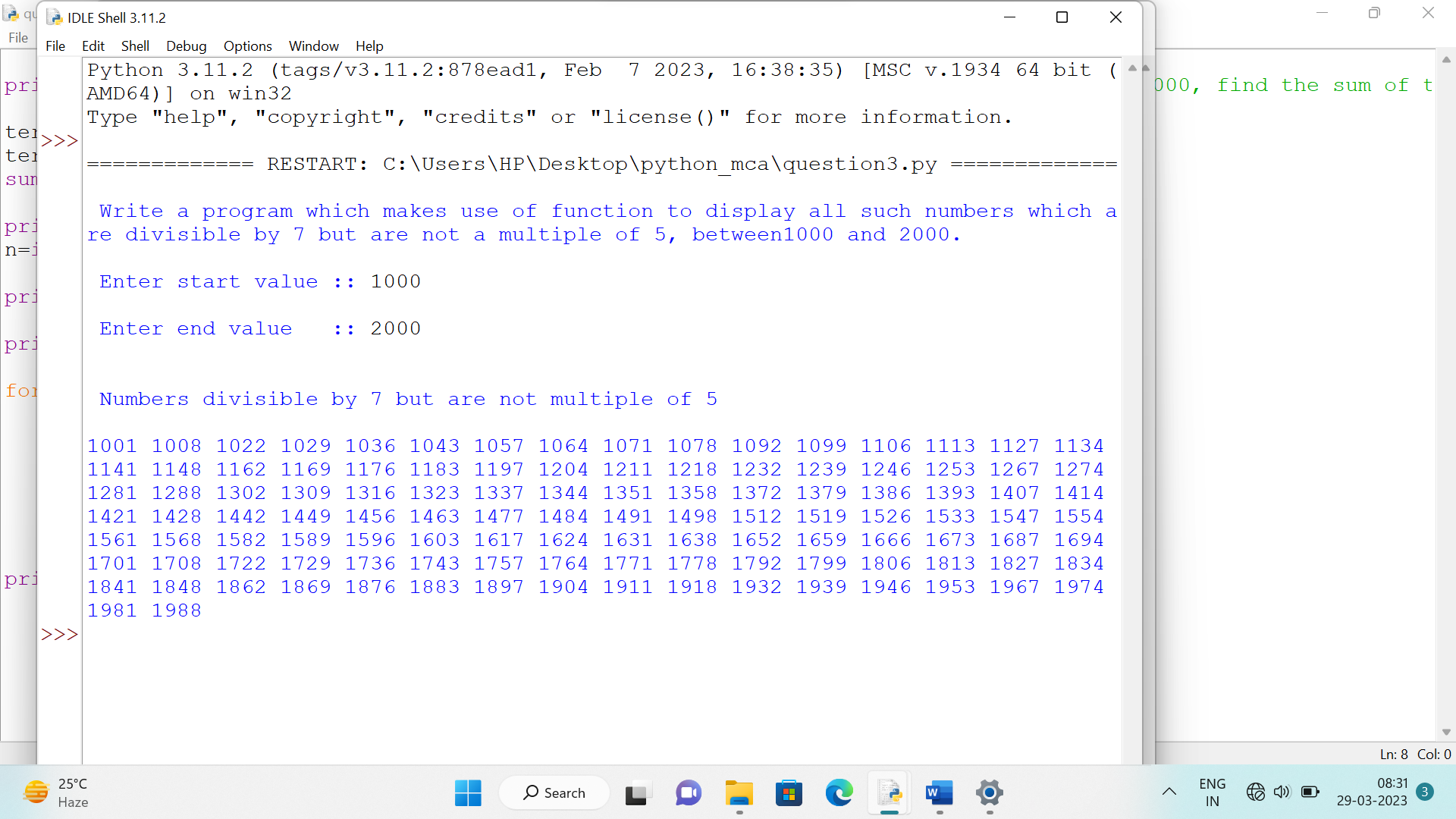
if i%7==0 and i%5!=0:

print(i,end=" ")

start=int(input("\n Enter start value :: "))

end=int(input("\n Enter end value :: "))

func(start,end)



**Question 4 . Write a function cumulative\_product to compute cumulative product of a list of numbers.**

**Solution**

def cumulative\_product(mylist):

product=1

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

product\*=mylist[i];

print("\n Product = ",product)

list1=[]

print("\n Enter n :: ",end=" ")

n=int(input())

print("\n Enter elements of list :: ")

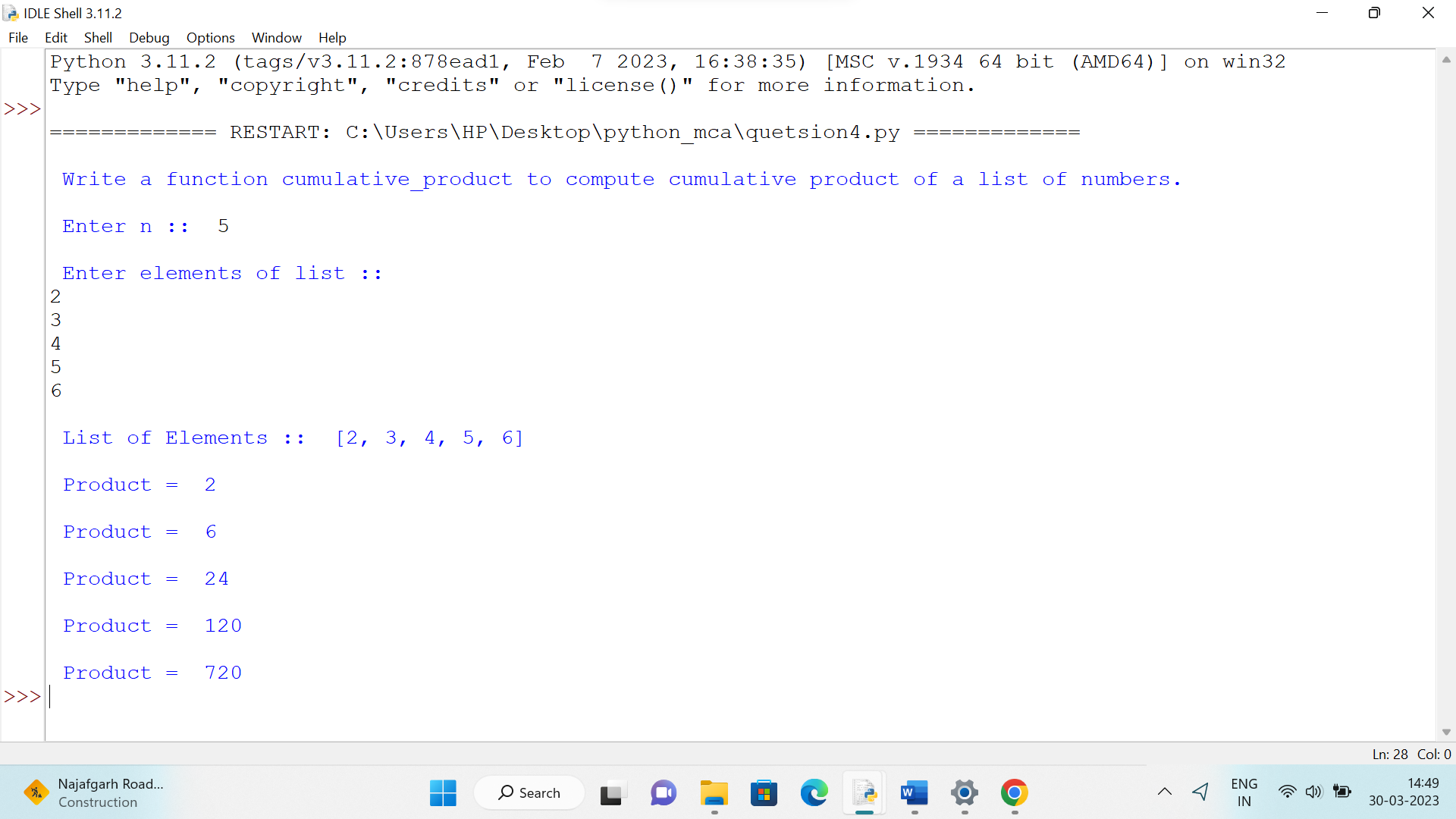
for i in range(0,n):

num=int(input())

list1.append(num)

print("\n List of Elements :: ",list1)

cumulative\_product(list1)

****

**Question 5 Write a function reverse to reverse a list. Without using the reverse function**

**Solution**

def reverse(list1,n):

for i in range(0,n//2):

temp=list1[i]

list1[i]=list1[n-i-1]

list1[n-i-1]=temp

print("\n Reverse of List :: ",list1)

list1=[]

print("\n Enter Size of List :: ",end=" ")

n=int(input())

print("\n Enter Elements :: ")

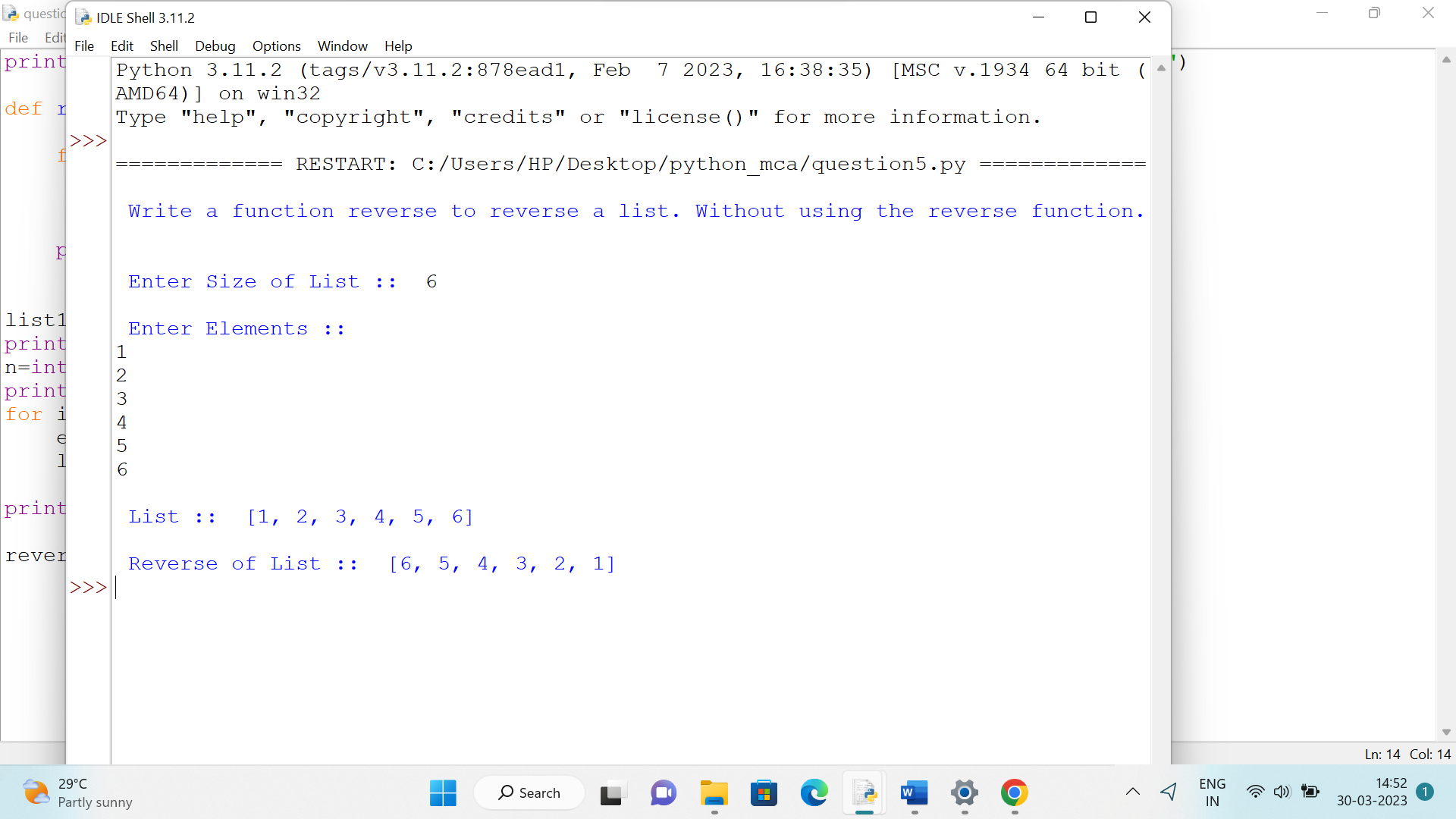
for i in range(0,n):

ele=int(input())

list1.append(ele)

print("\n List :: ",list1)

reverse(list1,n)



**Question 6 Define a function which generates Fibonacci series up to n numbers using RECURSION.**

**Solution**

def fibonacci(n):

if n<=1:

return n

else:

return (fibonacci(n-1)+fibonacci(n-2))

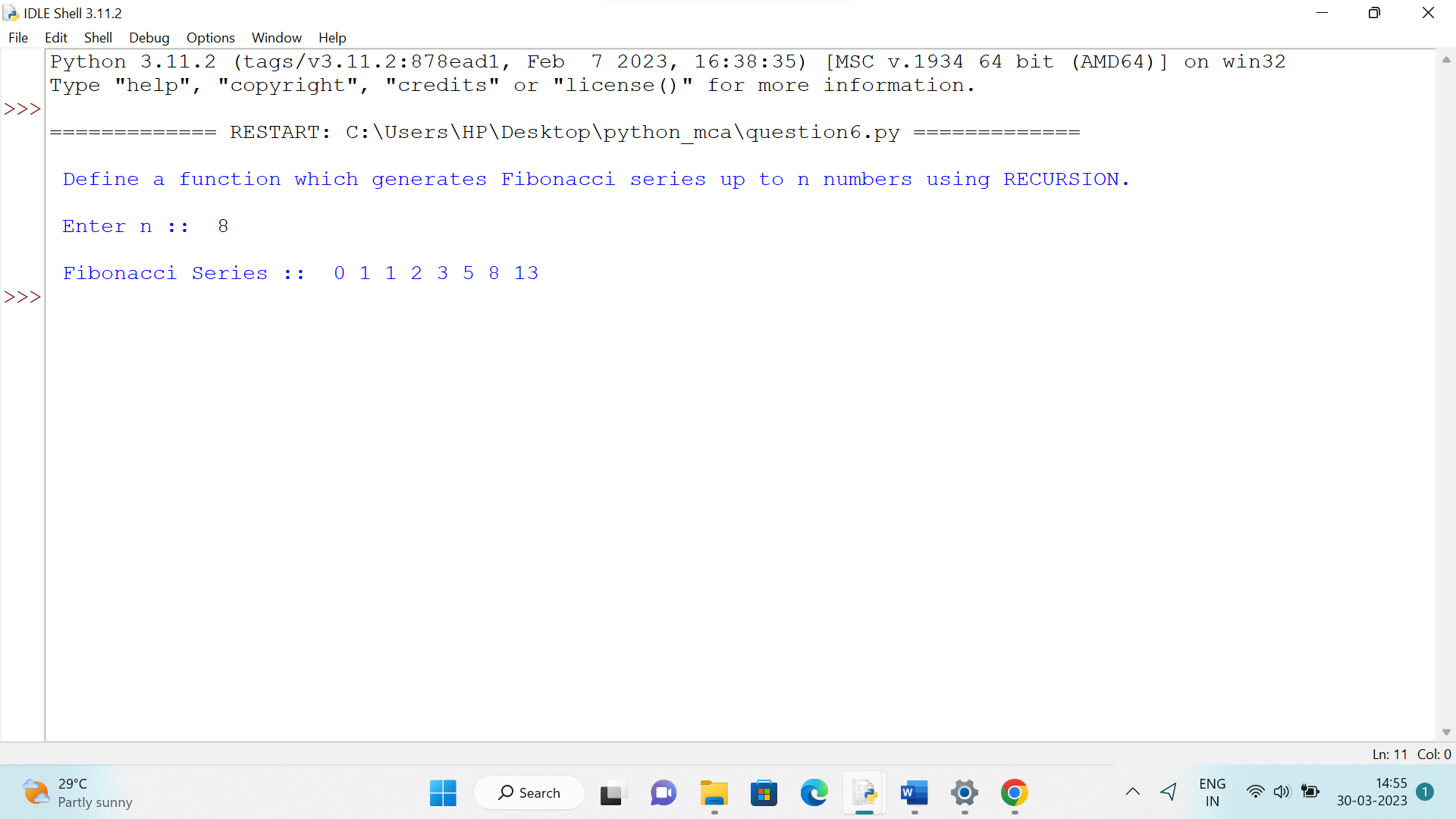
print("\n Enter n :: ",end=" ")

n=int(input())

print("\n Fibonacci Series :: ",end=" ")

for i in range(n):

print(fibonacci(i),end=" ")



**Question 7 With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.**

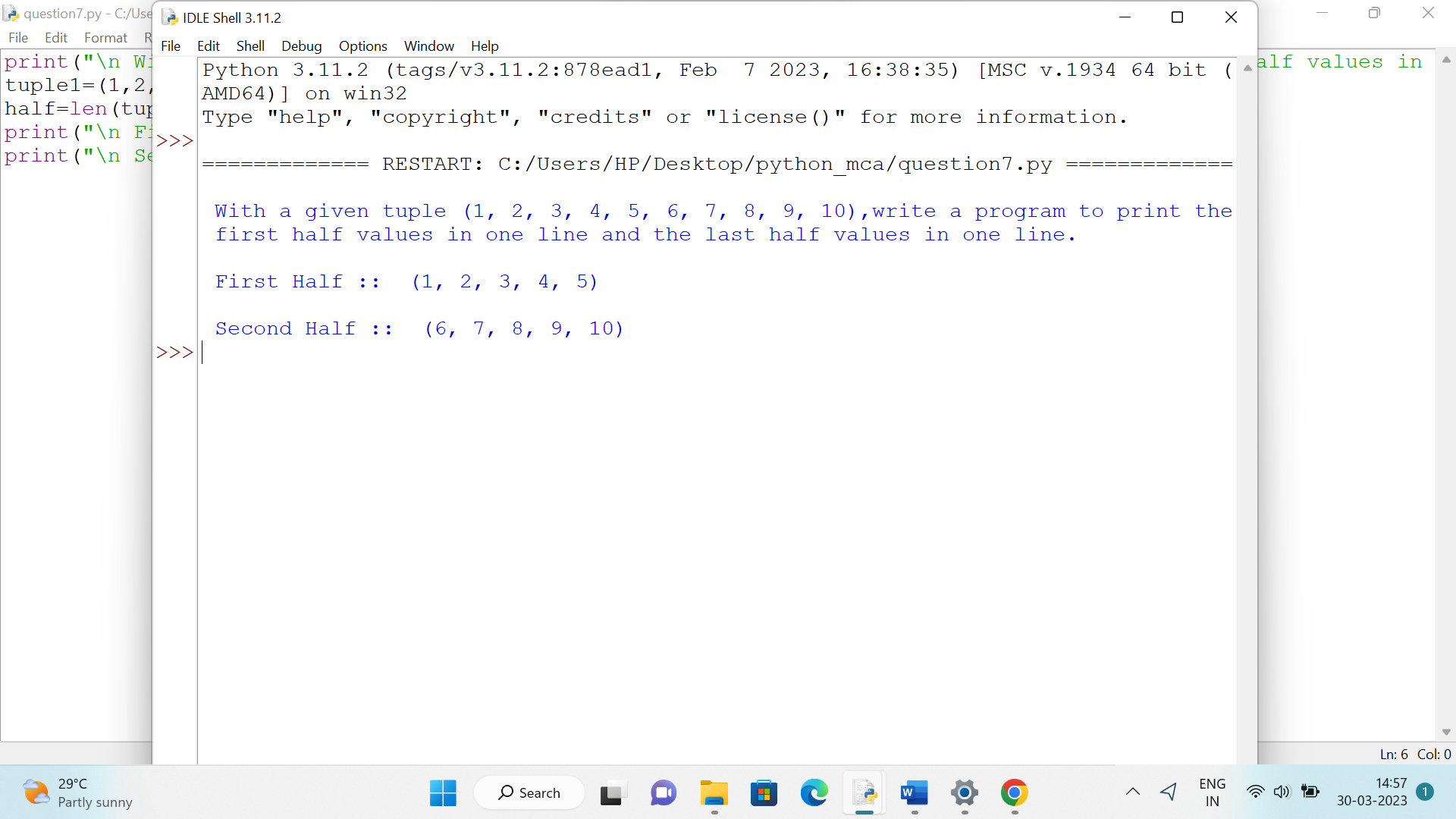
**Solution**

tuple1=(1,2,3,4,5,6,7,8,9,10)

half=len(tuple1)//2

print("\n First Half :: ",tuple1[:half])

print("\n Second Half :: ",tuple1[half:])



**Question 8 Write a program to count the numbers of characters in the string and store them in a dictionary data structure.**

**Solution**

print("\n Enter a String :: ",end=" ")

string1=input()

count=0

print("\n Entered String :: ",string1)

newstring=string1.lower()

mydictionary={}

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

if(newstring[i]>='a' and newstring[i]<='z'):

count=count+1

if newstring[i] in mydictionary:

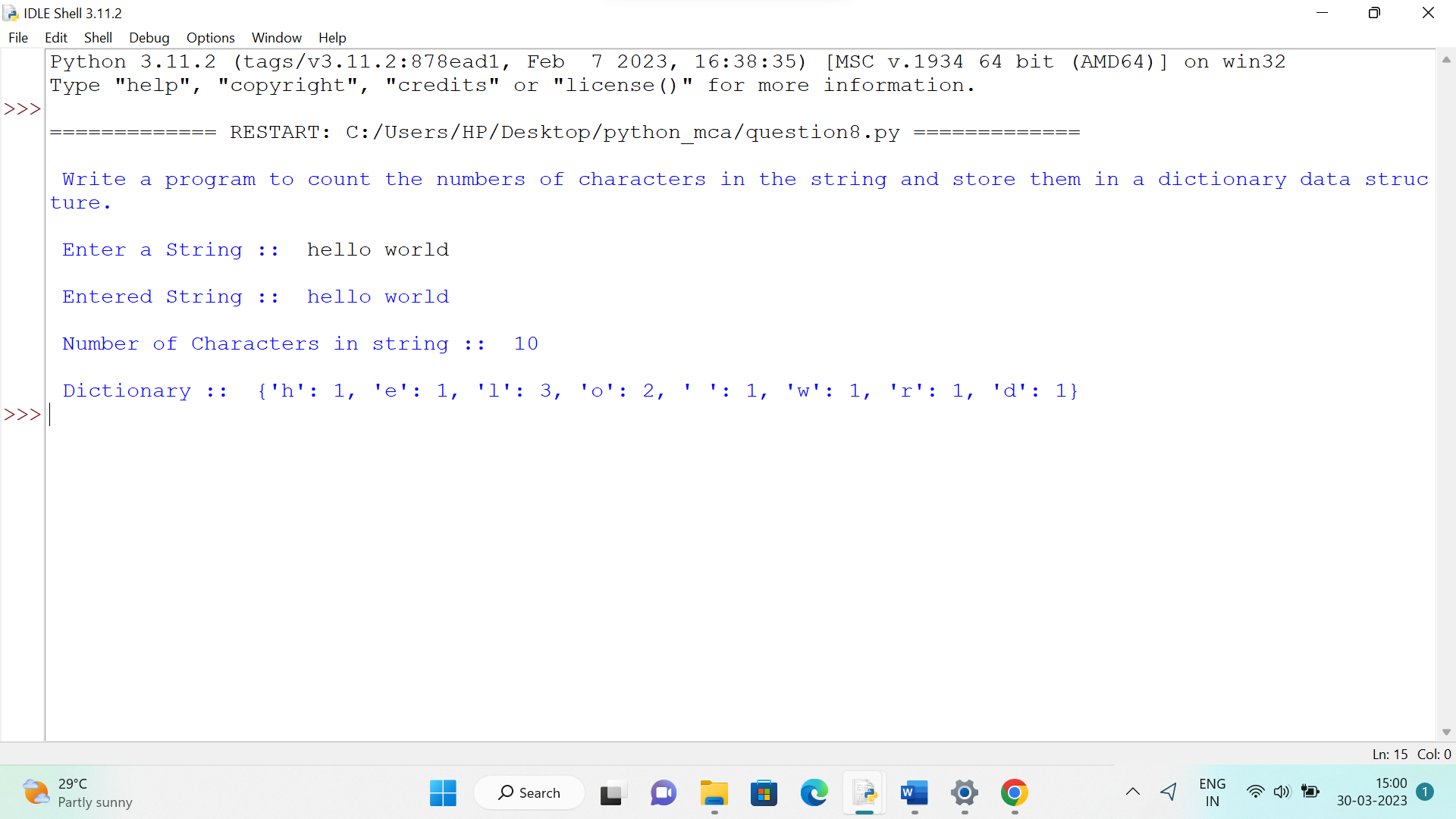
mydictionary[newstring[i]]+=1

else:

mydictionary[newstring[i]]=1

print("\n Number of Characters in string :: ",count)

print("\n Dictionary :: ",mydictionary)



**Question 9 Remove spaces from a string using recursion.**

**Solution**

print("\n Enter a string :: ",end=" ")

string1=input()

def remove\_spaces(string1):

# If we reach end of string

if len(string1)==0:

return ""

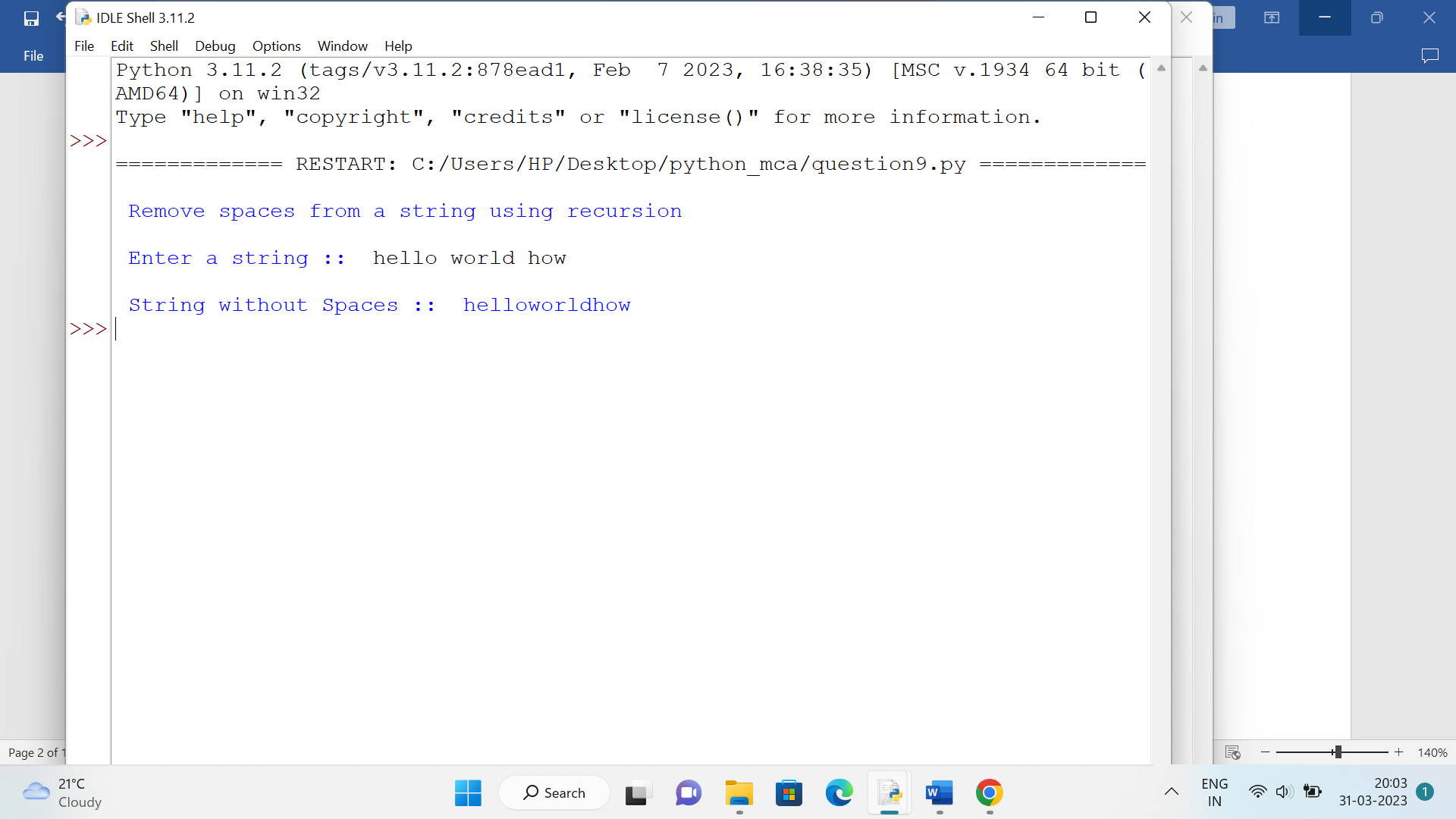
elif string1[0]==" ":

return remove\_spaces(string1[1:])

else:

return string1[0]+remove\_spaces(string1[1:])

print("\n String without Spaces :: ",remove\_spaces(string1))



**Question 10 Write a program to compute the number of characters, words and lines in a file.**

**Solution**

filename="a.txt"

list1=[]

print("\n Text in File ",filename," :: \n\n")

with open (filename,'r') as f:

lines=0

words=0

characters=0

for line in f:

lines=lines+1

list1=line.split(' ')

words=words+len(list1)

for i in range(len(line)):

if((line[i]>='a' and line[i]<='z') or(line[i]>='A' and line[i]<='Z')):

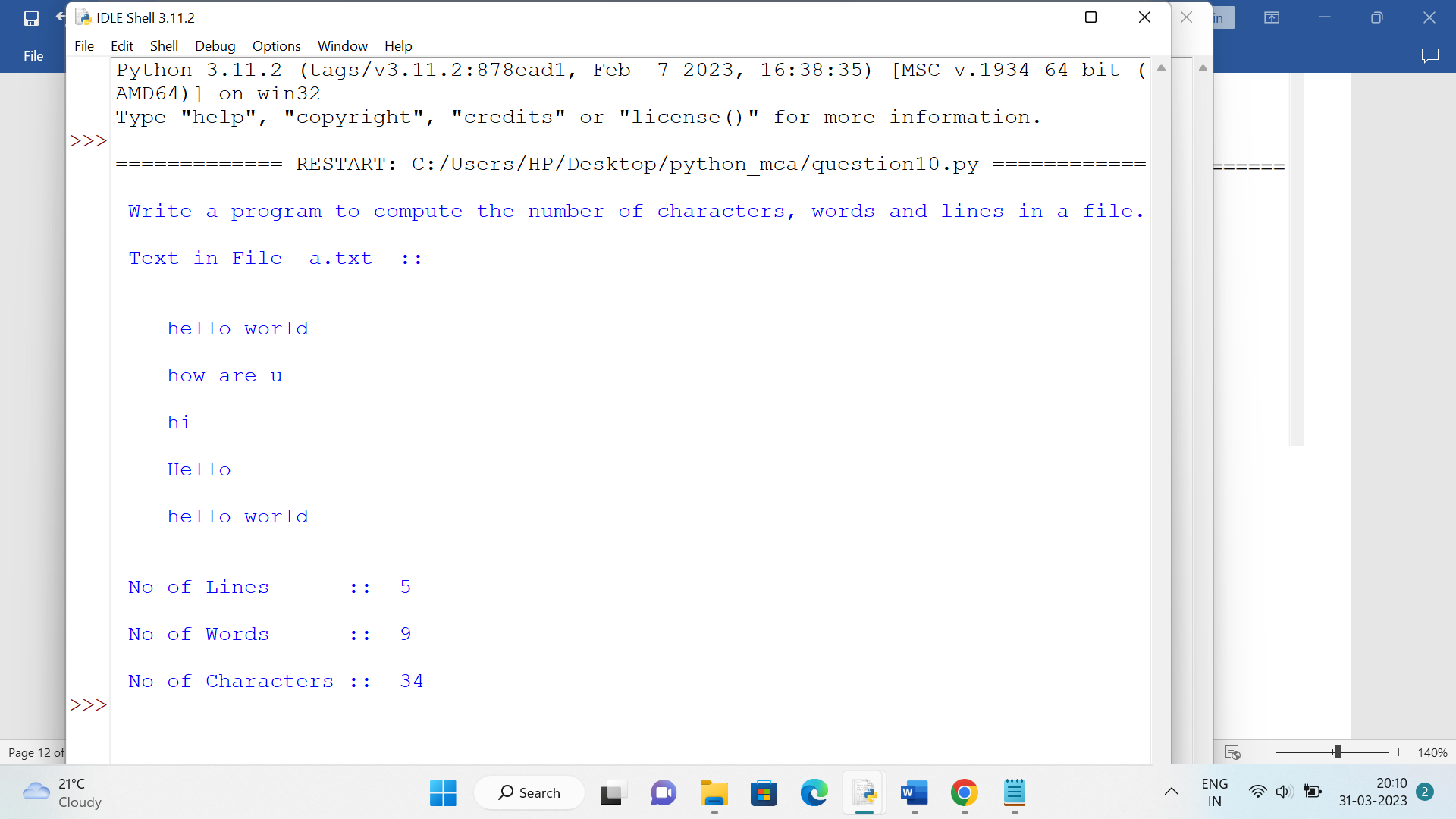
characters=characters+1

print(" ",line)

print("\n No of Lines :: ",lines)

print("\n No of Words :: ",words)

print("\n No of Characters :: ",characters)

****

**Question 11 Write a Python class which has two methods get\_String and print\_String. get\_String accept a string from the user and print\_String print the string in upper case**.

**Solution**

class myclass:

def get\_string(self):

print("\n Enter String :: ",end=" ")

self.name=input()

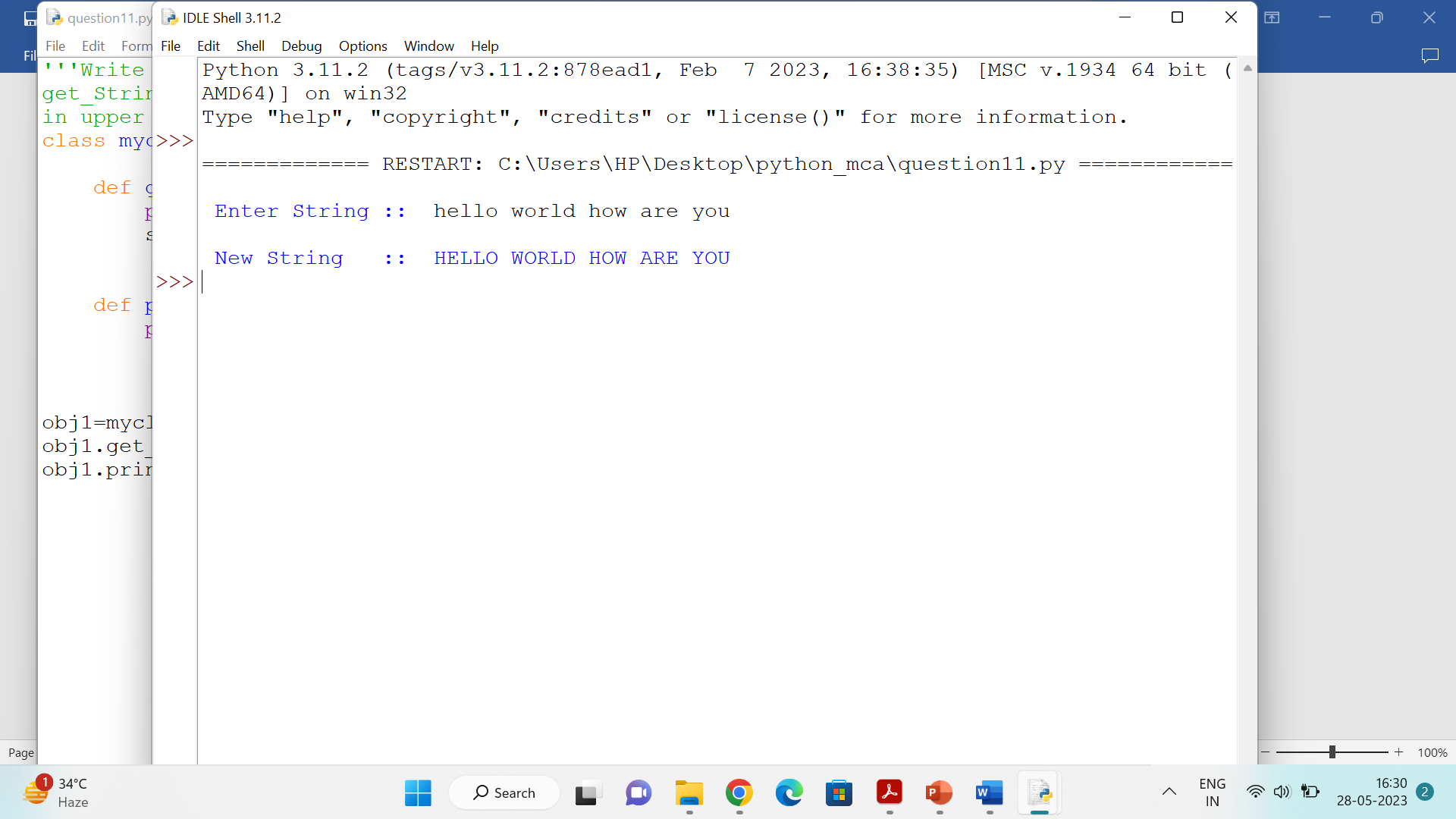
def print\_string(self):

print("\n New String :: ",self.name.upper())

obj1=myclass()

obj1.get\_string()

obj1.print\_string()



**Question 12 Write a Python class named Circle constructed by a radius and two**

**methods which will compute the area and the perimeter of a circle.**

**Solution**

import math

class Circle:

def \_\_init\_\_(self,radius):

self.radius=radius

def area\_circle(self):

area=3.14\*pow(self.radius,2)

print("\n Area of Circle :: ",area)

def perimeter\_circle(self):

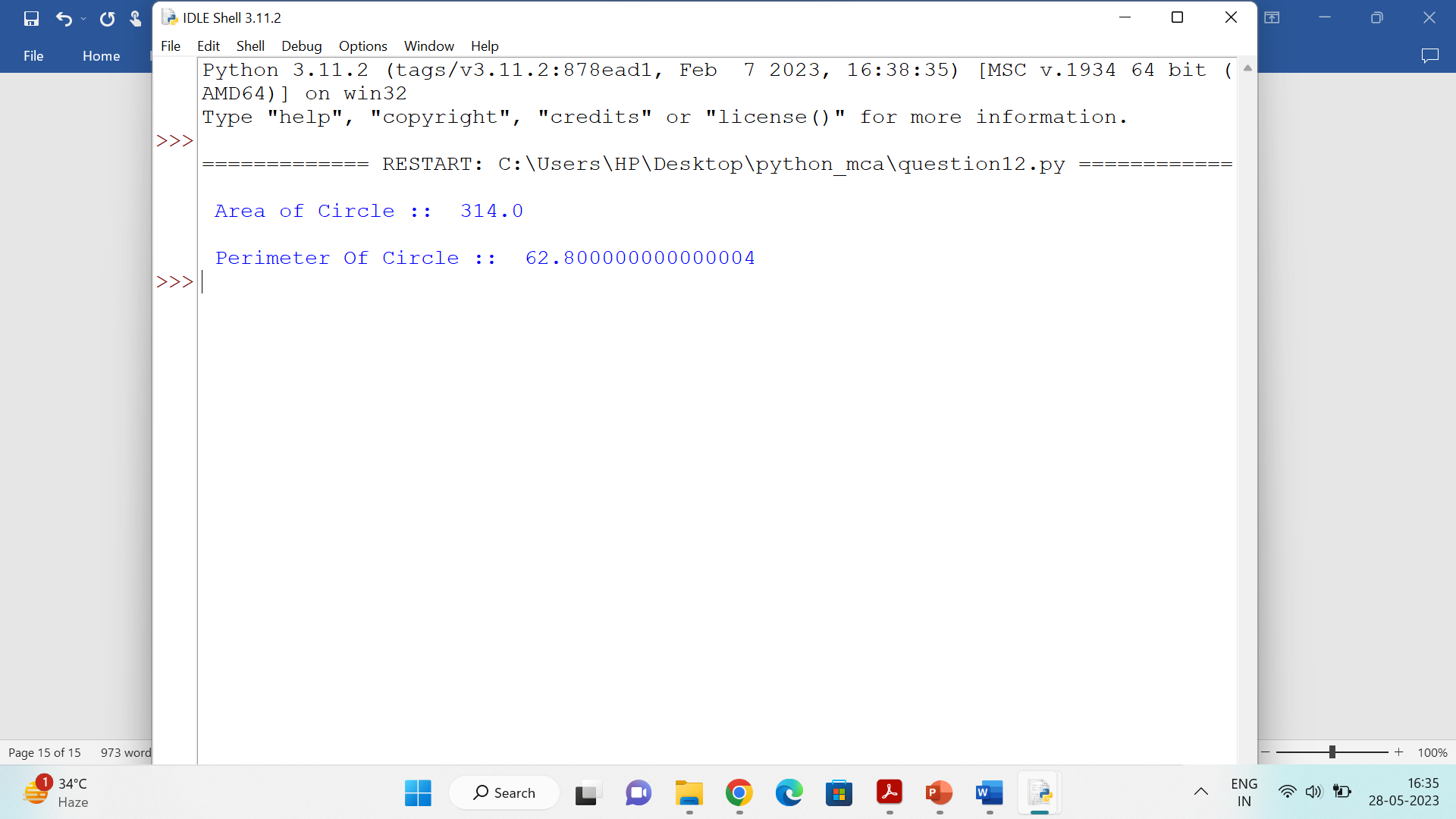
perimeter=2\*3.14\*self.radius

print("\n Perimeter Of Circle :: ",perimeter)

c1=Circle(10)

c1.area\_circle()

c1.perimeter\_circle()



**Question 13 Write a Python class to reverse a string word by word :**

**Input string : 'hello Python'**

**Expected Output : 'Python hello'**

**Solution**

class reverse\_string:

def get\_string(self):

print("\n Enter String :: ",end=" ")

self.string=input()

def rev\_string(self):

a=[]

a=self.string.split()

rev\_string=''

for i in range(len(a)-1,-1,-1):

rev\_string=rev\_string+" "+a[i]

self.string=rev\_string

def print(self):

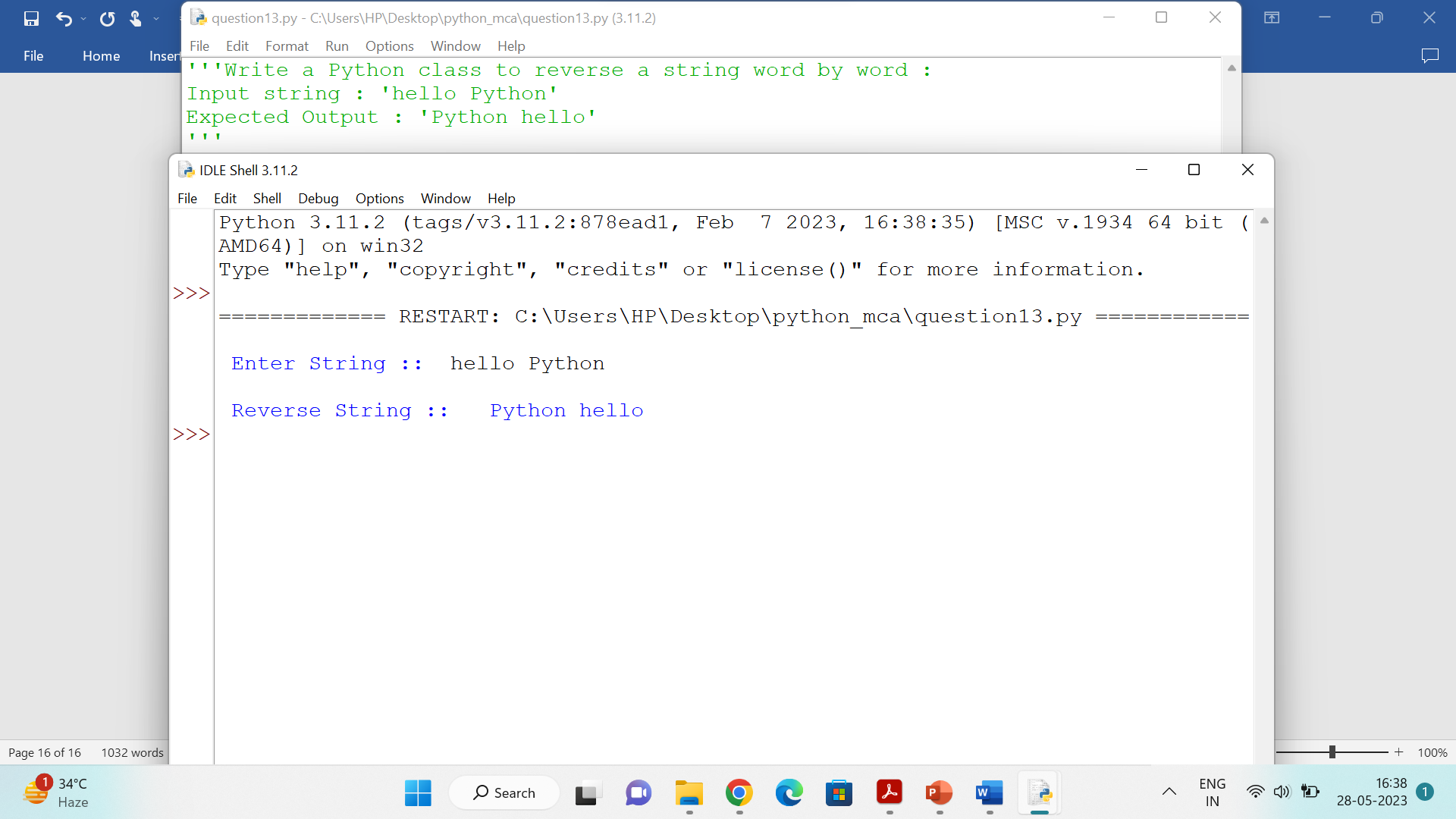
print("\n Reverse String :: ",self.string)

r1=reverse\_string()

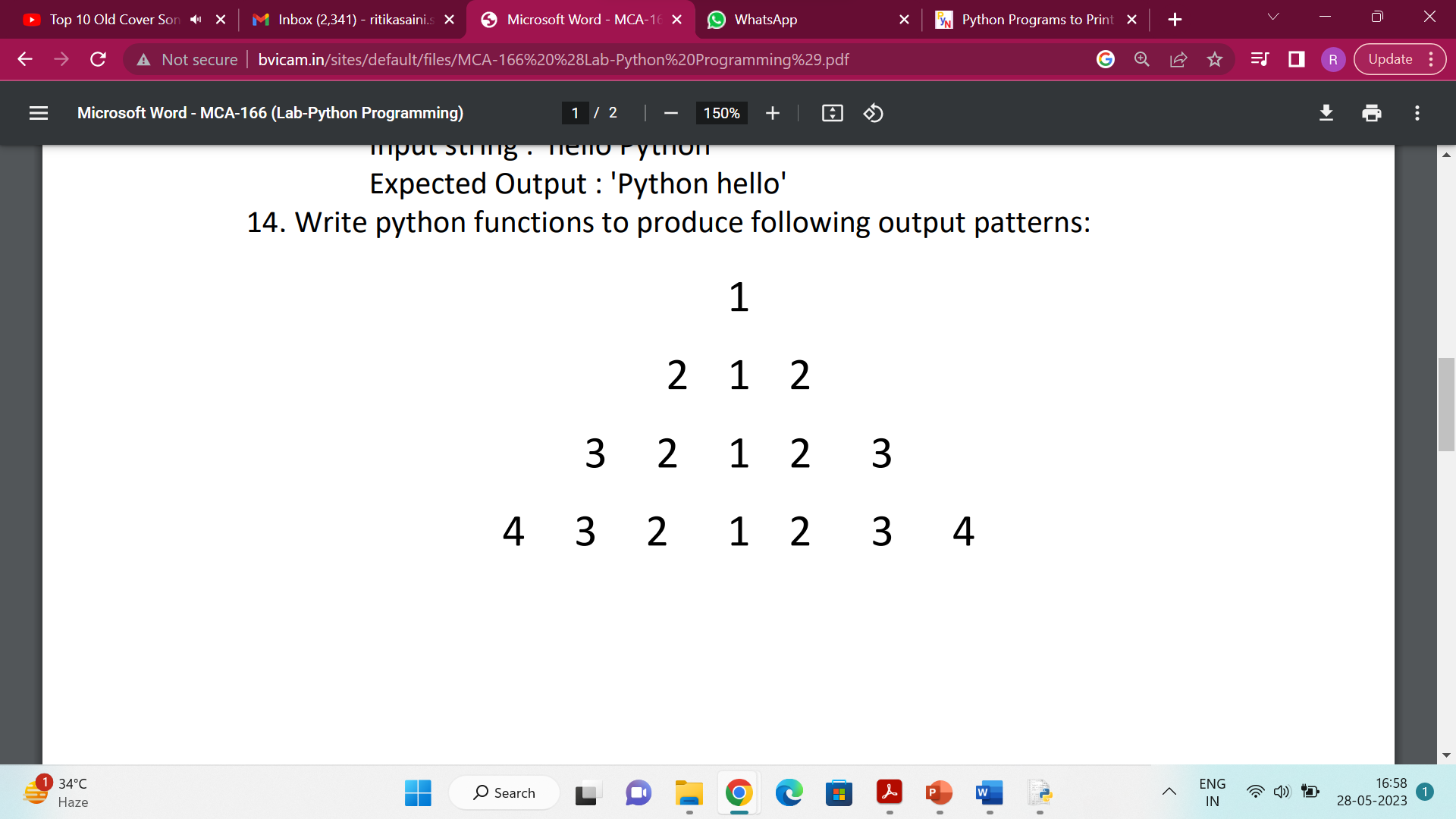
r1.get\_string()

r1.rev\_string()

r1.print()



**Question 14 . Write python functions to produce following output patterns:**

1. ****

**Solution**

print("\n Enter the size for the pattern :: ",end="")

n=int(input())

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

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

print(" ",end=" ")

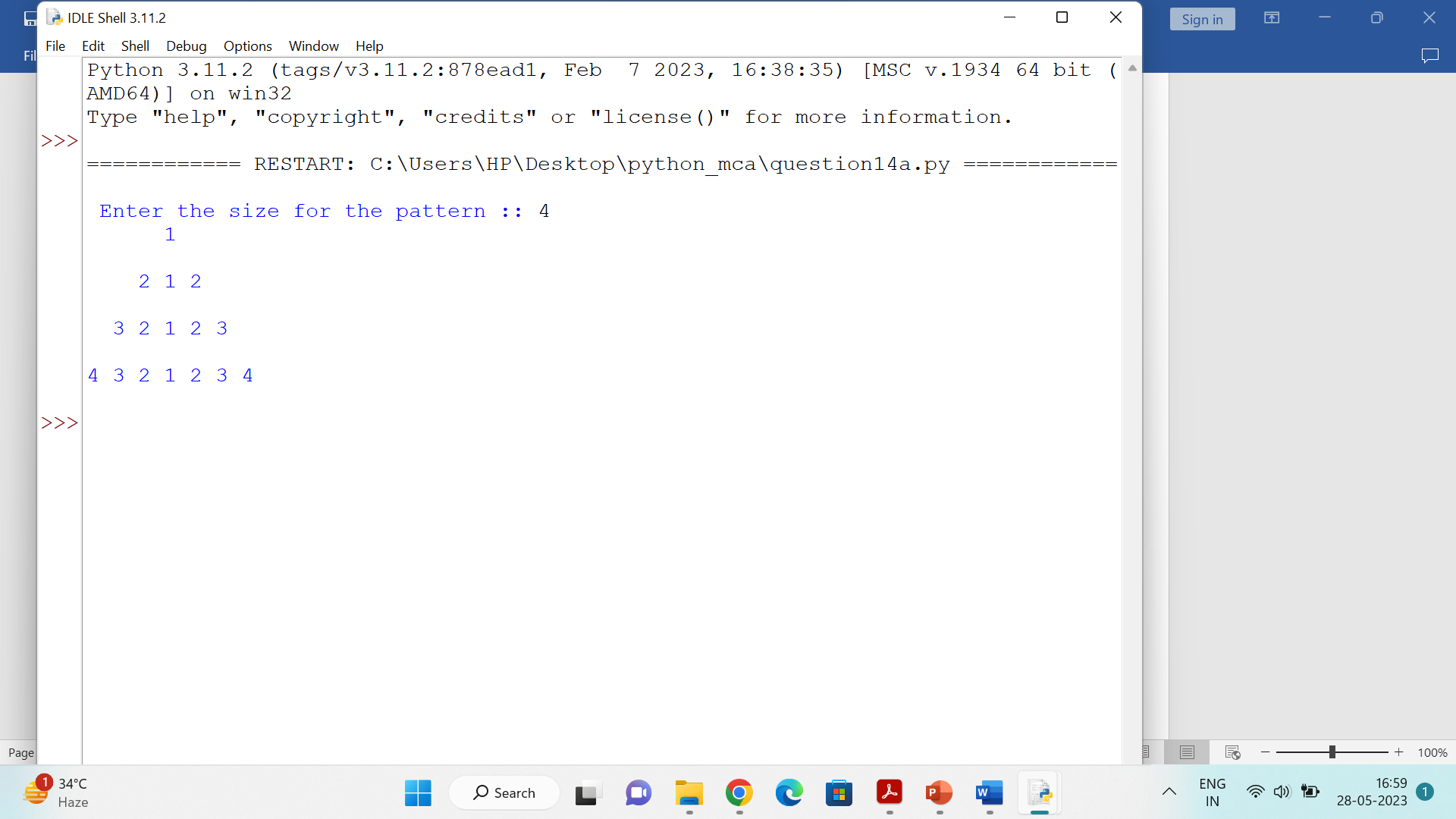
for k in range(i,0,-1):

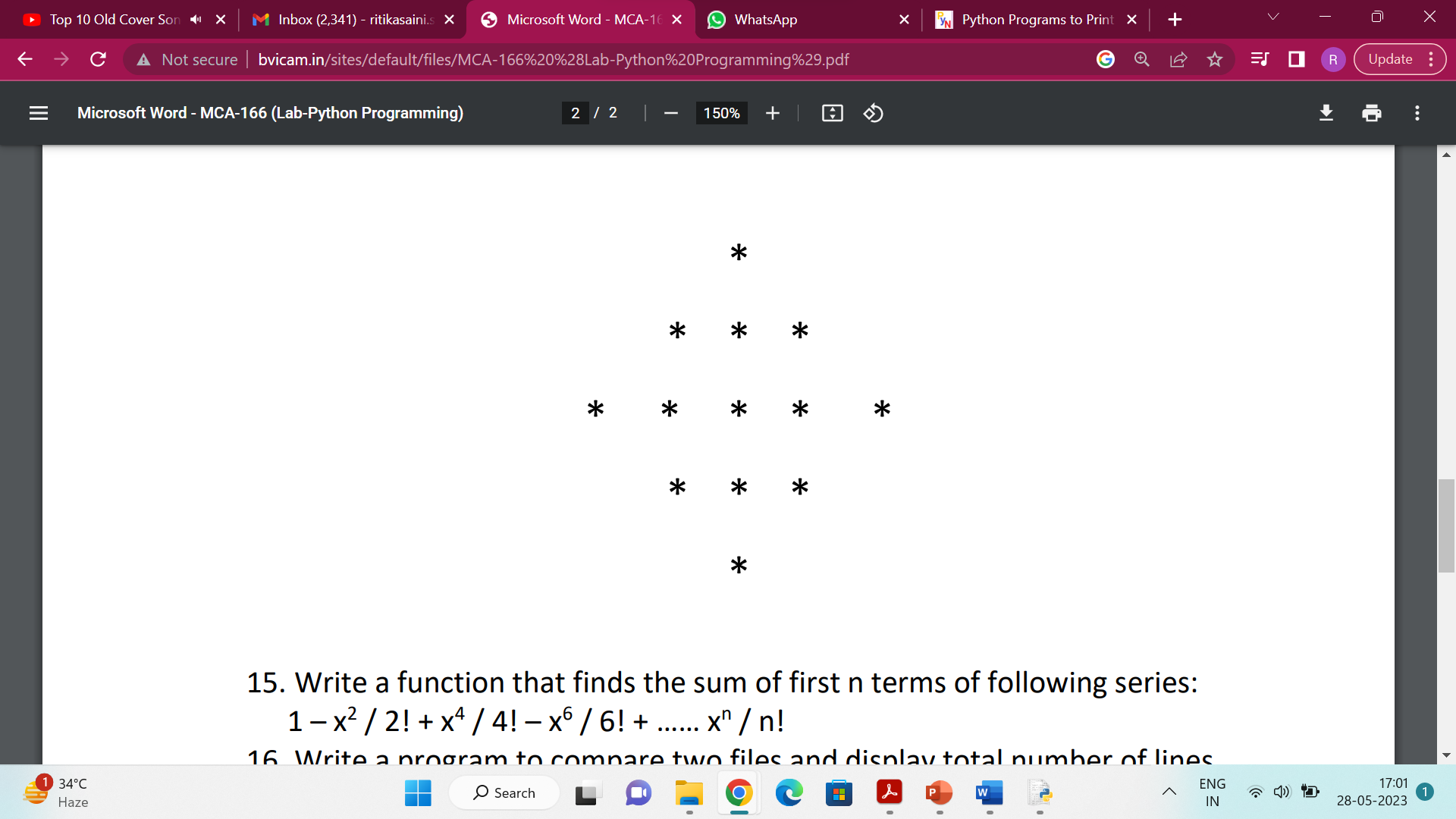
print(k,end=" ")

for j in range(2,i+1):

print(j,end=" ")

print("\n")



**b)**

**Solution**

print("\n Enter the size for the pattern :: ",end="")

n=int(input())

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

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

print(" ",end=" ")

for k in range(i,0,-1):

print("\*",end=" ")

for j in range(2,i+1):

print("\*",end=" ")

print("\n")

for i in range(n-1,0,-1):

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

print(" ",end=" ")

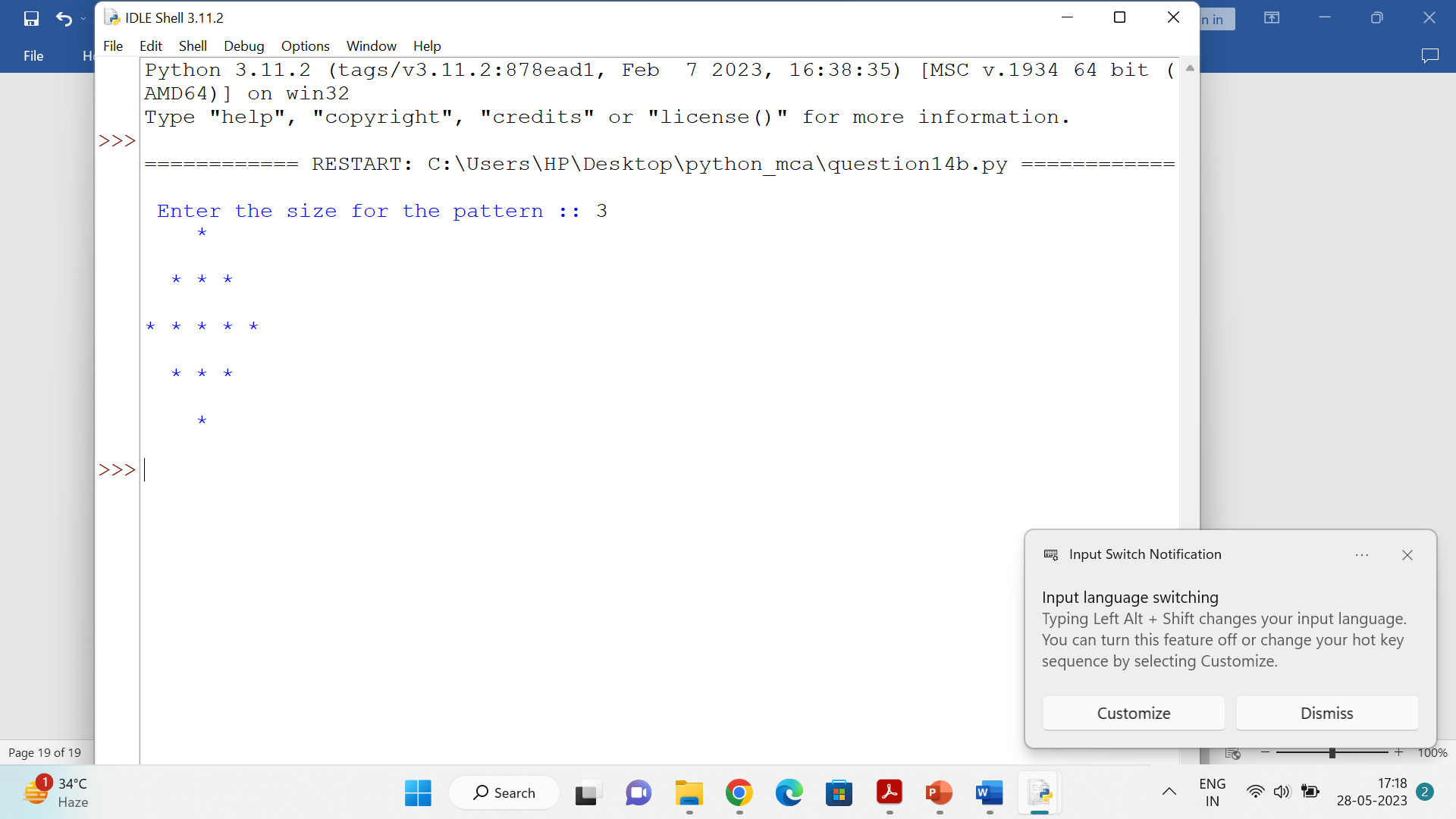
for k in range(i,0,-1):

print("\*",end=" ")

for j in range(2,i+1):

print("\*",end=" ")

print("\n")



**Question 15 Write a function that finds the sum of first n terms of following series: 1 – x2 / 2! + x4 / 4! – x6 / 6! + …… xn / n!**

**Solution**

import math

def q15( x , n ):

sum=1

s=1

p=2

for i in range(1,n):

f=1

for j in range(1,p+1):

f=f\*j

s=1 if i%2==0 else -1

sum = sum + s\* math.pow(x,p) / f

p+= 2

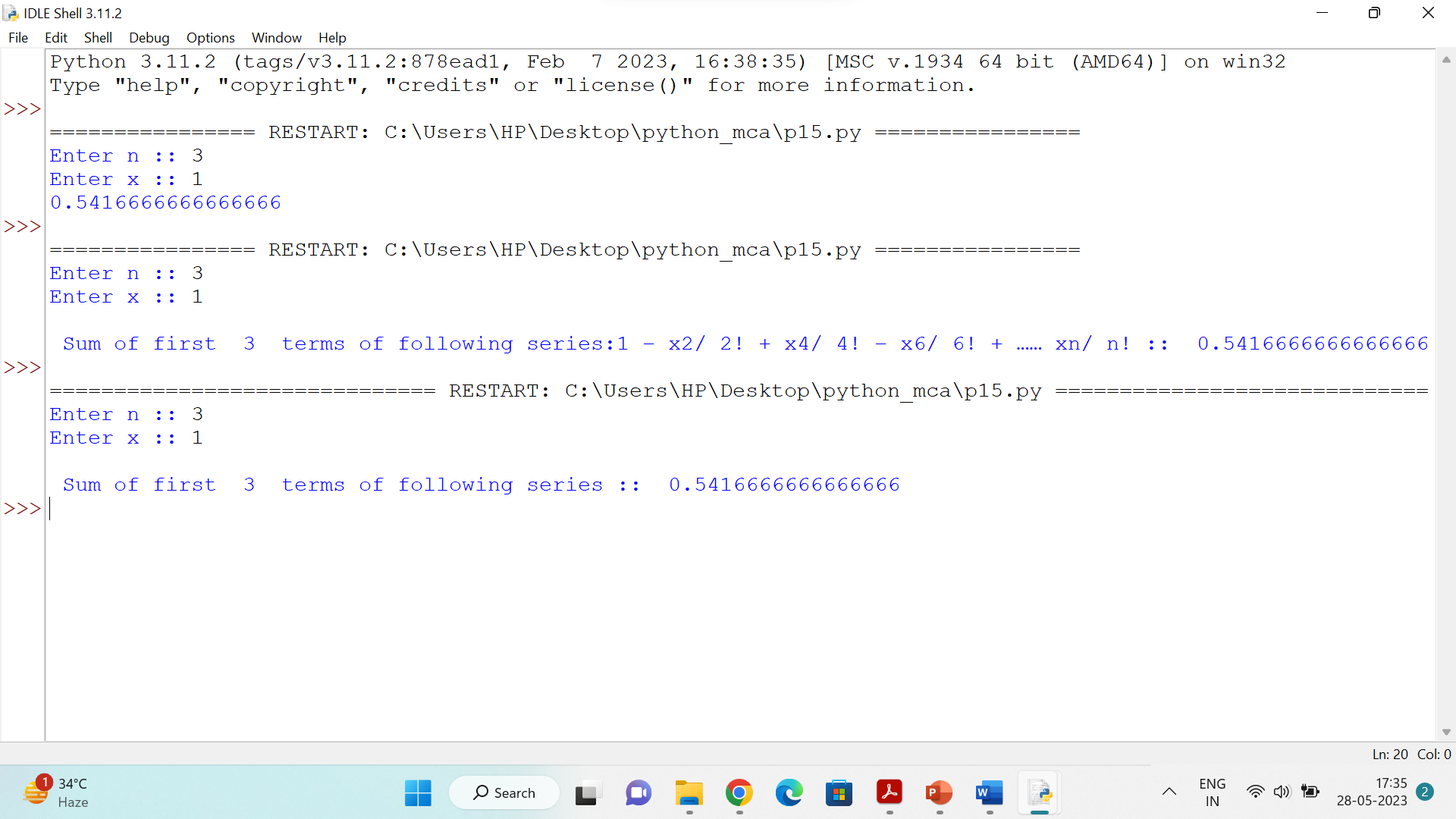
return sum

n=int(input("Enter n :: "))

x=int(input("Enter x :: "))

s=q15(x,n)

print("\n Sum of first ",n," terms of following series :: ",s)

****

**Question 16 Write a program to compare two files and display total number of lines in a file.**

**Solution**

print("\n Enter the name of the first file :: ",end="")

file1=input()

print("\n Enter the name of the second file :: ",end="")

file2=input()

with open(file1, 'r') as f1:

lines1 = len(f1.readlines())

print(f"\nNumber of lines in {file1}: {lines1}")

with open(file2, 'r') as f2:

lines2 = len(f2.readlines())

print(f"\nNumber of lines in {file2}: {lines2}")

if lines1 > lines2:

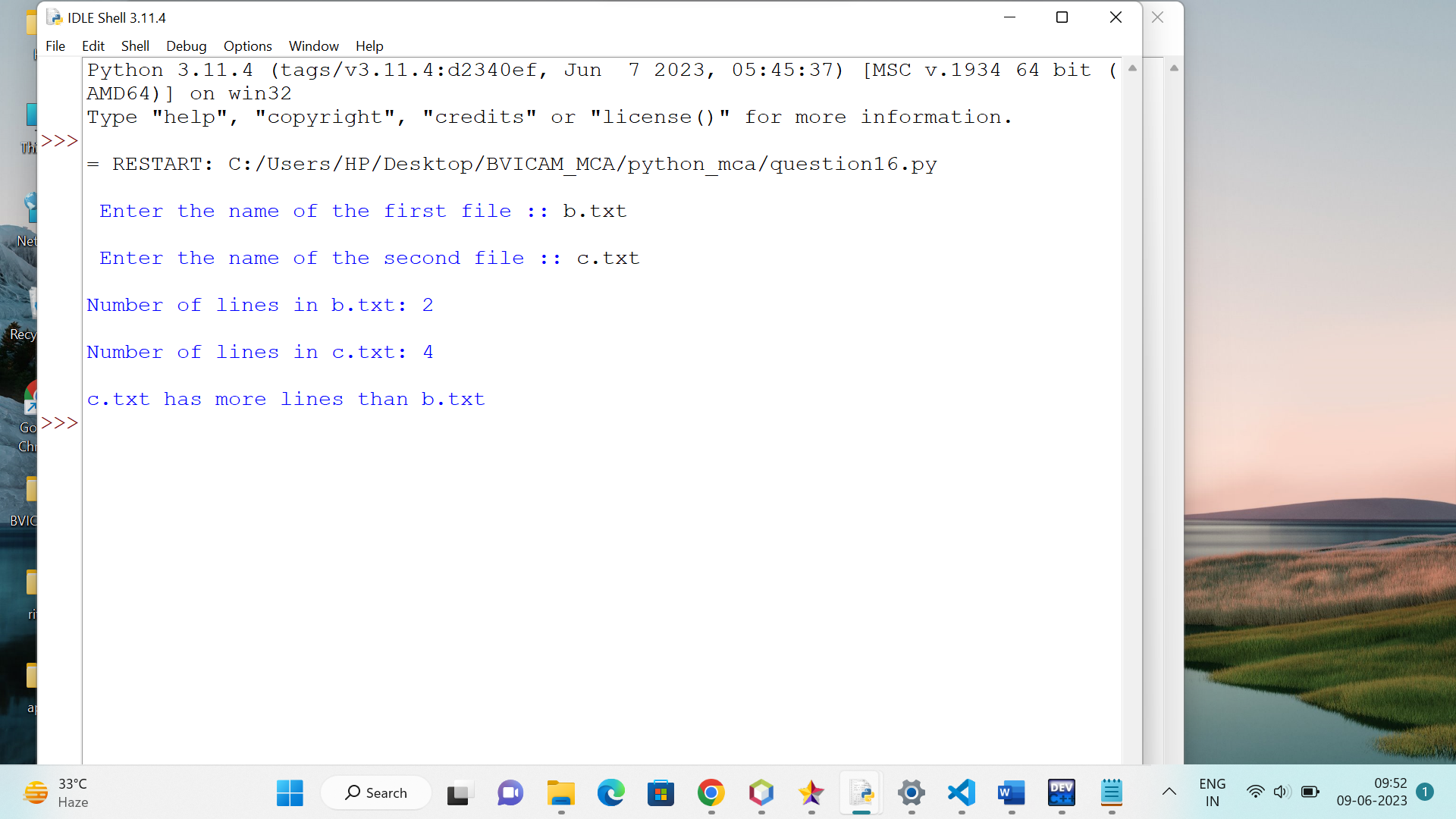
print(f"\n{file1} has more lines than {file2}")

elif lines2 > lines1:

print(f"\n{file2} has more lines than {file1}")

else:

print("\nThe two files have the same number of lines")

****

**Question17 Every time a vote is cast the name of the candidate is appended to the data structure. Print the names of candidates who received maximum vote in lexicographical order and if there is a tie print lexicographically smaller name.**

**Solution**

from collections import defaultdict

num\_votes = int(input("Enter the number of votes: "))

votes = defaultdict(int)

for \_ in range(num\_votes):

candidate\_name = input("Enter the name of the candidate: ")

votes[candidate\_name] += 1

max\_vote\_count = max(votes.values())

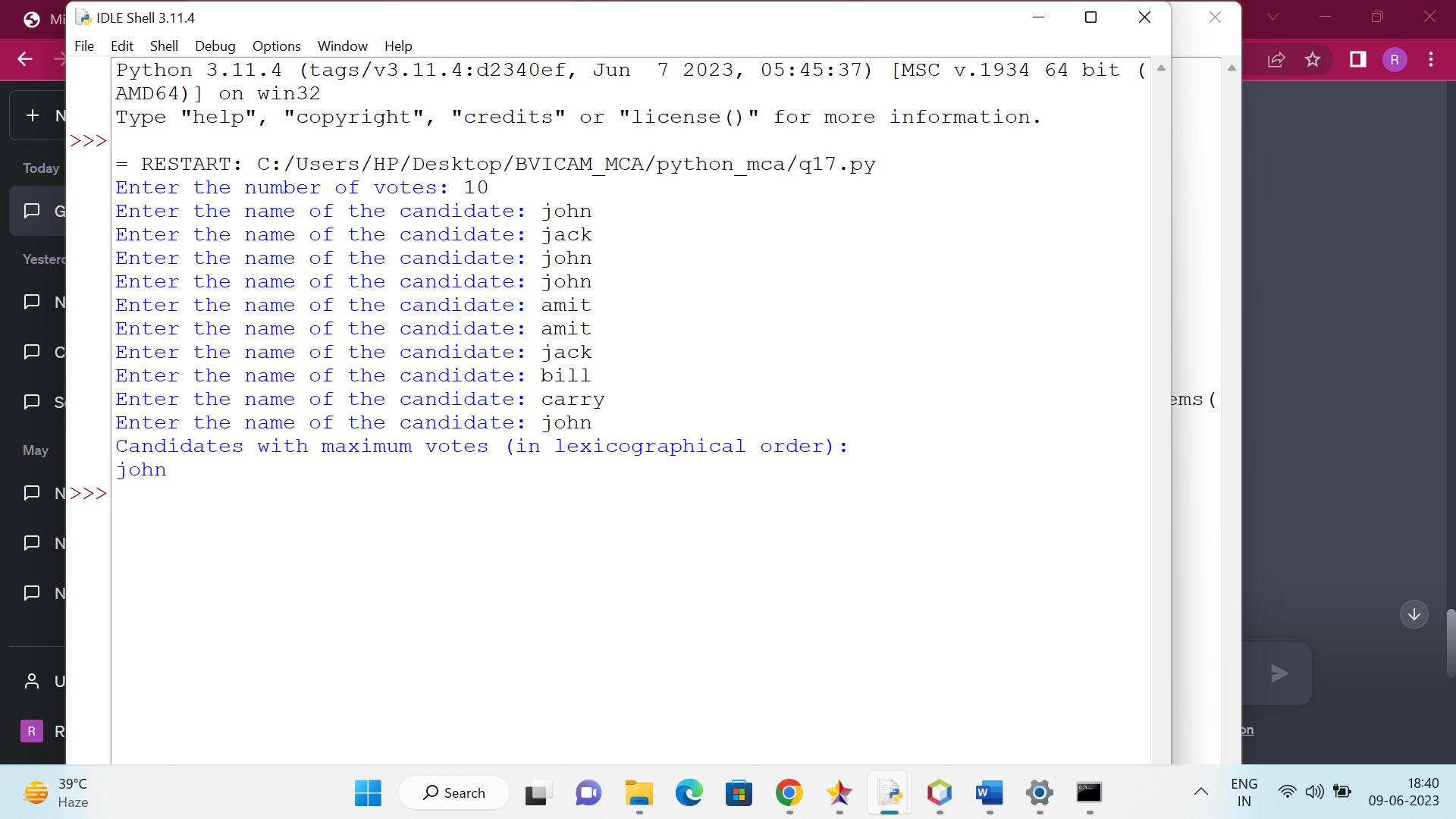
candidates\_with\_max\_votes = [candidate for candidate, vote\_count in votes.items() if vote\_count == max\_vote\_count]

candidates\_with\_max\_votes.sort()

print("Candidates with maximum votes (in lexicographical order):")

for candidate in candidates\_with\_max\_votes:

print(candidate)

****

**Question 18 Write a program to determine whether a given string has balanced parenthesis or not.**

**Solution**

def is\_balanced(string):

stack = []

opening\_brackets = ['(', '[', '{']

closing\_brackets = [')', ']', '}']

for char in string:

if char in opening\_brackets:

stack.append(char)

elif char in closing\_brackets:

if not stack:

return False

top = stack.pop()

if opening\_brackets.index(top) != closing\_brackets.index(char):

return False

return len(stack) == 0

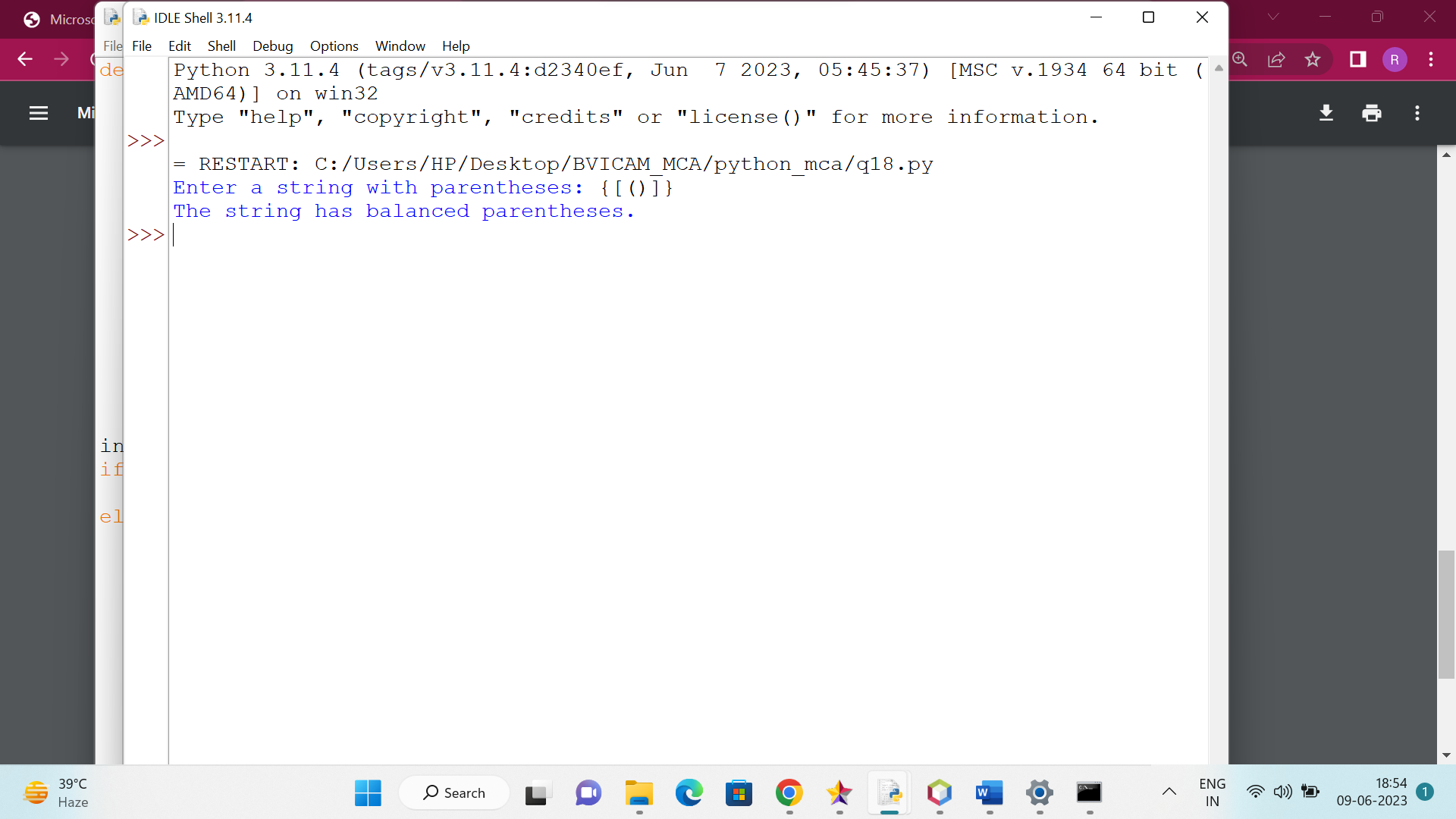
input\_string = input("Enter a string with parentheses: ")

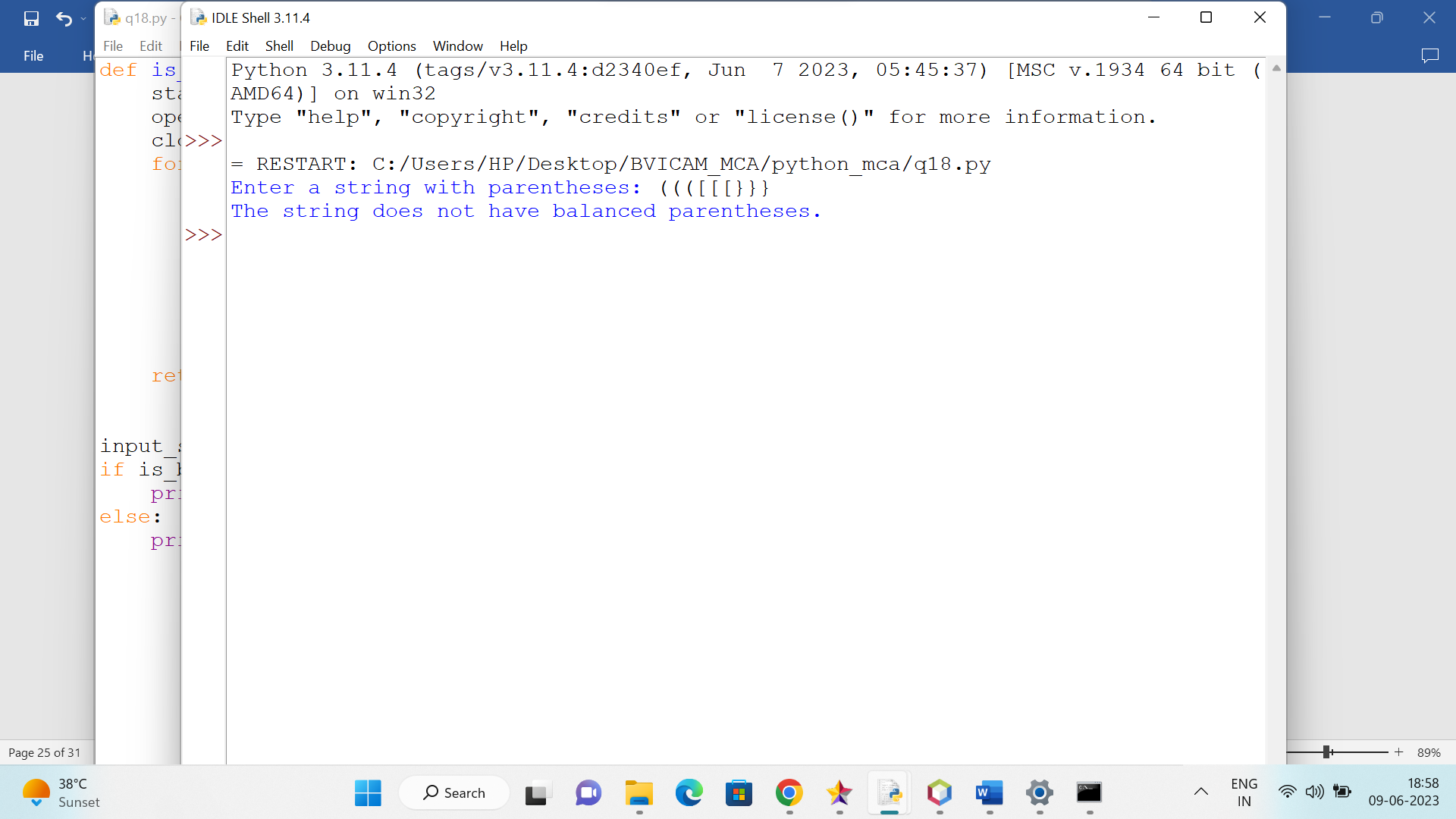
if is\_balanced(input\_string):

print("The string has balanced parentheses.")

else:

print("The string does not have balanced parentheses.")

****

****

**Question 19 Implement a python script to check the element is in the list or not by using Linear search & Binary search**

**Solution**

def linearsearch(arr,item):

for i in range(len(arr)):

if(arr[i]==item):

return True

return False

def binarysearch(arr,item):

low=0

high=len(arr)-1

while(low<=high):

mid=(low+high)//2

if(arr[mid]==item):

return True

elif(arr[mid]>item):

high=mid-1

else:

low=mid+1

return False

arr=[2,4,5,8,9,11,23,89]

item=90

print("\n Array :: ",arr)

print("\n Item to find :: ",item)

print("\n Linear Search ")

lresult=linearsearch(arr,item)

if(lresult==True):

print("\n Element Found !! ")

else:

print("\n Element Not Found !! ")

print("\n Binary Search ")

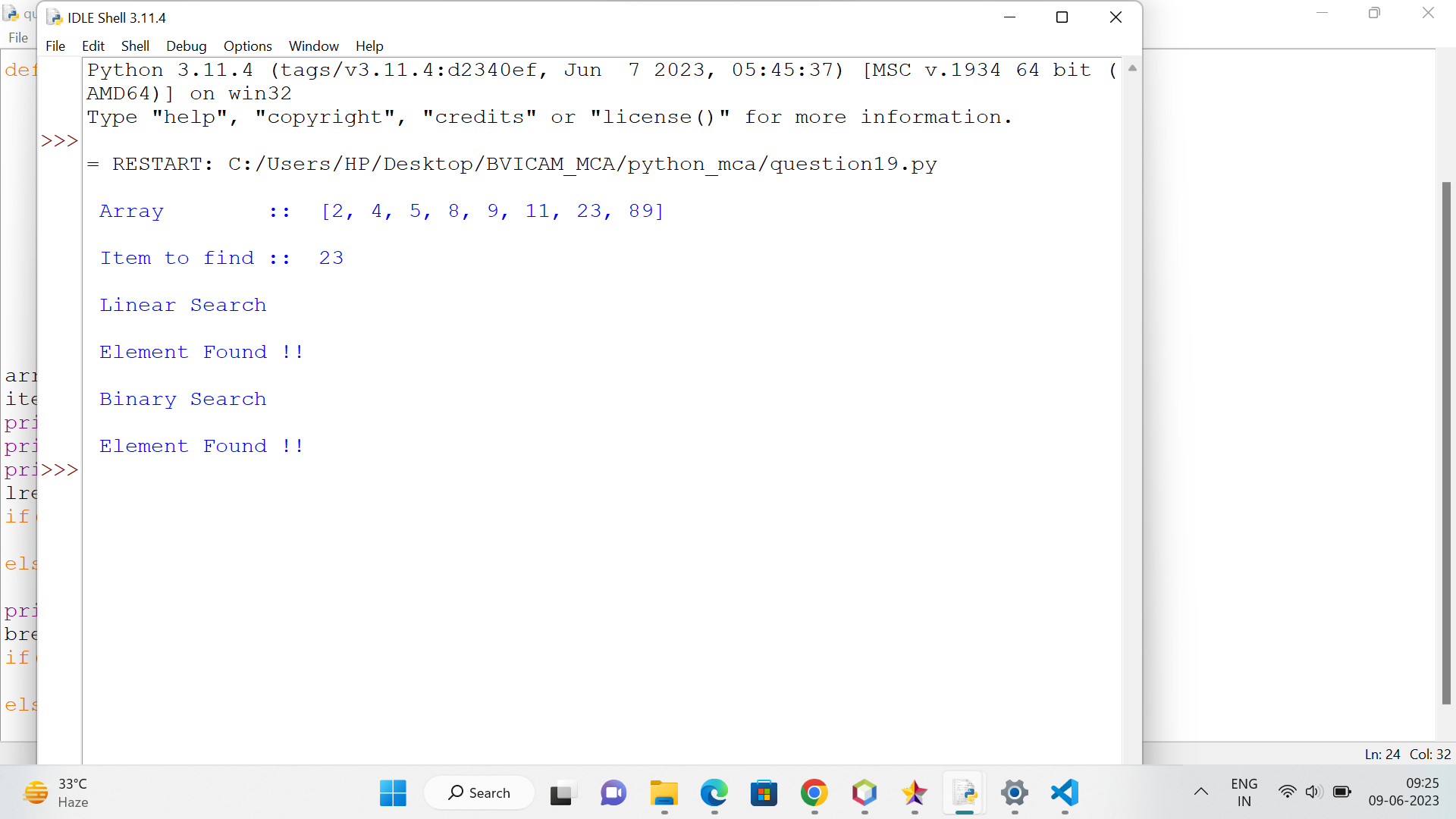
bresult=binarysearch(arr,item)

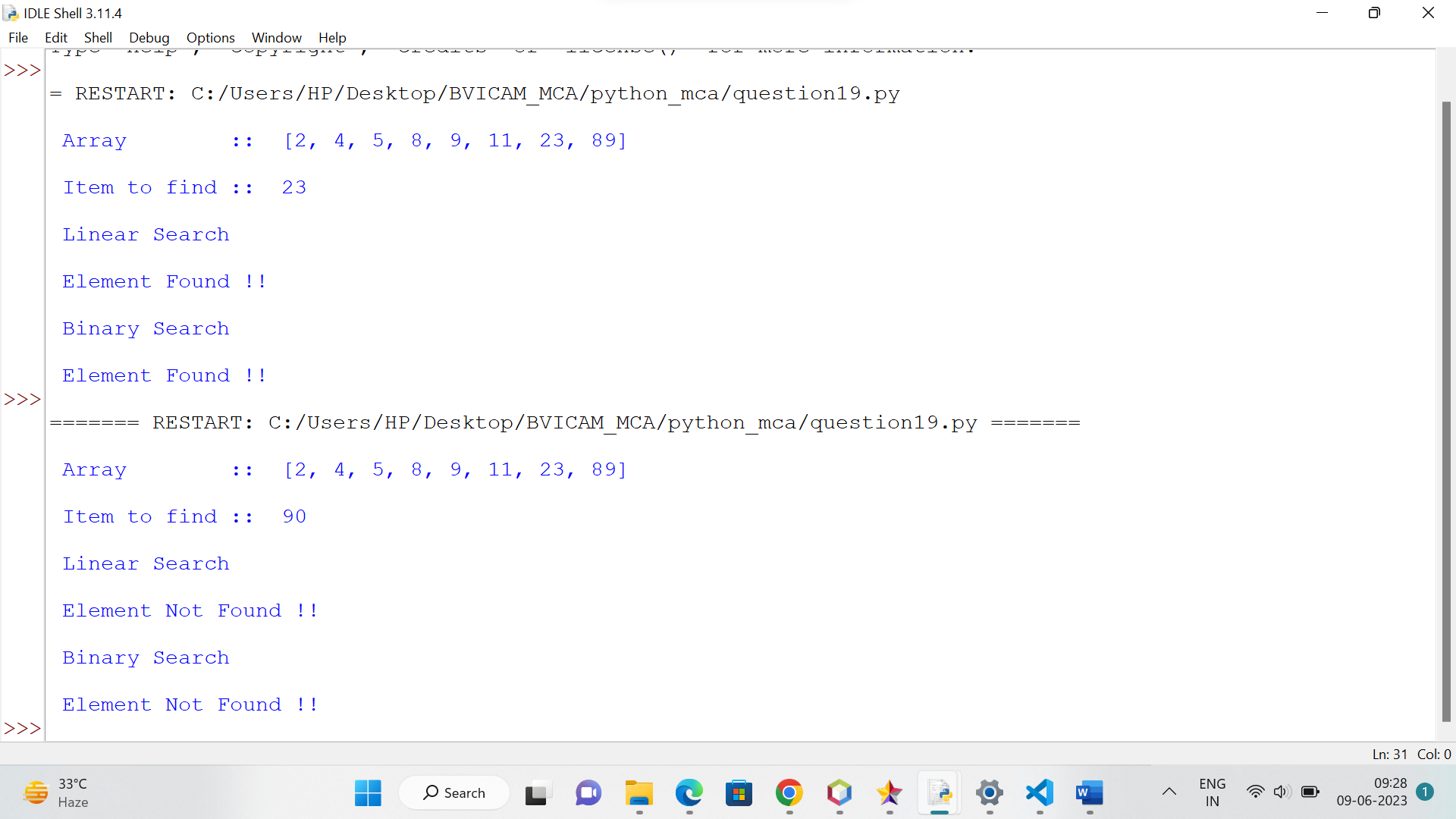
if(bresult==True):

print("\n Element Found !! ")

else:

print("\n Element Not Found !! ")

****

****

**Question 20 Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques.**

**Solution**

def bubble\_sort(arr):

n = len(arr)

for i in range(n - 1):

for j in range(n - 1 - i):

if arr[j] > arr[j + 1]:

arr[j], arr[j + 1] = arr[j + 1], arr[j]

def selection\_sort(arr):

n = len(arr)

for i in range(n - 1):

min\_idx = i

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

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

def insertion\_sort(arr):

n = len(arr)

for i in range(1, n):

key = arr[i]

j = i - 1

while j >= 0 and arr[j] > key:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left\_half = arr[:mid]

right\_half = arr[mid:]

left\_half = merge\_sort(left\_half)

right\_half = merge\_sort(right\_half)

return merge(left\_half, right\_half)

def merge(left, right):

result = []

i = j = 0

while i < len(left) and j < len(right):

if left[i] <= right[j]:

result.append(left[i])

i += 1

else:

result.append(right[j])

j += 1

while i < len(left):

result.append(left[i])

i += 1

while j < len(right):

result.append(right[j])

j += 1

return result

arr = [23,45,11,67,19]

print("Original array:", arr)

bubble\_sort(arr)

print("After bubble sort:", arr)

arr = [23,45,11,67,19]

selection\_sort(arr)

print("After selection sort:", arr)

arr = [23,45,11,67,19]

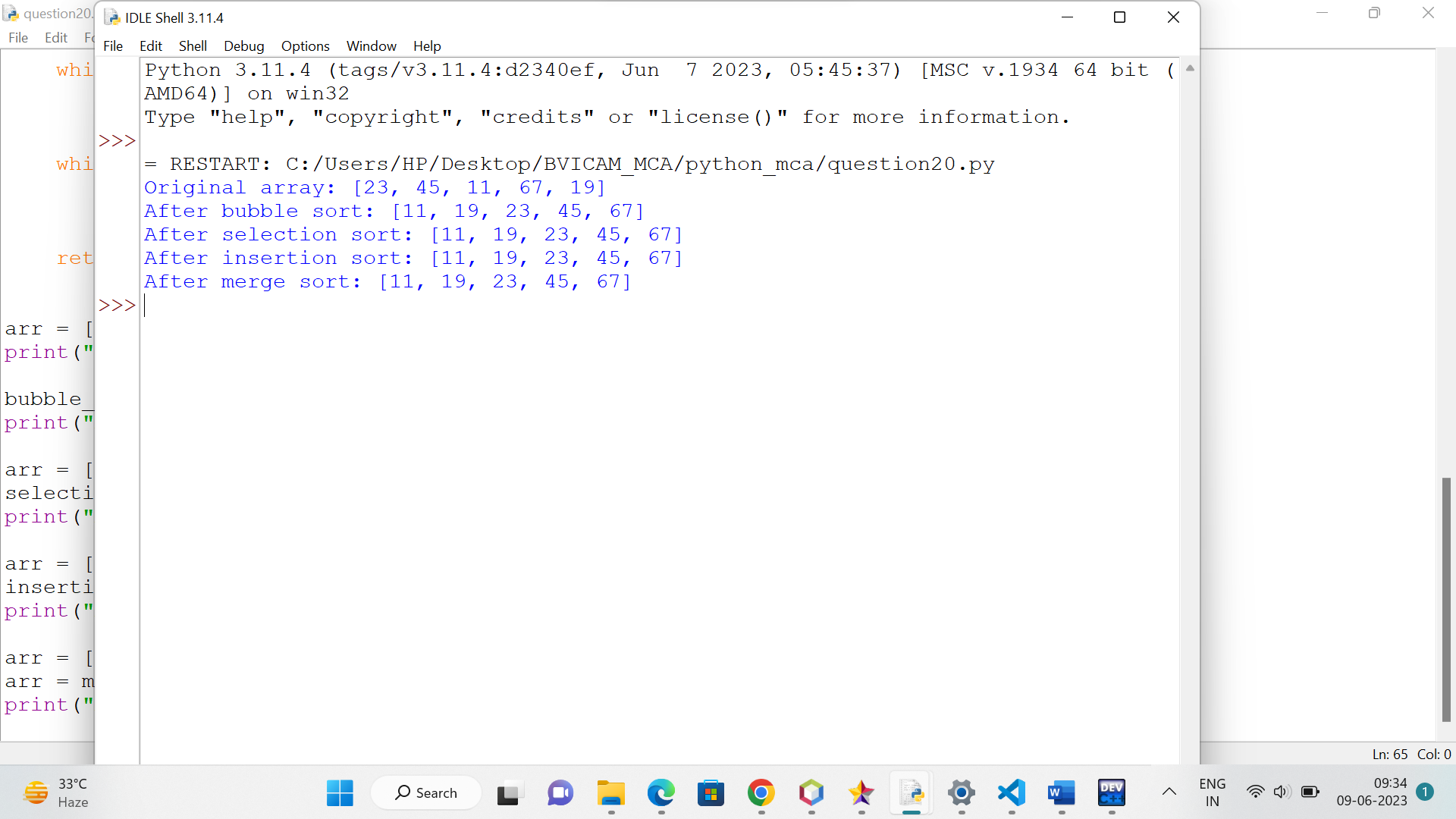
insertion\_sort(arr)

print("After insertion sort:", arr)

arr = [23,45,11,67,19]

arr = merge\_sort(arr)

print("After merge sort:", arr)

****

**Question 21 Create a menu driven program to perform various matrices operations.**

**Solution**

import numpy as np

def print\_menu():

print("Matrix Operations Menu")

print("1. Addition of Matrices")

print("2. Subtraction of Matrices")

print("3. Multiplication of Matrices")

print("4. Transpose of a Matrix")

print("5. Exit")

def read\_matrix():

rows = int(input("Enter the number of rows: "))

cols = int(input("Enter the number of columns: "))

matrix = []

for i in range(rows):

row = []

for j in range(cols):

element = int(input(f"Enter the element at position ({i+1},{j+1}): "))

row.append(element)

matrix.append(row)

return np.array(matrix)

def add\_matrices():

print("Addition of Matrices")

matrix1 = read\_matrix()

matrix2 = read\_matrix()

result = matrix1 + matrix2

print("Result:")

print(result)

def subtract\_matrices():

print("Subtraction of Matrices")

matrix1 = read\_matrix()

matrix2 = read\_matrix()

result = matrix1 - matrix2

print("Result:")

print(result)

def multiply\_matrices():

print("Multiplication of Matrices")

matrix1 = read\_matrix()

matrix2 = read\_matrix()

result = np.matmul(matrix1, matrix2)

print("Result:")

print(result)

def transpose\_matrix():

print("Transpose of a Matrix")

matrix = read\_matrix()

result = np.transpose(matrix)

print("Result:")

print(result)

while True:

print\_menu()

choice = input("Enter your choice (1-5): ")

if choice == '1':

add\_matrices()

elif choice == '2':

subtract\_matrices()

elif choice == '3':

multiply\_matrices()

elif choice == '4':

transpose\_matrix()

elif choice == '5':

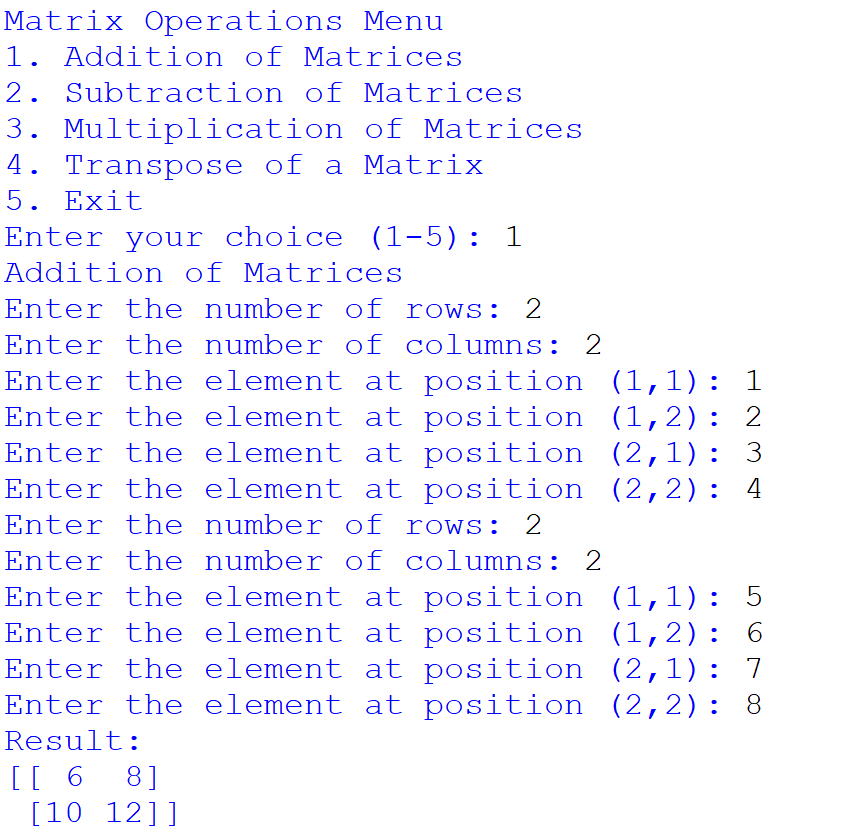
print("Exiting the program...")

break

else:

print("Invalid choice! Please enter a valid option (1-5).")

print() # Add a new line for better readability



**Question 22 Draw different graphs and plots in Python using Matplotlib Library.**

**Solution**

1. **Line Plot**

import matplotlib.pyplot as plt

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

y = [2, 4, 6, 8, 10]

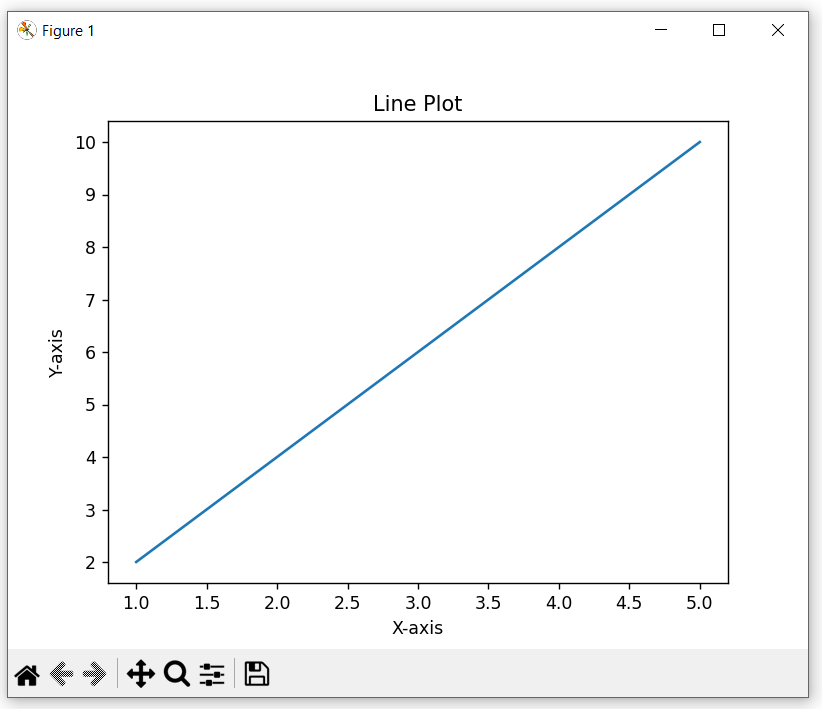
plt.plot(x, y)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Line Plot')

plt.show()



1. **Scatter Plot**

import matplotlib.pyplot as plt

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

y = [2, 4, 6, 8, 10]

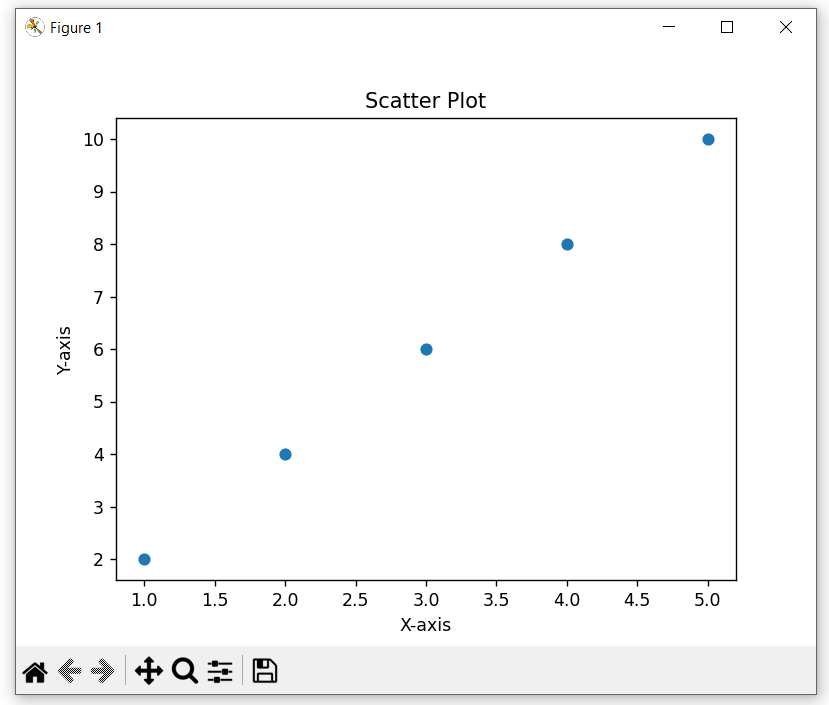
plt.scatter(x, y)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Scatter Plot')

plt.show()



1. **Bar Plot**

import matplotlib.pyplot as plt

x = ['A', 'B', 'C', 'D', 'E']

y = [10, 7, 5, 3, 8]

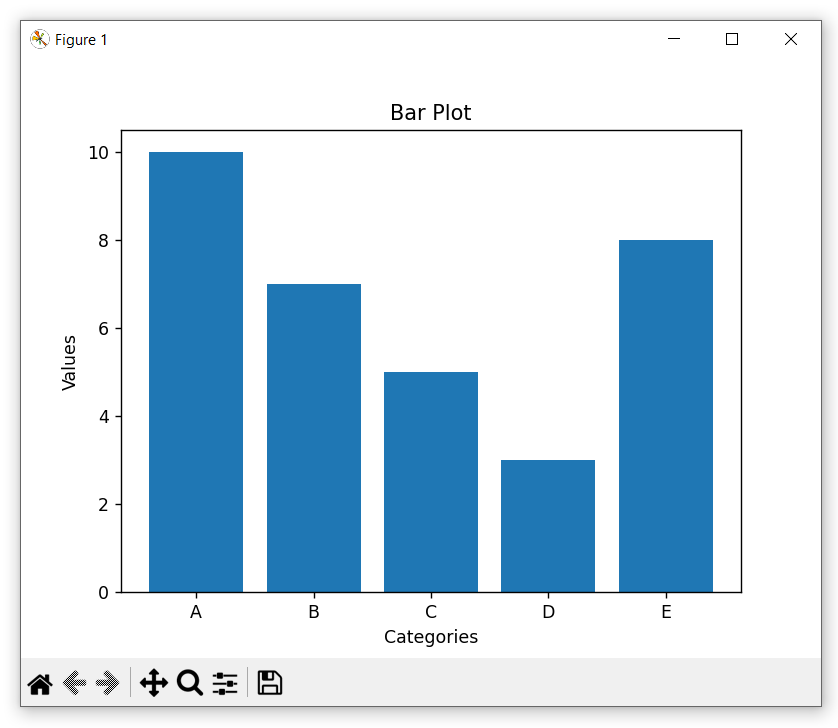
plt.bar(x, y)

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Bar Plot')

plt.show()



**Question 23 Write a python code for simple GUI calculator using Tk.**

**Solution**

import tkinter as tk

def button\_click(number):

current = display.get()

display.delete(0, tk.END)

display.insert(tk.END, current + str(number))

def button\_clear():

display.delete(0, tk.END)

def button\_equal():

expression = display.get()

try:

result = eval(expression)

display.delete(0, tk.END)

display.insert(tk.END, str(result))

except:

display.delete(0, tk.END)

display.insert(tk.END, "Error")

window = tk.Tk()

window.title("Simple Calculator")

display = tk.Entry(window, width=40, justify="right")

display.grid(row=0, column=0, columnspan=4)

buttons = [

("7", 1, 0), ("8", 1, 1), ("9", 1, 2), ("/", 1, 3),

("4", 2, 0), ("5", 2, 1), ("6", 2, 2), ("\*", 2, 3),

("1", 3, 0), ("2", 3, 1), ("3", 3, 2), ("-", 3, 3),

("0", 4, 0), (".", 4, 1), ("=", 4, 2), ("+", 4, 3)

]

for button\_text, row, col in buttons:

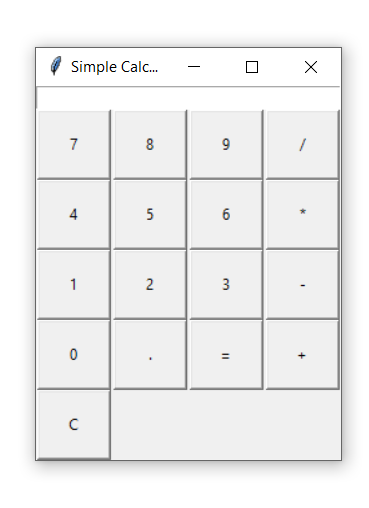
button = tk.Button(window, text=button\_text, width=7, height=3, command=lambda text=button\_text: button\_click(text))

button.grid(row=row, column=col)

clear\_button = tk.Button(window, text="C", width=7, height=3, command=button\_clear)

clear\_button.grid(row=5, column=0)

window.mainloop()



**Question 24 Write a python program to perform various database operations (create, insert, delete, update)**

**Solution**

**//Connection**

import mysql.connector

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12")

print(mydb)

if(mydb):

print("Connection successful")

else:

print("Connection not successful")

**//Create database**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

)

c = db.cursor()

c.execute("CREATE DATABASE employee\_db")

c.execute("SHOW DATABASES")

for i in c:

print(i)

c = db.cursor()

db.close()

**//Create Table**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

)

c = db.cursor()

employeetbl\_create = """CREATE TABLE `employee\_db`.`tb2employee` (

`empid` INT NOT NULL AUTO\_INCREMENT,

`empname` VARCHAR(45) NULL,

`department` VARCHAR(45) NULL,

`salary` INT NULL,

PRIMARY KEY (`empid`))"""

c.execute(employeetbl\_create)

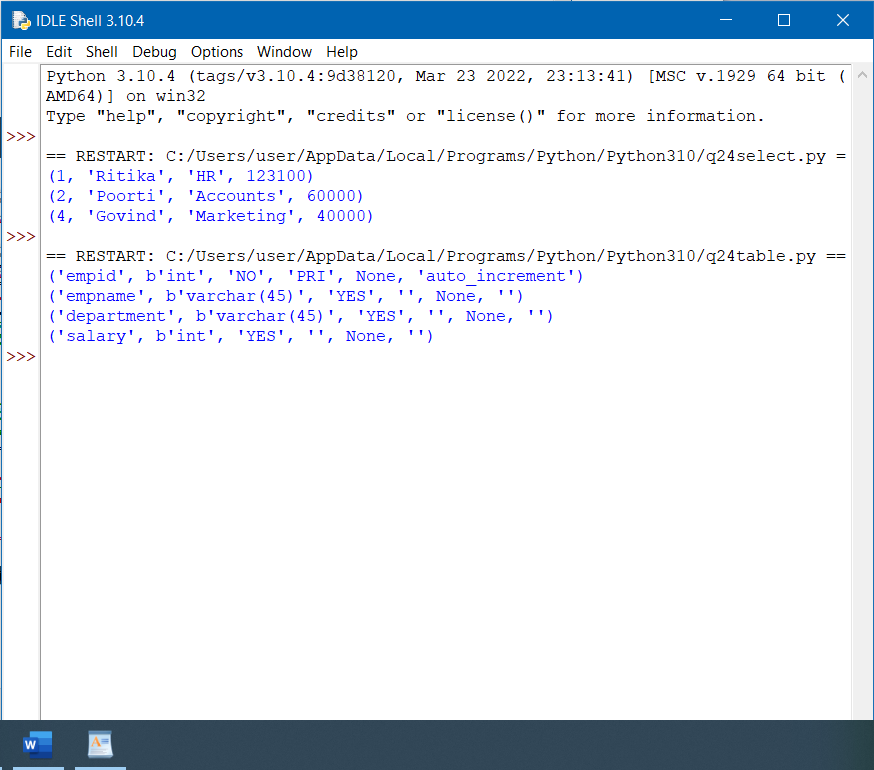
c = db.cursor()

c.execute("desc tblemployee")

for i in c:

print(i)

db.close()



**//Insert**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

)

c = db.cursor()

employeetbl\_insert = """INSERT INTO tb2employee (

empname,

department,

salary)

VALUES (%s, %s, %s)"""

data = [("Ritika", "HR", "100000"),

("Poorti", "Accounts", "60000"),

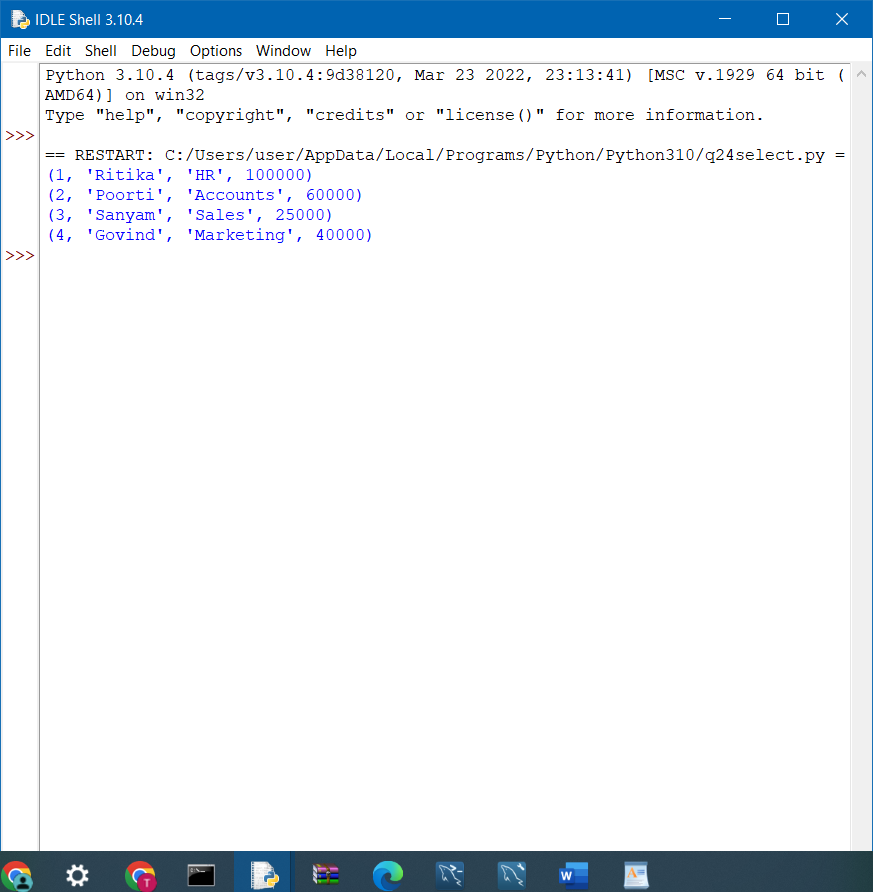
("Sanyam", "Sales", "25000"),

("Govind", "Marketing", "40000")]

c.executemany(employeetbl\_insert, data)

db.commit()

db.close()



**//Update**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

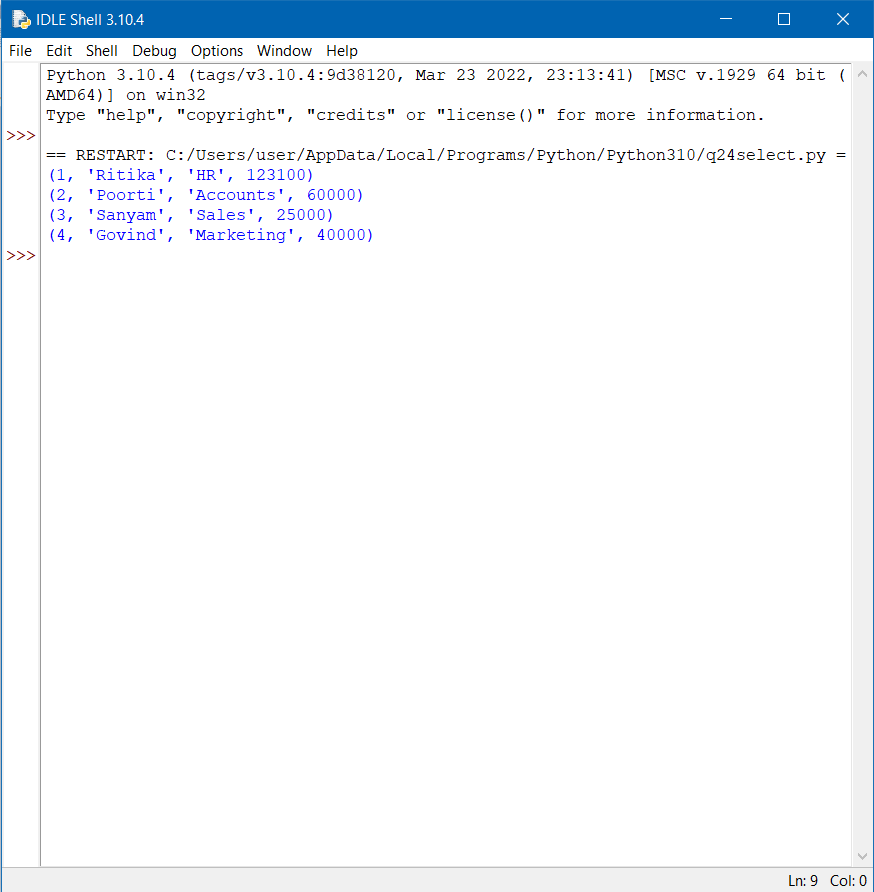
)

c = db.cursor()

employeetbl\_update = "UPDATE tb2employee SET salary = 123100 WHERE empid = 1"

c.execute(employeetbl\_update)

db.commit()



**//Delete**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

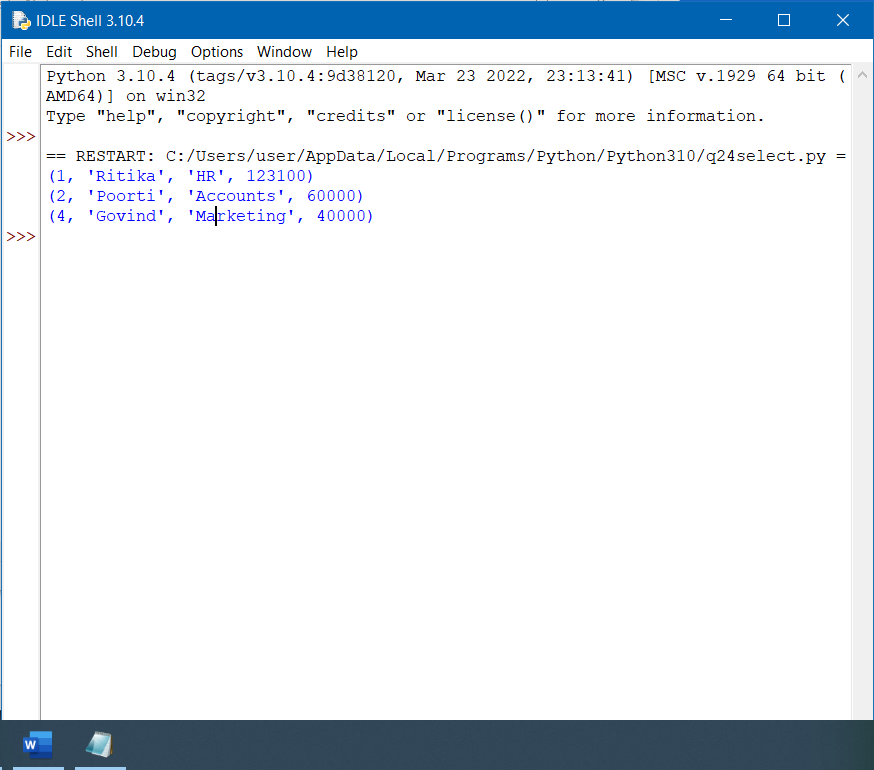
)

c = db.cursor()

employeetbl\_delete = "DELETE FROM tb2employee WHERE empid=3"

c.execute(employeetbl\_delete)

db.commit()



**//Select**

import mysql.connector

db = mysql.connector.connect(

host="localhost",

user="root",

passwd="12"

)

c = db.cursor()

c.execute(employeetbl\_select)

employee\_data = c.fetchall()

for e in employee\_data:

print(e)