

**NEW ERA PUBLIC SCHOOL, MAYAPURI**

**Computer Science**

**PRACTICAL FILE (2021-22)**

# Made by:

**Name: Vartika Sood**

**Class: XII Section: F**

**Roll No.: 40**

**CERTIFICATE**

**This is to certify that this Practical file is the bona fide work of Master/Miss Vartika Sood of Class XI F of New Era Public School, Mayapuri, New Delhi, carried out under my supervision.**

**Teacher’s Signature Principal’s Signature**

**INDEX**

|  |  |
| --- | --- |
| **S. No.** | **Program** |
|  | **Python Basics** |
| 1. | WAP to accept two numbers and one operator (+, -, \*, /, %) from the user and display the calculated result, according to the operator input from the user. |
| 2. | WAP that inputs a student’s marks in five subjects (out of 100), then prints the percentage of marks, and calculates grades as per the given grades. |
| 3. | WAP to print the roots of a quadratic equation ax2 + bx + c = 0 (where a≠0). The Program should check all possibilities of the discriminant d. |
| 4. | WAP that uses a ‘while’ loop to display all integers from -20 to 10 and a ‘for’ loop to display integers backwards from 15 to -10. |
| 5. | WAP to display sum and product of all natural numbers up to n, where n is the number of terms input from the user. |
| 6. | WAP to take an integer input N from the user. Print N Fibonacci numbers. Recall that Fibonacci series progresses as 0 1 1 2 3 5 8 13………….… |
| 7. | WAP to take an integer input N from the user and print the multiplication table of N. |
| 8. | WAP to find the sum of digits of a given number input from the user. |
| 9. | WAP that converts a decimal number into a binary number and vice-versa. |
| 10. | WAP that defines 3 user-defined functions rect(), tri(), and circ() to calculate the area of a rectangle, a triangle, and a circle respectively. The main Program should be menu-driven and accept appropriate data from the user and call these functions according to the user’s choice. |
| 11. | WAP to print the following pyramid patterns:  & & & & & &  & & & & &  & & & &  & & &  & &  &  & &  & & &  & & & &  & & & & &  & & & & & & |
| 12. | WAP that uses string slicing to find the reverse of a string input from the user and also check if the string is a palindrome or not. |
| 13. | WAP to input a string from the user and calculate the number of digits, alphabets, upper-case letters, lower-case letters, and special characters in the string. |
| 14. | WAP to input a string from the user and check for vowels and consonants. Remove all vowels, change the case of the consonants and generate the resultant string. |
| 15. | WAP to calculate the average length of a word in a string input by user. |
| 16. | WAP to display the number of vowels in a string. |
| 17. | WAP to find the length of the list and also find the index value of the elements. |
| 18. | WAP to use some list functions and methods |
| 19. | WAP to use list comprehension. |
| 20. | WAP to create a nested list to store details of students. |
| 21. | WAP to input n numbers, store them and then print the maximum, minimum, sum and mean of these elements. |
| 22. | WAP to count frequency of elements in a tuple. |
| 23. | WAP to input any 2 tuples and swap their values. |
| 24. | WAP to input total number of sections and streams in class 11 and display on screen. |
| 25. | WAP to count the frequency of elements using a dictionary. |
| 26. | WAP to iterate over dictionaries using for loops. |
| 27. | WAP to get minimum and maximum value in dictionary. |
| 28. | WAP to concatenate 3 Python dictionaries. |
| 29. | WAP to add a key value pair to a dictionary. |
| 30. | WAP to check if a given key exists in a dictionary or not. |
| 31. | WAP to create a dictionary with key as first character and value as words starting with that character. |
| 32. | WAP using dictionary to print the name and salary of employee. |
| 33. | WAP to show the use of math module. |
| 34. | WAP to make a guessing game using random module. |
| 35. | WAP to make a calculation game. |
| 36. | WAP to show the use of statistics module. |
|  | **FUNCTIONS** |
| 37. | WAP to make a function to find the average of numbers. |
| 38. | WAP to make a function to find cube of a number given by the user. |
| 39. | WAP to make a function to find the area of the circle. |
| 40. | WAP to make a function to make a calculator. |
| 41. | WAP to make a function to try different kinds of arguments. |
| 42. | WAP to use different functions of random module. |
| 43. | WAP to calculate sum of the digits of a random three-digit number. |
| 44. | WAP that fills a list with numbers. |
| 45. | WAP to perform binary search using randint(). |
| 46. | WAP to choose a random direction. |
| 47. | WAP to shuffle a list using random module |
| 48. | WAP to make a function to count the vowels in a string. |
| 49. | WAP to make a higher and lower number guessing game. |
| 50. | WAP to count the frequency of a character in a string. |
| 51. | WAP to make a function having tuple as the argument. |
| 52. | WAP to determine whether the elements in the tuple are even or odd. |
| 53. | WAP to pass student record as a dictionary to a function and update their marks. |
| 54. | WAP to pass dictionary to a function with list of elements as keys and frequency of its occurrence as value and return as a dictionary. |
| 55. |  |
| 56. |  |
| 57. |  |
| 58. |  |
| 59. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Python**

**Basics**

**Program:**

Write a Python Program to accept two numbers and one operator (+, -, \*, /, %) from the user and display the calculated result, according to the operator input from the user.

**Source code:**

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

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

c=input("Enter the Operator +,-,\*,/, %: ")

if c=='+':

print("The Result is : ",a+b)

elif c=='-':

print("The Result is : ",a-b)

elif c=='\*':

print("The Result is : ",a\*b)

elif c=='/':

print("The Result is : ",a/b)

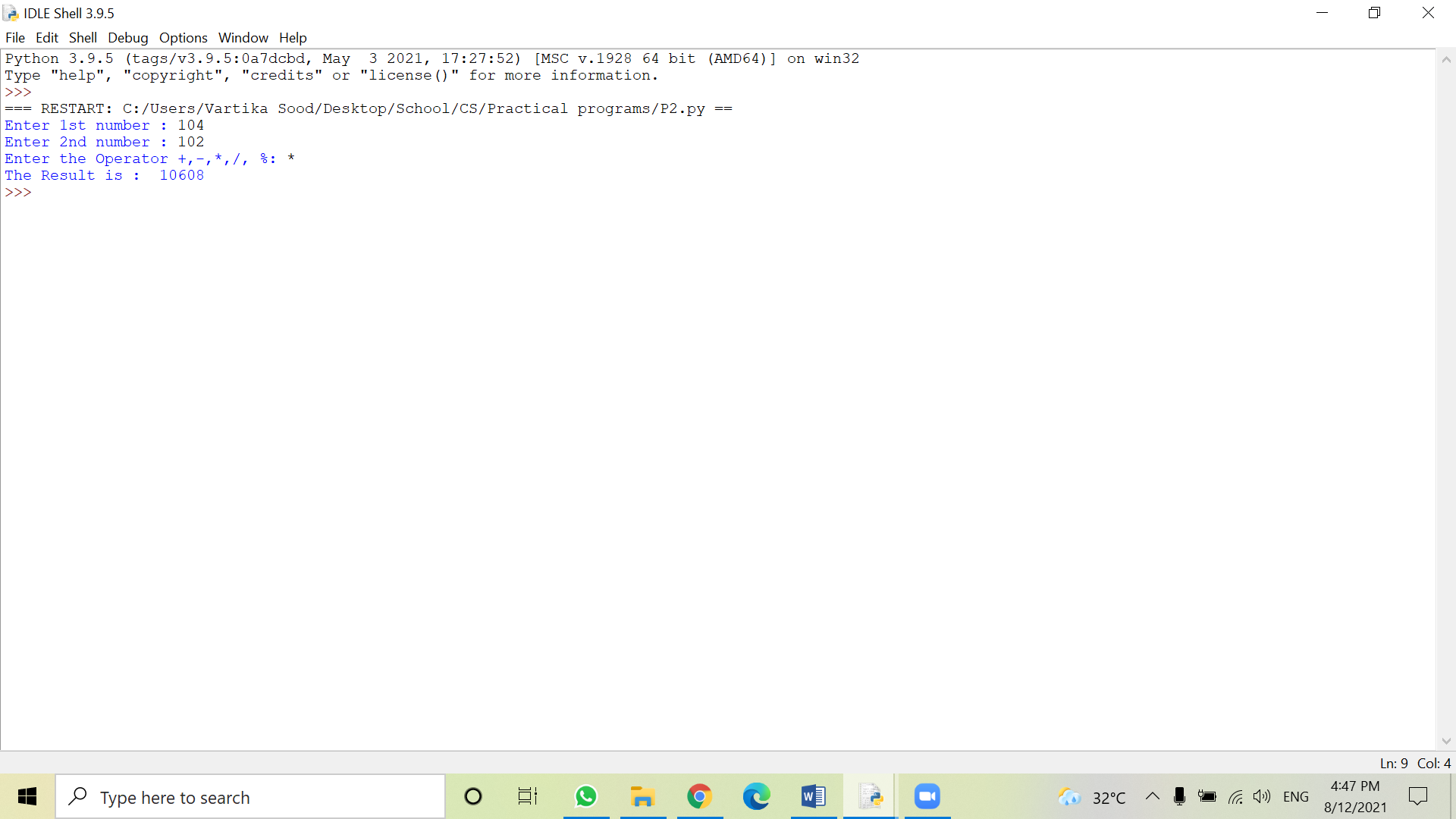
elif c=='%':

print("The Result is : ",a%b)

else:

print("Wrong Operator Entered")

**Output:**



**Program:**

Write a Python Program that inputs a student’s marks in five subjects (out of 100), then prints the percentage of marks, and calculates grades as per the following rules:

Marks Grade

85% and above A

75-84% B

65-74% C

50-64% D

40-49% E

39% and below Fail

**Source code:**

sub1=int(input("Enter marks of the first subject: "))

sub2=int(input("Enter marks of the second subject: "))

sub3=int(input("Enter marks of the third subject: "))

sub4=int(input("Enter marks of the fourth subject: "))

sub5=int(input("Enter marks of the fifth subject: "))

avg=(sub1+sub2+sub3+sub4+sub4)/5

if(avg>=85):

print("Grade: A")

elif(avg>=75&avg<84):

print("Grade: B")

elif(avg>=65&avg<74):

print("Grade: C")

elif(avg>=50&avg<64):

print("Grade: D")

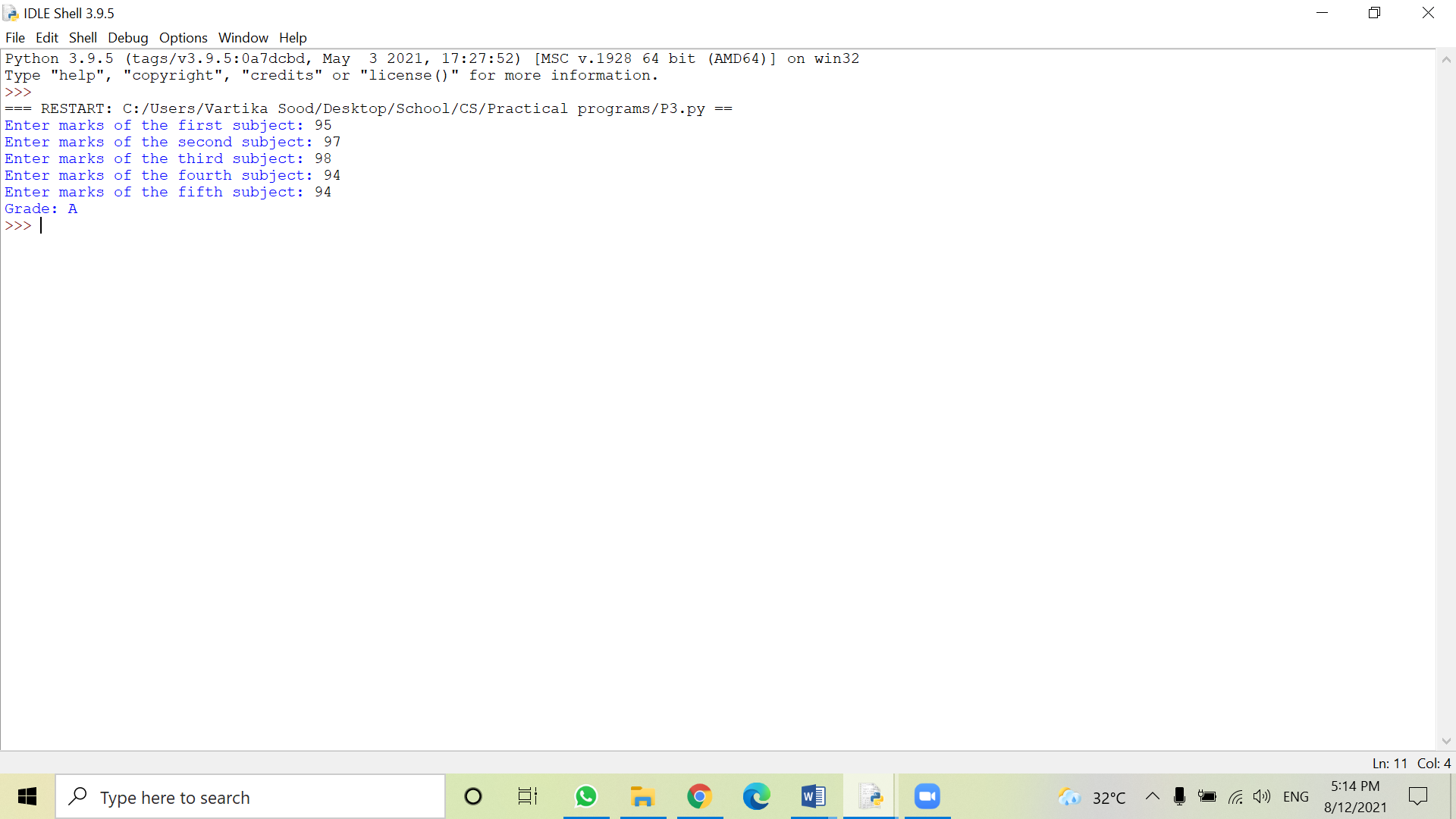
elif(avg>=40&avg<49):

print("Grade: D")

else:

print("Grade: F")

**Output:**



**Program:**

Write a Python Program to print the roots of a quadratic equation ax2 + bx + c = 0 (where a≠0). The Program should check all possibilities of the discriminant d.

**Source code:**

print("Quadratic function : (a \* x^2) + b\*x + c")

a = float(input("a: "))

b = float(input("b: "))

c = float(input("c: "))

r = b\*\*2 - 4\*a\*c

if r > 0:

num\_roots = 2

x1 = (((-b) + sqrt(r))/(2\*a))

x2 = (((-b) - sqrt(r))/(2\*a))

print("There are 2 roots: %f and %f" % (x1, x2))

elif r == 0:

num\_roots = 1

x = (-b) / 2\*a

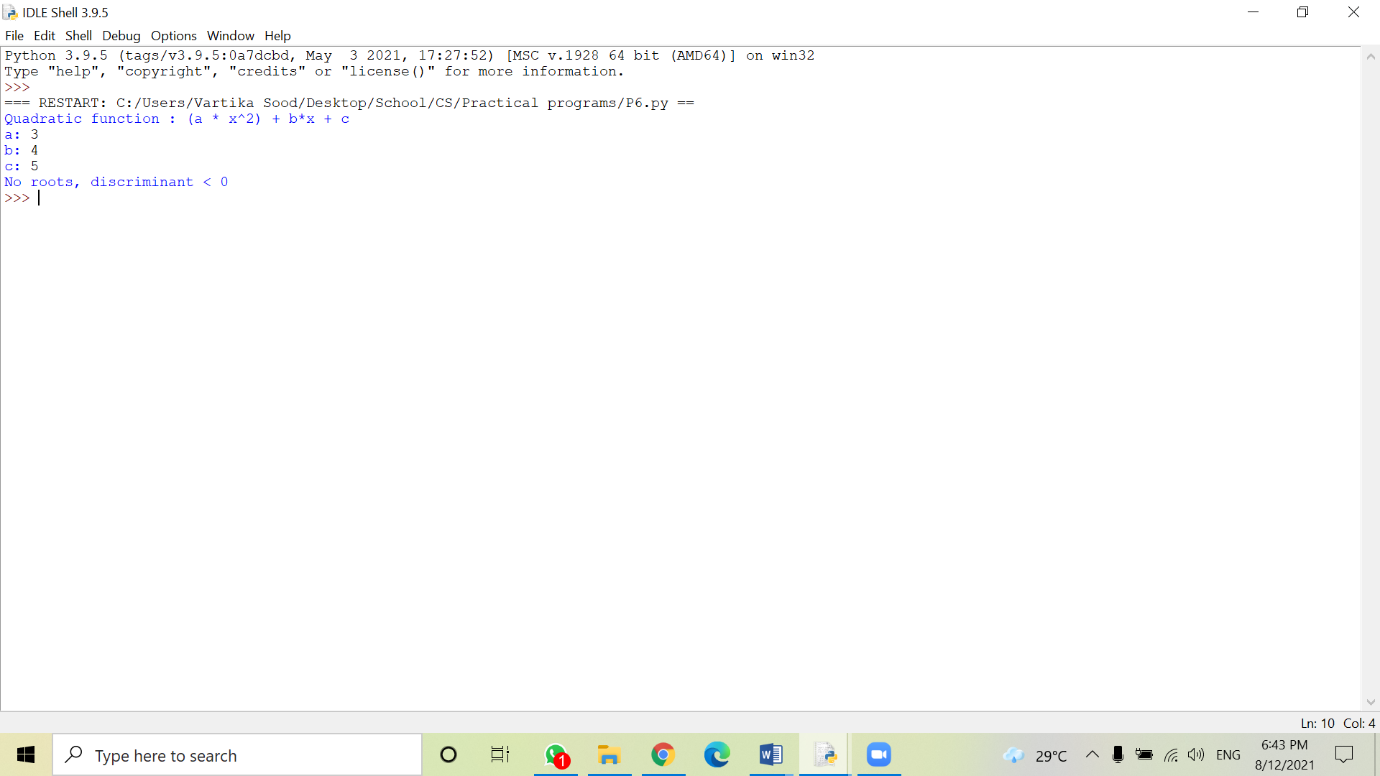
print("There is one root: ", x)

else:

num\_roots = 0

print("No roots, discriminant < 0")

**Output:**



**Program:**

Write a Python Program that uses a ‘while’ loop to display all integers from -20 to 10 and a ‘for’ loop to display integers backwards from 15 to -10.

**Source code:**

i=-20

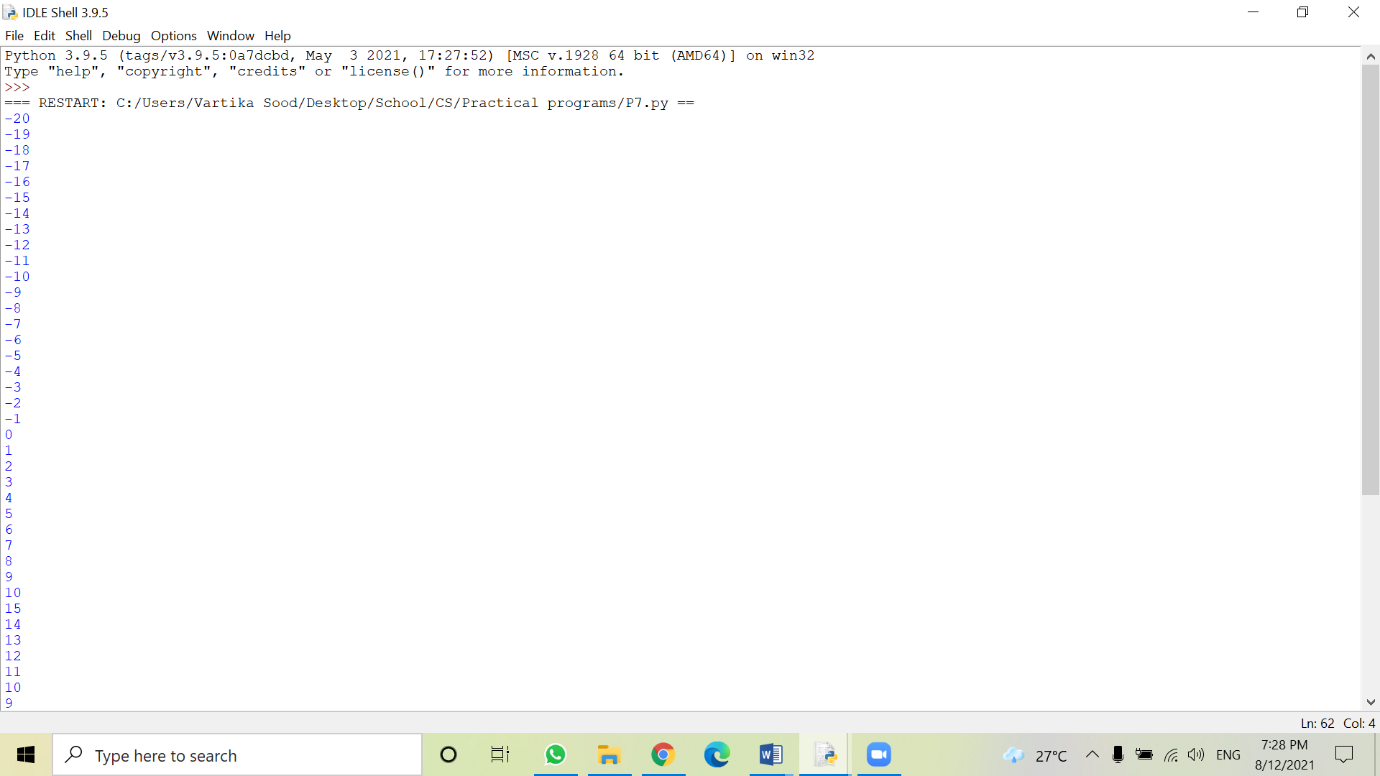
while(i<=10):

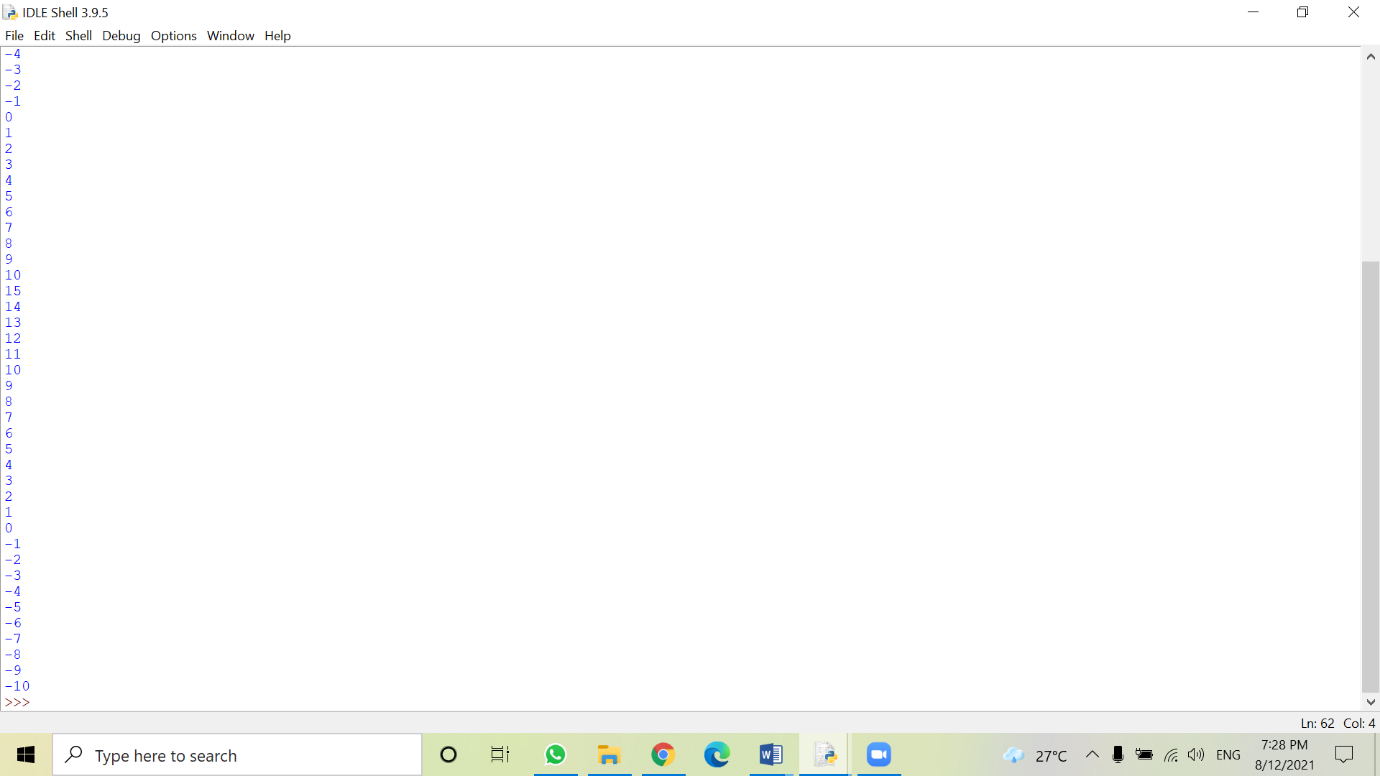
print(i)

i+=1

for j in range(15,-11, -1):

print(j)

**Output:**



**Program:**

Write a Python Program to display sum and product of all natural numbers up to n, where n is the number of terms input from the user.

**Source code:**

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

factorial,sum=1,0

if n<0:

print("Invalid Number, Please enter positive number")

elif n==0:

pass

else:

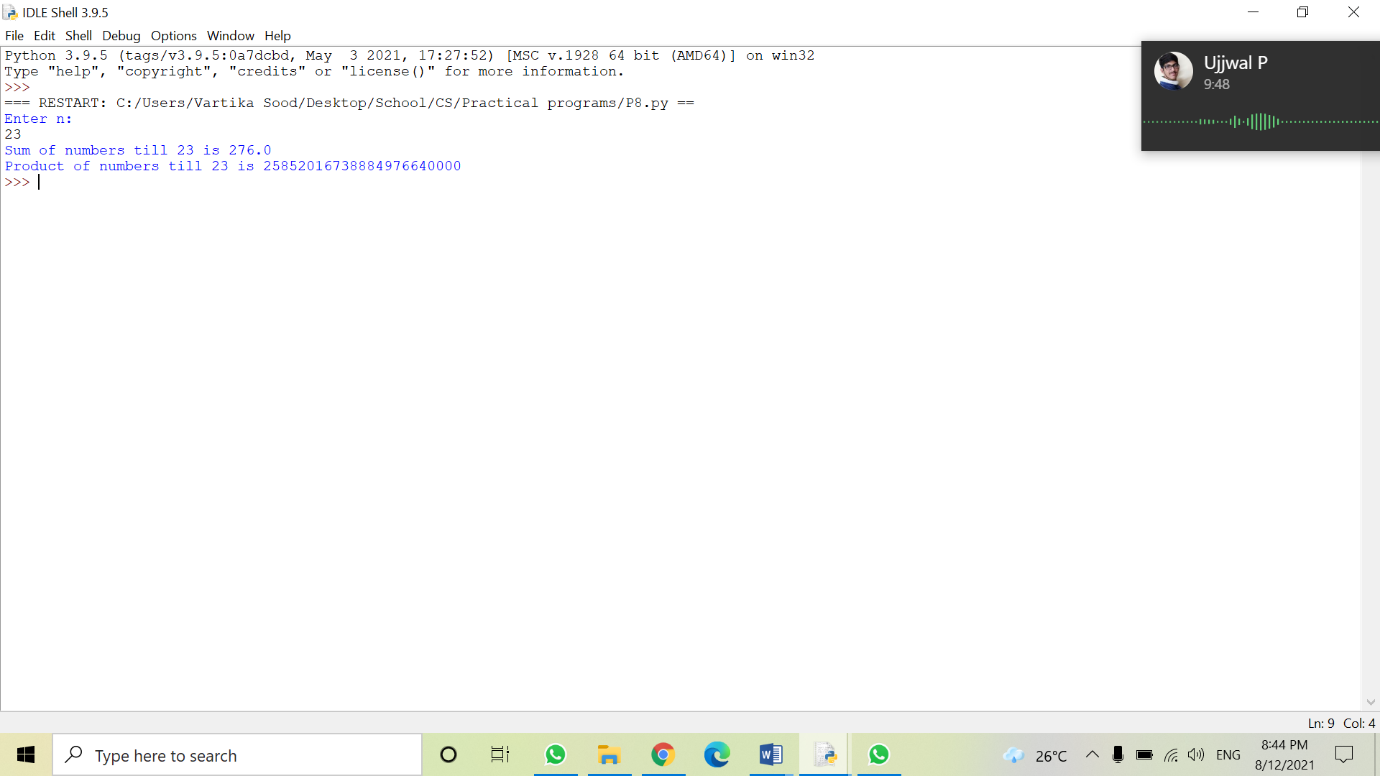
sum=n\*(n+1)/2

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

factorial\*=num

print("Sum of numbers till",n,"is",sum)

print("Product of numbers till",n,"is",factorial)

**Output:**

**Program:**

Write a Python Program to take an integer input N from the user. Print N Fibonacci numbers. Recall that Fibonacci series progresses as 0 1 1 2 3 5 8 13………….…

**Source code:**

nterms = int(input("The numbers of terms of fibonacci series are: "))

n1,n2=0,1

count=0

if nterms<=0:

print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < nterms:

print(n1)

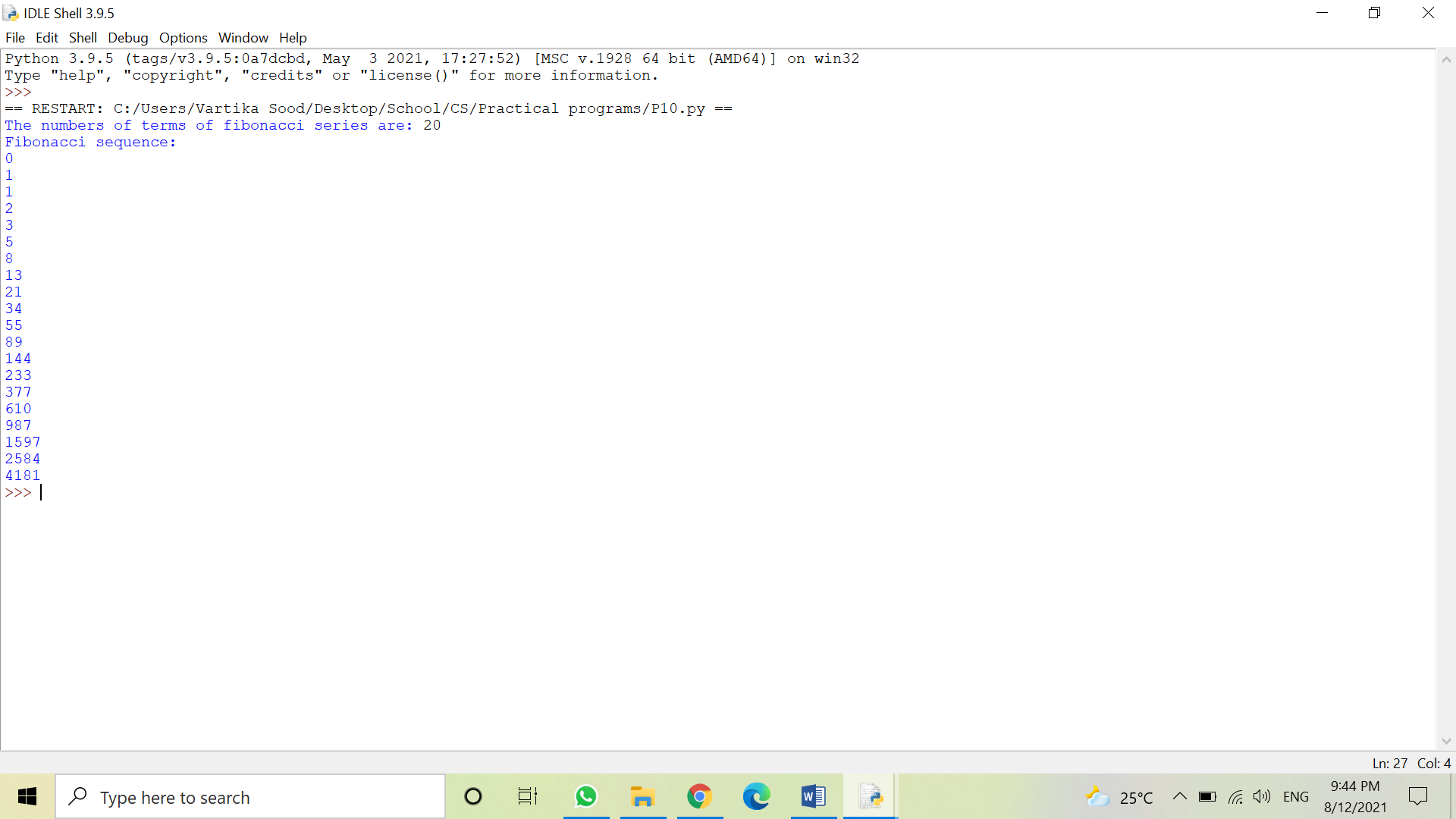
nth = n1 + n2

n1 = n2

n2 = nth

count += 1

**Output:**



**Program:**

Write a Python Program to take an integer input N from the user and print the multiplication table of N.

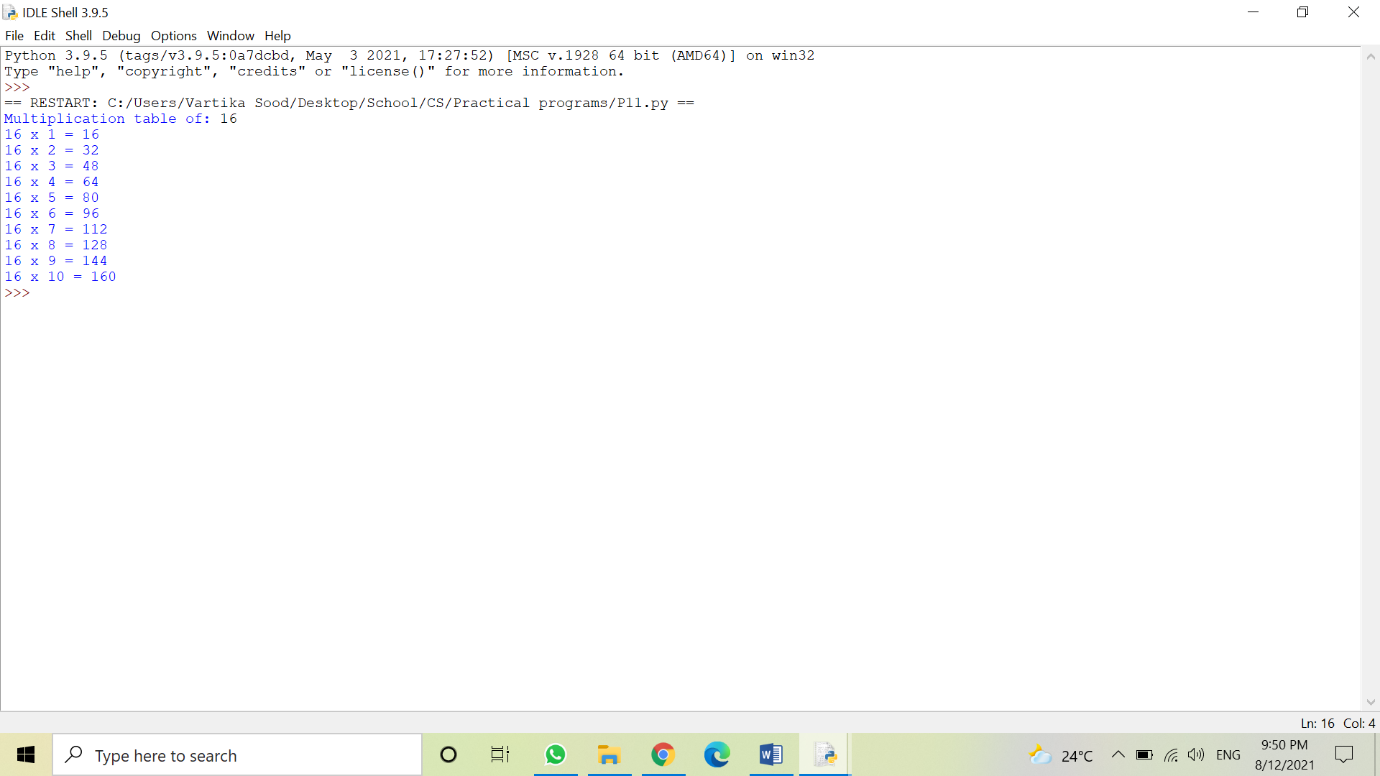
**Source code:**

num=int(input("Multiplication table of: "))

for i in range(1,11):

print(num,'x',i,'=',num\*i)

**Output:**



**Program:**

Write a Python Program that converts a decimal number into a binary number and vice-versa.

**Source code:**

print("Enter the Decimal Number: ")

dnum = int(input())

i = 0

bnum = []

while dnum!=0:

rem = dnum%2

bnum.insert(i, rem)

i = i+1

dnum = int(dnum/2)

i = i-1

print("The binary value of the number is:")

while i>=0:

print(end=str(bnum[i]))

i = i-1

print()

print("\n")

b\_num = list(input("Enter the binary number: \n"))

value = 0

for i in range(len(b\_num)):

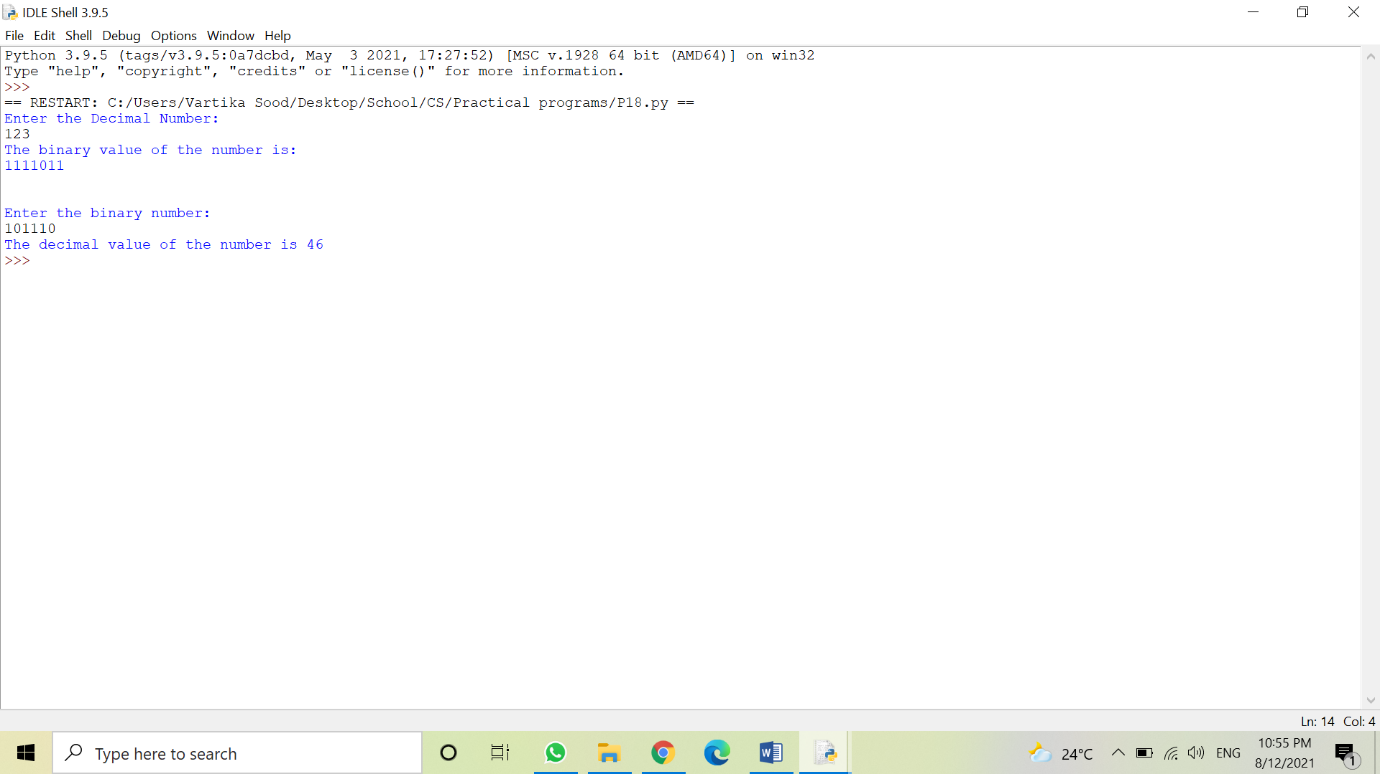
digit = b\_num.pop()

if digit == '1':

value = value + pow(2, i)

print("The decimal value of the number is", value)

**Output:**



**Program:**

Write a Python Program that defines 3 user-defined functions rect(), tri(), and circ() to calculate the area of a rectangle, a triangle, and a circle respectively. The main Program should be menu-driven and accept appropriate data from the user and call these functions according to the user’s choice.

**Source code:**

def calculate\_area(name):

if name == "rectangle":

l = int(input("Enter rectangle's length: "))

b = int(input("Enter rectangle's breadth: "))

rect\_area=l\*b

print("The area of rectangle is: ",{rect\_area})

elif name == "triangle":

h = int(input("Enter triangle's height length: "))

b = int(input("Enter triangle's breadth length: "))

tri\_area = 0.5 \* b \* h

print("The area of triangle is: ",{tri\_area})

elif name == "circle":

r = int(input("Enter circle's radius length: "))

pi = 3.14

circ\_area = pi \* r \* r

print("The area of triangle is: ",{circ\_area})

else:

print("Sorry! This shape is not available")

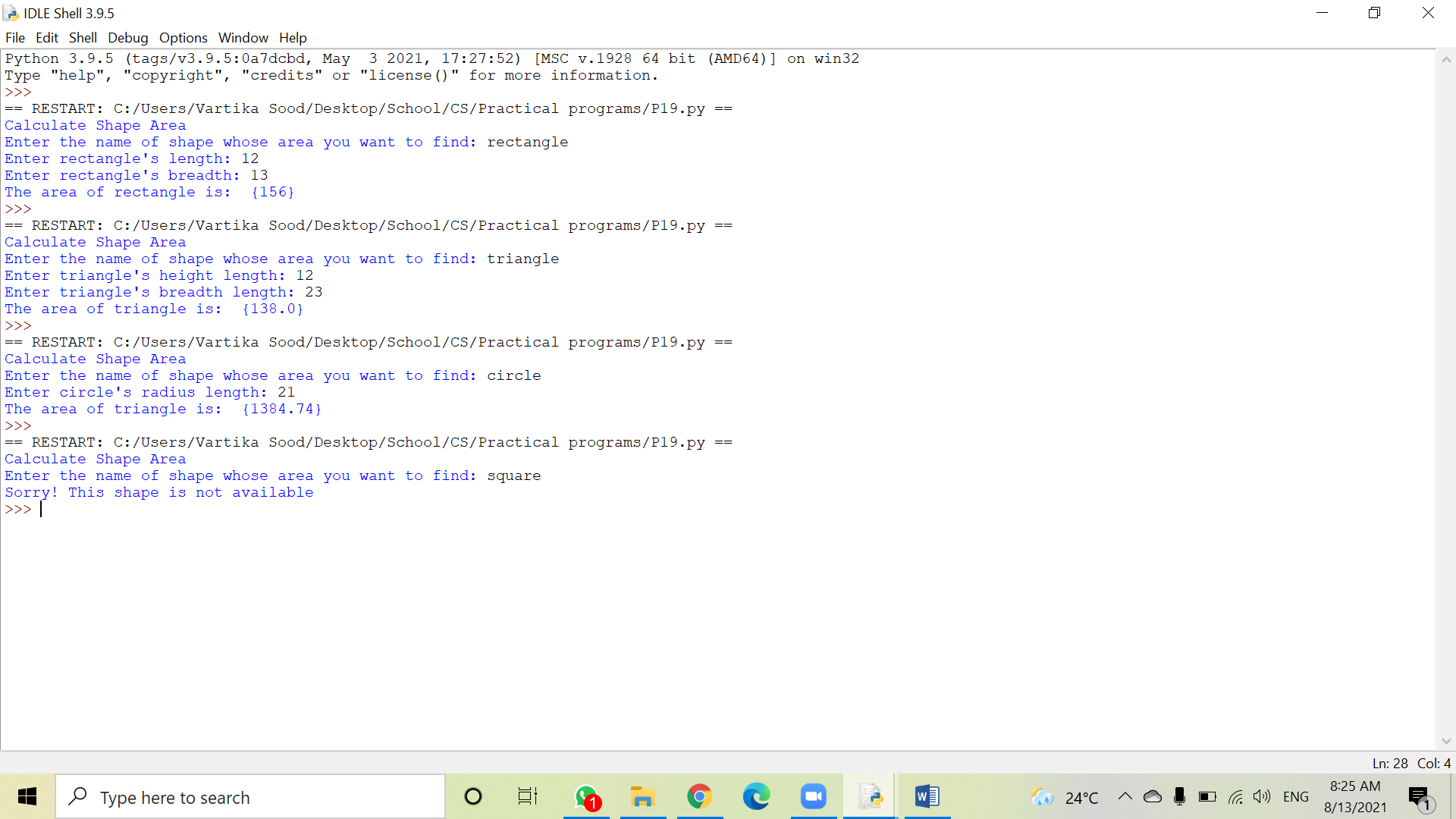
if \_\_name\_\_ == "\_\_main\_\_" :

print("Calculate Shape Area")

shape\_name = input("Enter the name of shape whose area you want to find: ")

calculate\_area(shape\_name)

**Output:**



**Program:**

Write a Python Program to print the following pyramid patterns:

& & & & & &

& & & & &

& & & &

& & &

& &

&

& &

& & &

& & & &

& & & & &

& & & & & &

**Source code:**

def pyramid(n):

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

print(" ",end='')

for num in range(j,0,-1):

print("& ",end="")

print('')

count=6

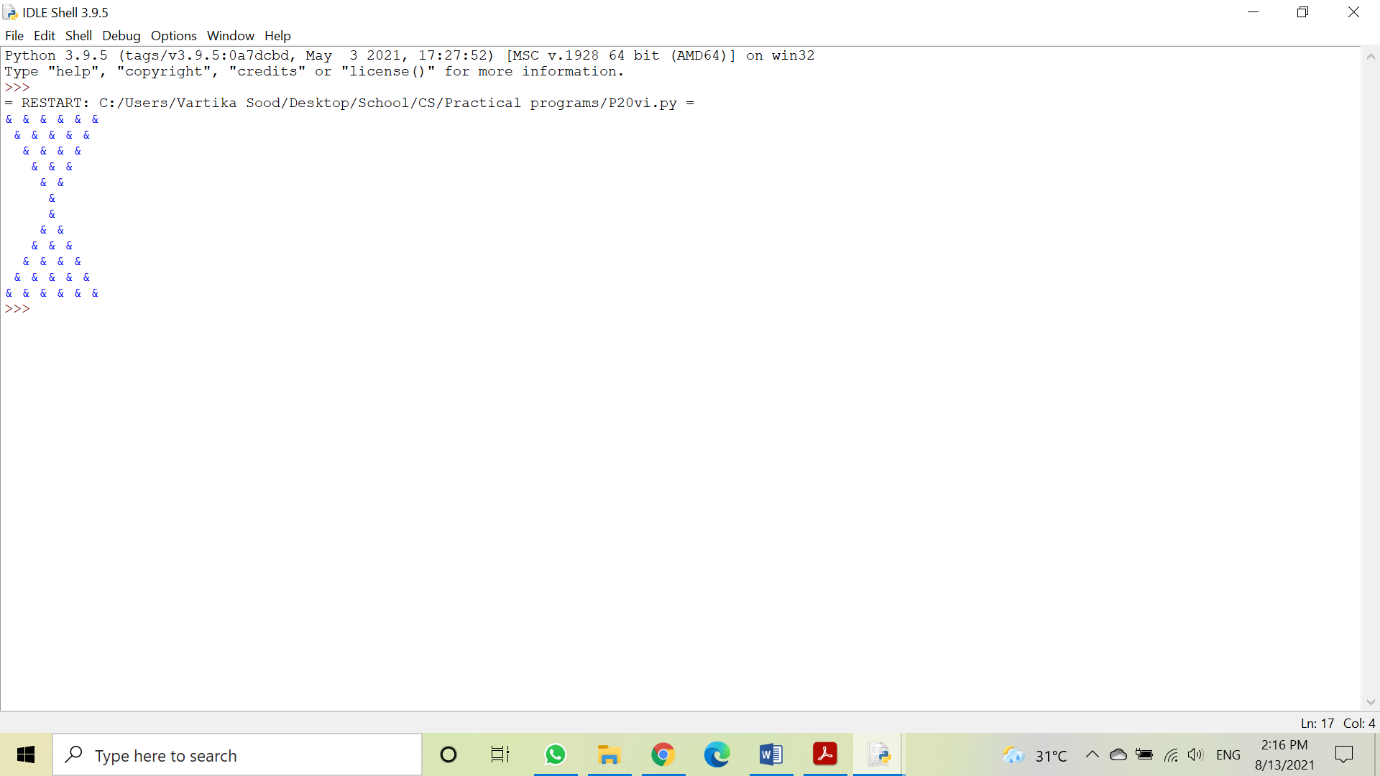
for j in range(count,0,-1):

pyramid(count)

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

pyramid(count)

**Output:**



**Program:**

Write a Python Program that uses string slicing to find the reverse of a string input from the user and also check if the string is a palindrome or not.

**Source code:**

print("String Palindrome Check")

stringA = input("Enter String for palindrome check: ")

stringB = str(stringA)[::-1]

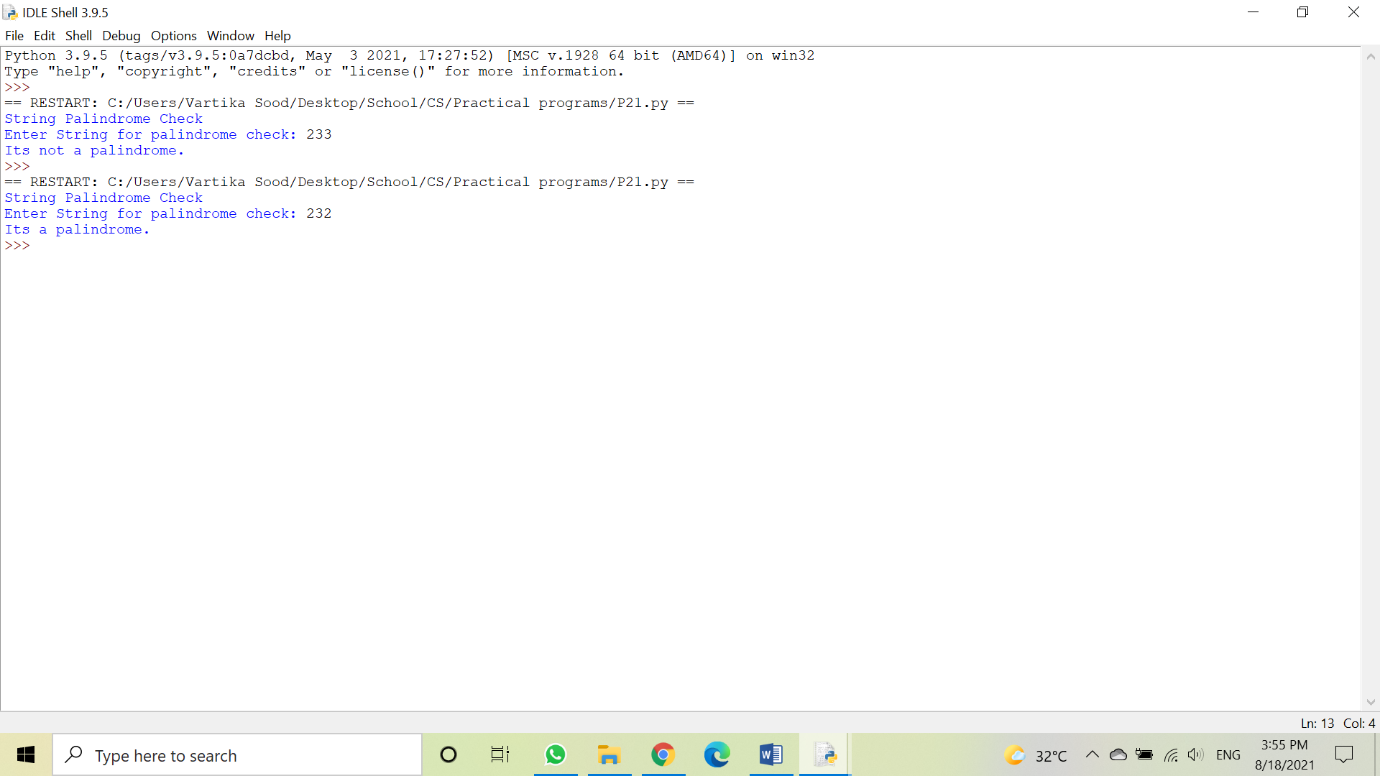
if stringA==stringB:

print("Its a palindrome.")

else:

print("Its not a palindrome.")

**Output:**



**Program:**

Write a Python Program to input a string from the user and calculate the number of digits, alphabets, upper-case letters, lower-case letters, and special characters in the string.

**Source code:**

print("String Anatomy (Character Types)")

stringA = input("Enter String Here: ")

lc , uc , dig , spec = 0 , 0 , 0 , 0

for j in range (0 , (len(stringA) - 1)):

chk = str(stringA[j])

if (chk.islower()):

lc+=1

elif (chk.isupper()):

uc+=1

elif (chk.isnumeric()):

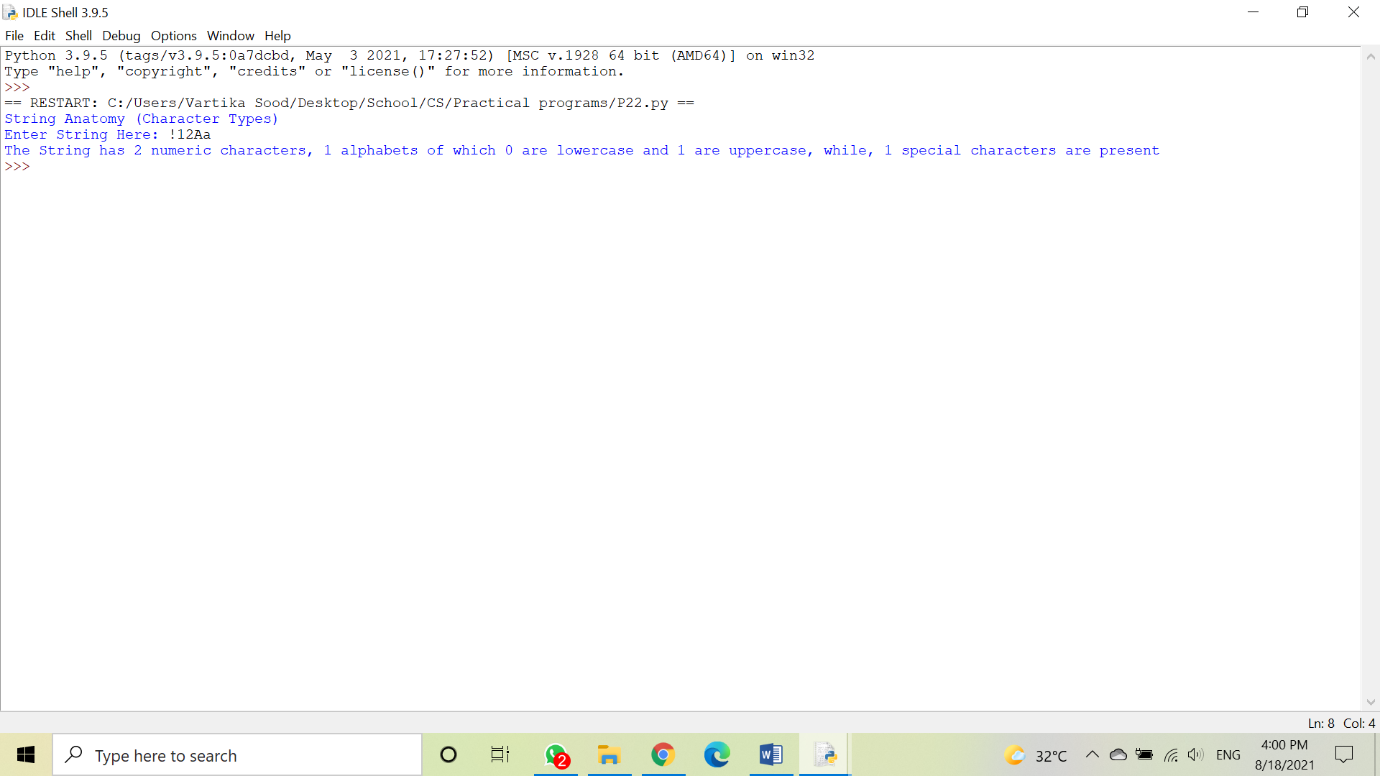
dig+=1

elif (chk.isascii()):

spec+=1

print("The String has",dig,"numeric characters,",(lc + uc),"alphabets of which",lc,"are lowercase and",uc,"are uppercase, while,",spec,"special characters are present")

**Output:**



**Program:**

Write a Python Program to input a string from the user and check for vowels and consonants. Remove all vowels, change the case of the consonants and generate the resultant string.

**Source code:**

Sen,ts=input("Enter a few Lines: \n"),[]

for j in range(0,len(Sen)):

ts.append(Sen[j])

for num in range(0,len(Sen)):

a=ts[num]

if(a=="a" or a=="A" or a=="e" or a=="E" or a=="i" or a=="I" or a=="o" or a=="O" or a=="u" or a=="U"):

ts[num]=""

else:

ts[num]=ts[num].upper()

print(ts[num],end="")

**Output:**

**Program:**

Write a Python Program to calculate the average length of a word in a string input by user.

**Source code:**

n = input("Enter Sentence: ").split()

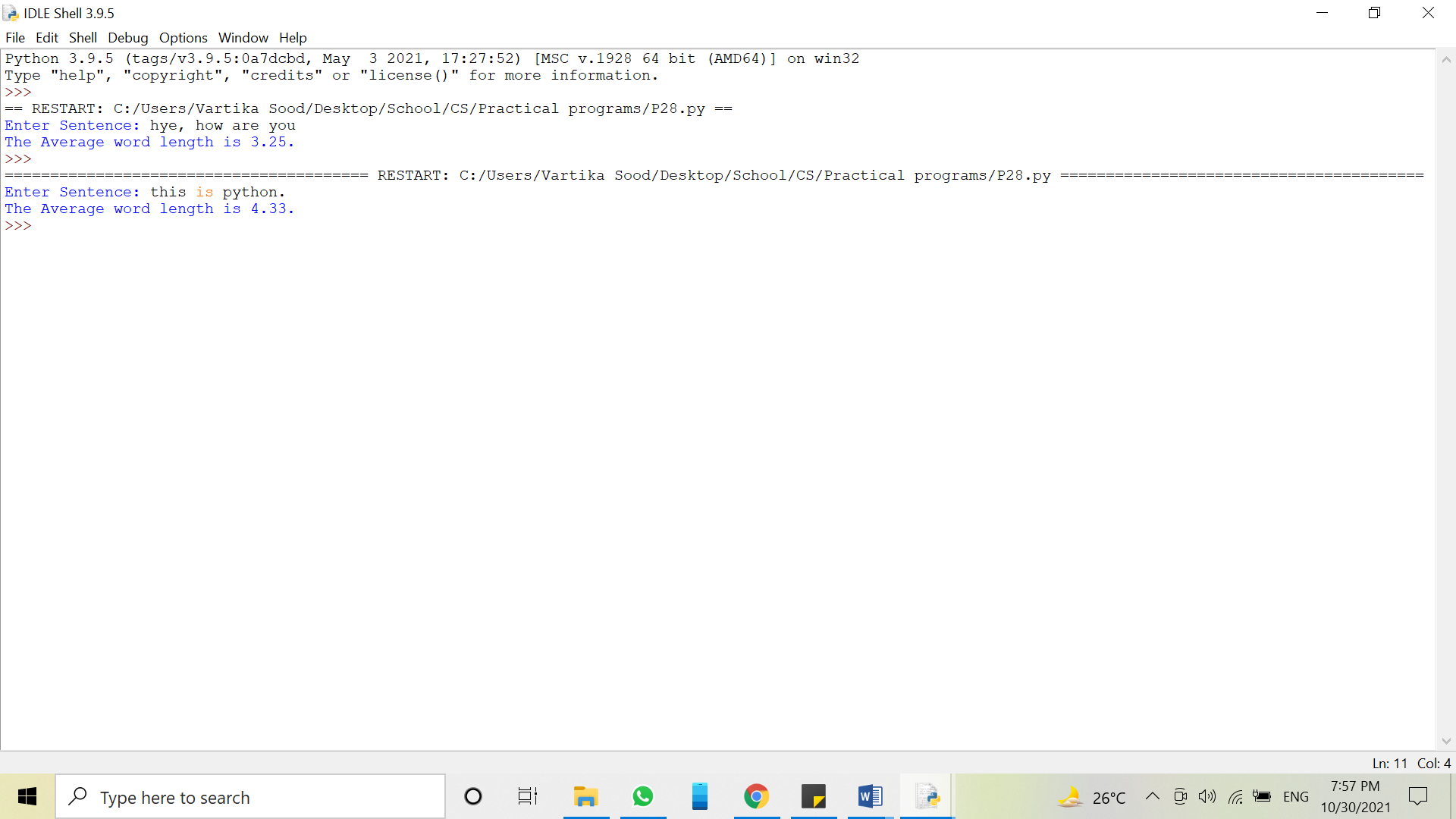
charCount = 0

for j in n:

charCount+=len(j)

print("The Average word length is ",round((charCount/len(n)),2),'.',sep='')

**Output:**



**Program:**

Write a menu driven Python Program to perform isdigit, len and isalpha function on input string.

**Source code:**

s = input("Enter a string: ")

while True:

c = input("What function do you want to use with the string? (Press Corresponding Number)" + "\n" + "(1)isdigit()" + "\n" + "(2)len()" + "\n" + "(3)isalpha()"+"\n")

if(c=='1'):

print("The result of isdigit() function on string '",s,"' is: ",s.isdigit(),sep="")

break

elif(c=='2'):

print("The result of len() function on string '",s,"' is: ",len(s),sep="")

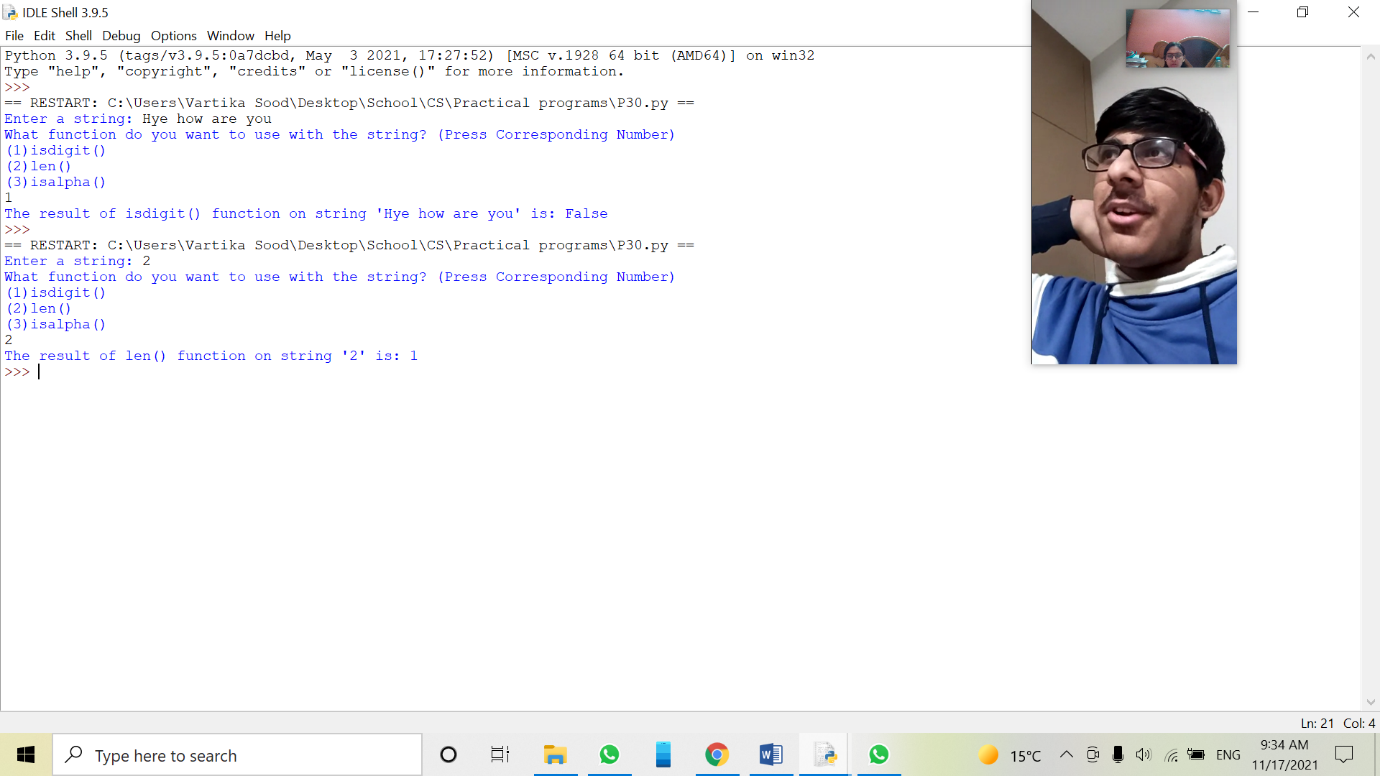
break

else(c=='3'):

print("The result of isalpha() function on string '",s,"' is: ",s.isalpha(),sep="")

break

**Output:**



**Program:**

Write a Program to find the length of the list and also find the index values of the elements.

**Source code:**

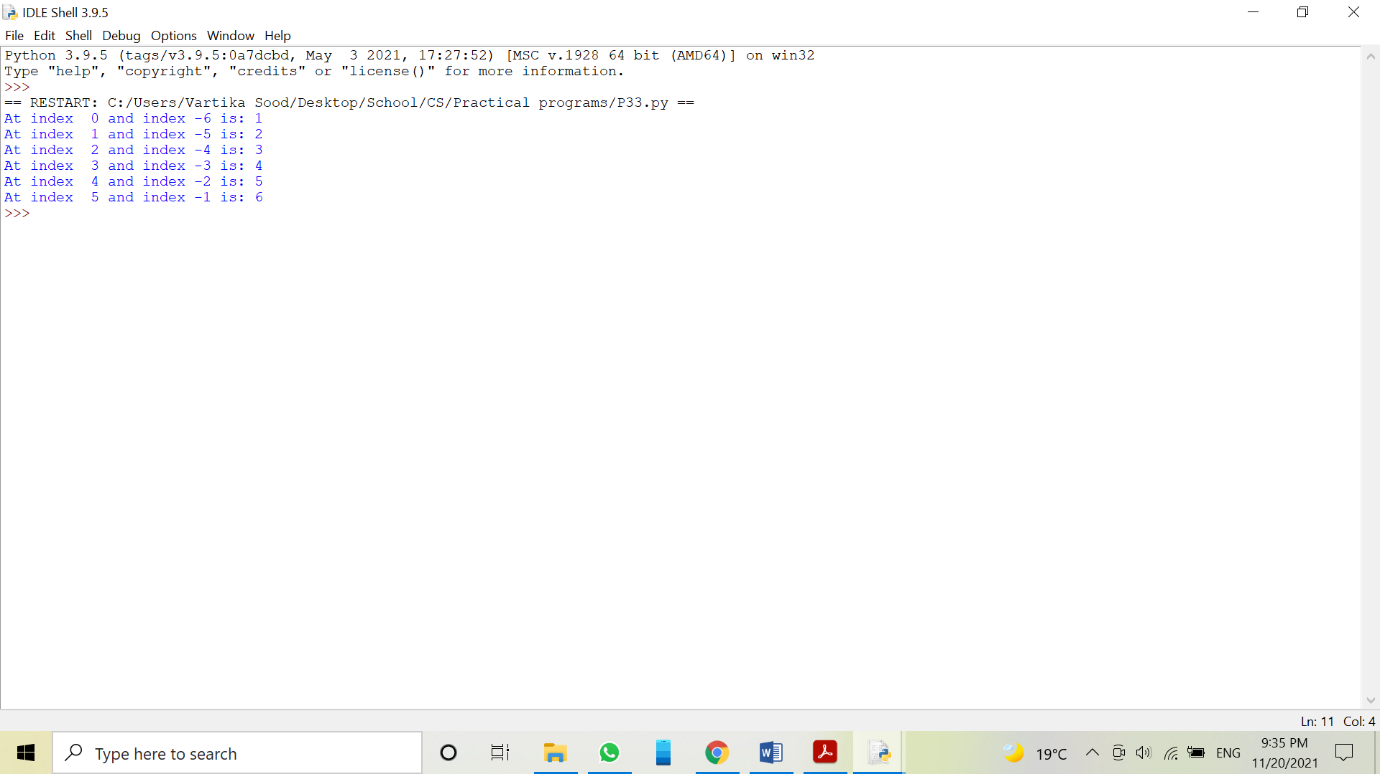
list=[1,2,3,4,5,6]

length=len(list)

for i in range(length):

print("At index ",i,"and index",i-length,"is:",list[i])

**Output:**



**Program:**

Write a Program to use some list functions and methods.

**Source code:**

list=[10,20,30,40,50,60,10]

print("list=",list)

print("\n")

print("index() method helps to find the index value of any element")

print("index value of 40 is",list.index(40))

print("\n")

print("append() helps to add items to the end of the list")

list.append(70)

print(list)

print("\n")

print("extend() method helps to add a list to another list")

list2=[80,90]

list.extend(list2)

print(list)

print("\n")

print("insert() method helps to add elements at desired position")

list.insert(2,25)

print(list)

print("\n")

print("remove() method helps to remove elemnts from a list without index value")

list.remove(90)

print(list)

print("\n")

print("count() method helps us to find the number of times an element occured in a list")

print(list.count(10))

print("\n")

print("reverse() method helps us to reverse the list")

list.reverse()

print(list)

print("\n")

print("sort() method helps us to arrange elements in a list in increasing or decreasing order.")

l1=[10,1,4,50,3,8,2,8,100]

print(l1)

l1.sort()

print(l1)

l1.sort(reverse=True)

print(l1)

print("\n")

print("clear() helps to remove all the elements from a list")

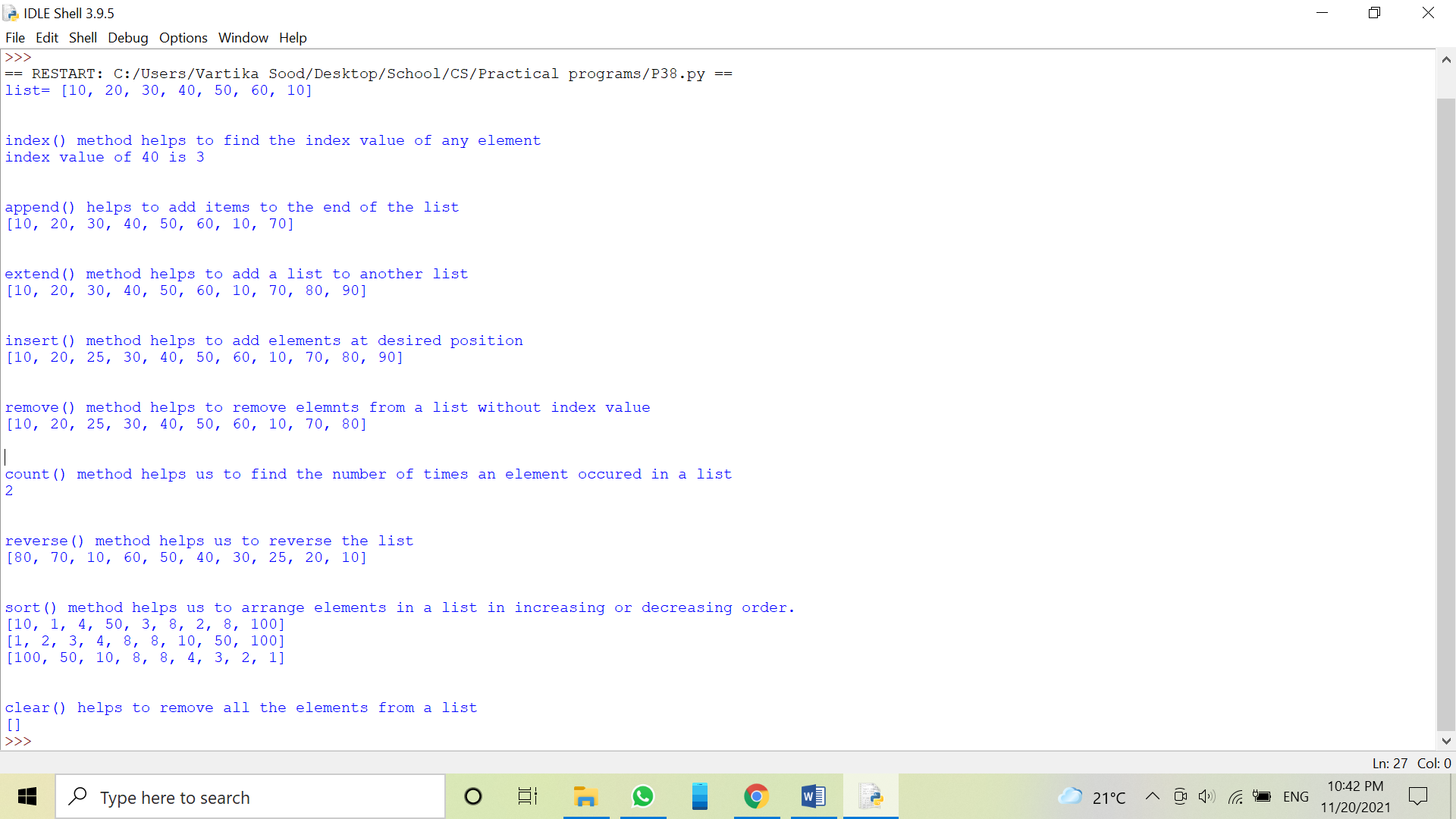
list.clear()

print(list)

**Output:**







**Program:**

Write a Program to use list comprehension.

**Source code:**

print("Program 1")

a=[i\*2 for i in range(1,6)]

print(a)

print("\n")

print("Program 2")

b=[(x,y) if x%2==0 else(x,y,1) for x in range(5) for y in range(5) if y%2==1]

print(b)

print("\n")

print("Program 3")

c=[num if num<5 else num\*2 for num in range(2,9)]

print(c)

print("\n")

print("Program 3")

d=[('a',11),('b',12),('c',13)]

e=[n\*3 for(x,n) in d if x=='b' or x=='c']

print(e)

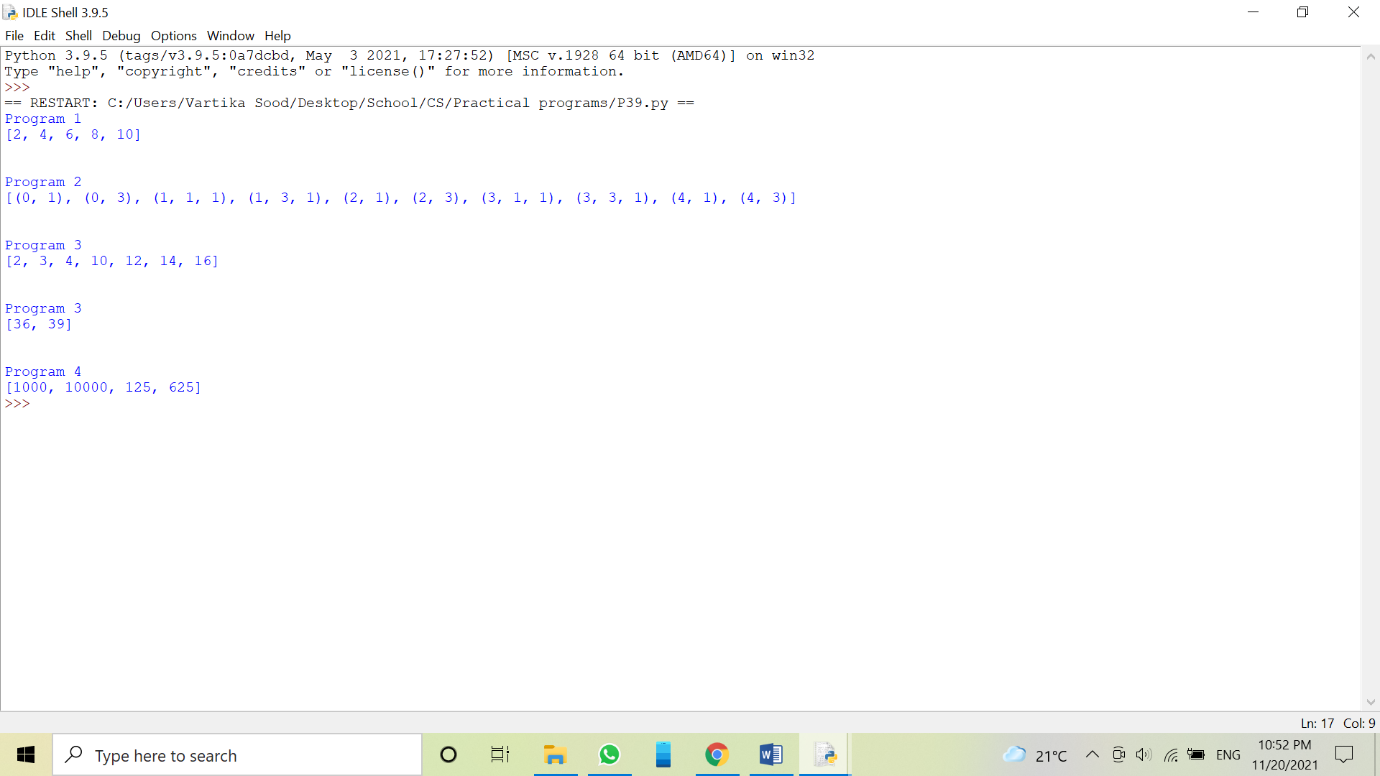
print("\n")

print("Program 4")

f=[x\*\*y for x in [10,5] for y in [3,4]]

print(f)

**Output:**



**Program:**

Write a Python Program to create a nested list to store details of students.

**Source code:**

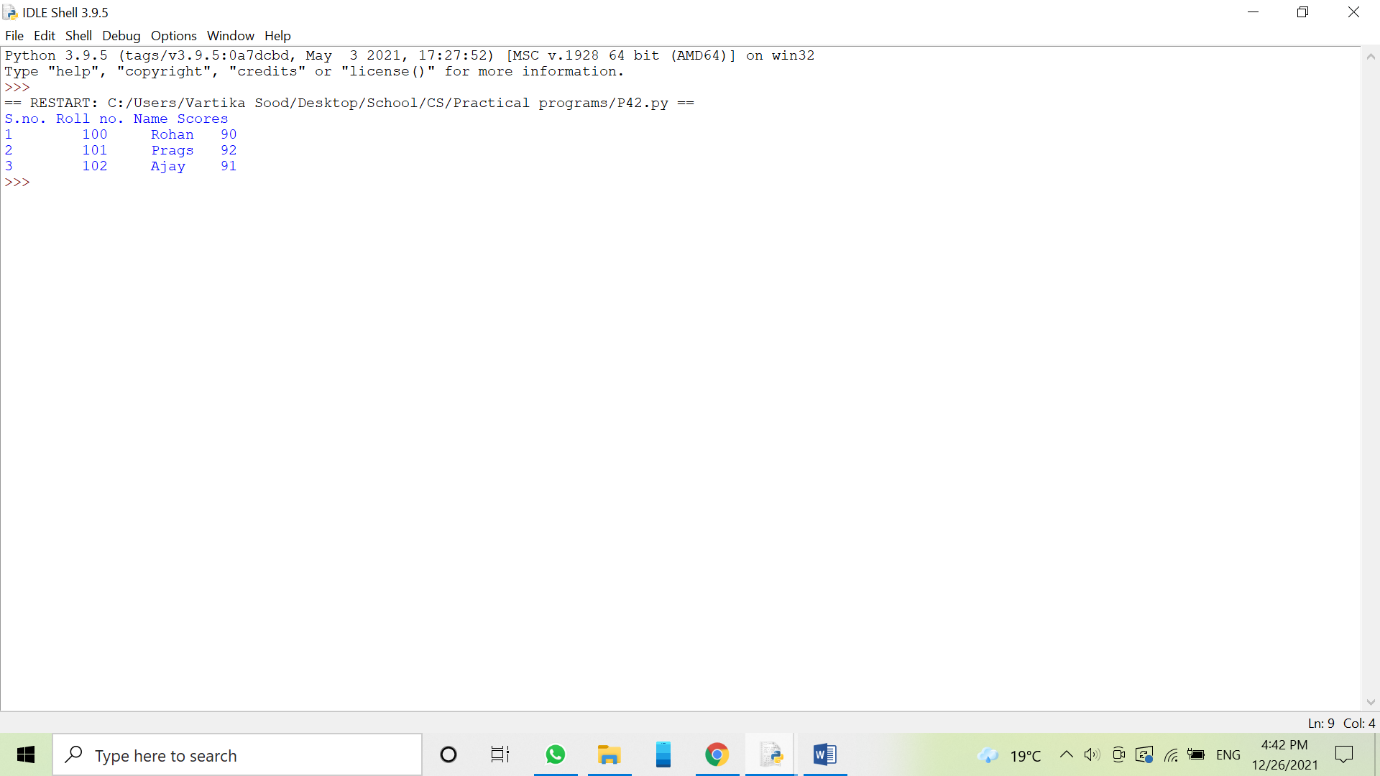
s=((100,"Rohan",90),(101,"Prags",92),(102,"Ajay",91))

print("S.no.","Roll no.","Name","Scores")

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

print((i+1),"\t",s[i][0],"\t",s[i][1],"\t",s[i][2])

**Output:**



**Program:**

Write a Python Program to input n numbers, store them and then print the maximum, minimum, sum and mean of these elements.

**Source code:**

nu=tuple()

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

for i in range(0,n):

num=int(input())

nu=nu+(num,)

print("The tupple is: ",nu)

print("\n")

print("The maximum numer is: ",max(nu))

print("\n")

print("The minimum number is:",min(nu))

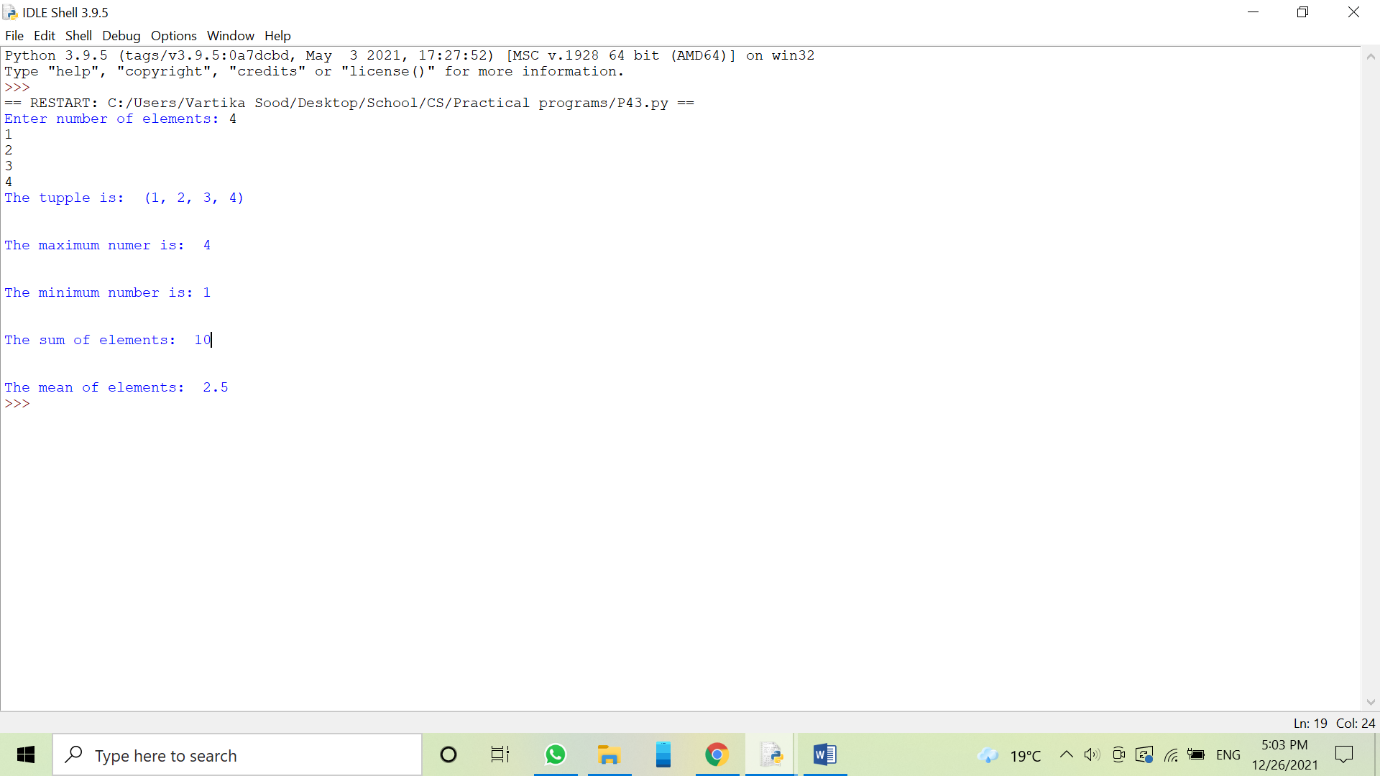
print("\n")

print("The sum of elements: ",sum(nu))

print("\n")

print("The mean of elements: ",sum(nu)/len(nu))

**Ouput:**



**Program:**

Write a Python Program to count frequency of elements in a tuple.

**Source code:**

print("Enter numbers separated by commas: ")

t=tuple([int(e) for e in input().split(',')])

i=0

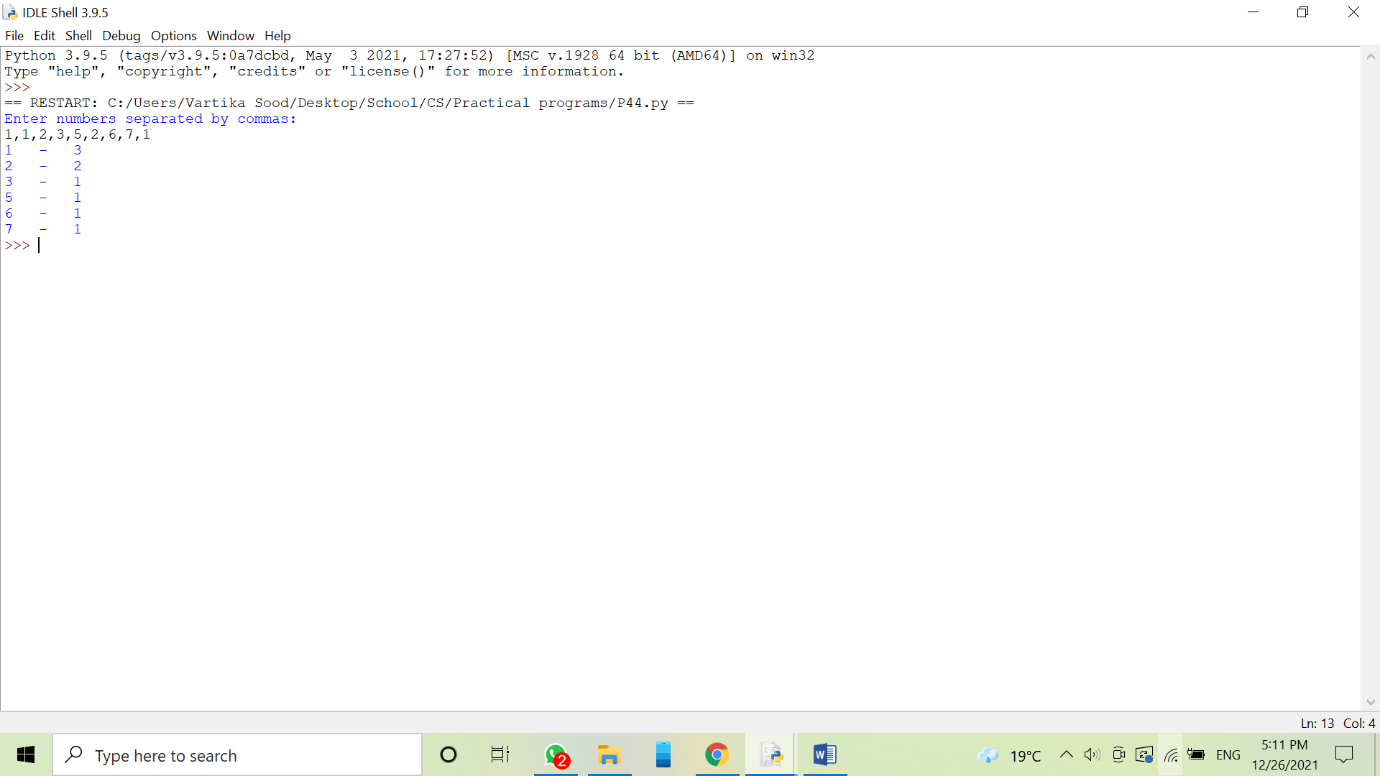
for e in t:

if i==t.index(e):

print(e," - ",t.count(e))

i+=1

**Output:**



**Program:**

Write a Python Program to input any 2 tuples and swap their values.

**Source code:**

t1=tuple()

n=int(input("Enter numbers of values in first tuple: "))

for i in range(n):

a=input("Enter element: ")

t1=t1+(a,)

t2=tuple()

m=int(input("Enter number of values in second tuple: "))

for i in range(m):

a=input("Enter element: ")

t2=t2+(a,)

print("Fisrt tuple: ",t1)

print("Second tuple: ",t2)

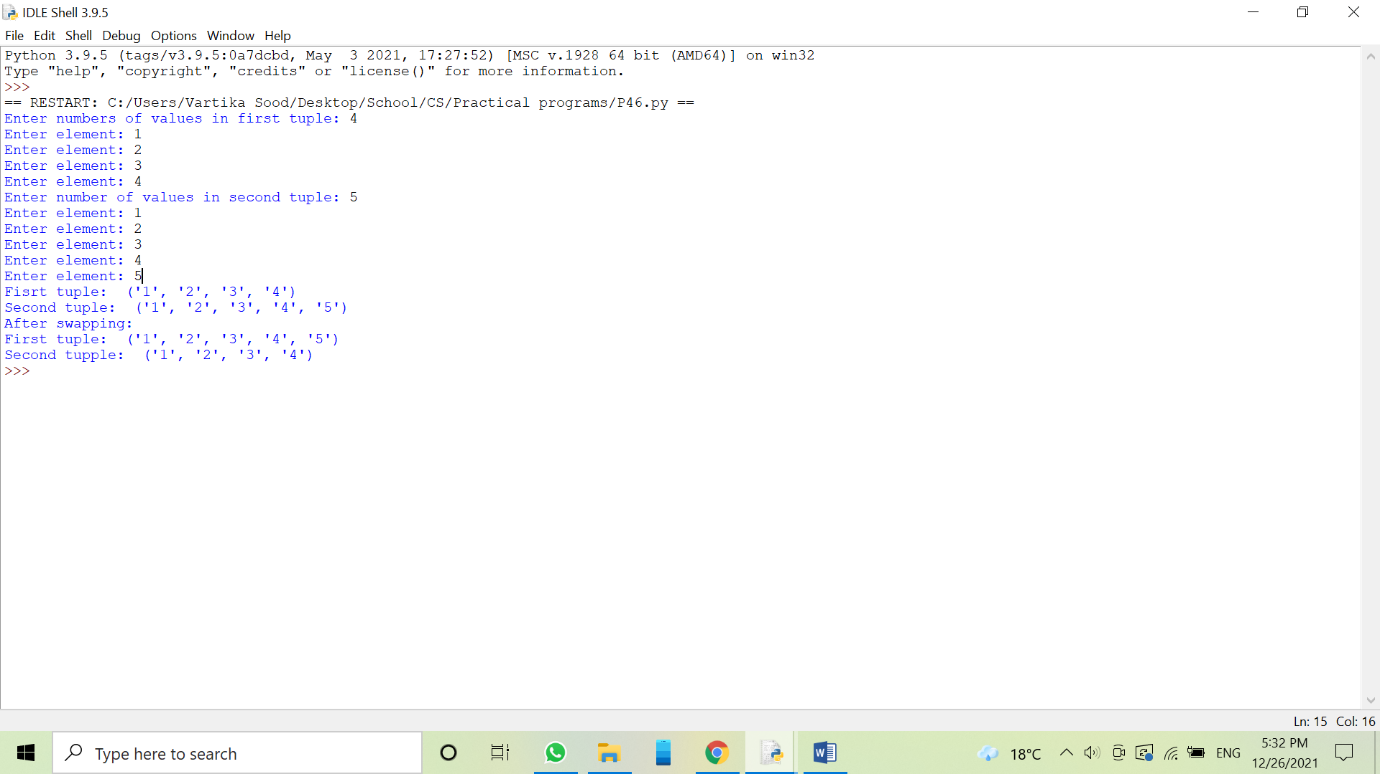
t1,t2=t2,t1

print("After swapping: ")

print("First tuple: ",t1)

print("Second tupple: ",t2)

**Output:**



**Program 47:**

Write a Python Program to input total number of sections and streams in class 11 and display on screen.

**Source code:**

class11=dict()

n=int(input("Enter the total number of sections in class 11: "))

i=1

while i<=n:

a=input("Enter section: ")

b=input("Enter stream: ")

class11[a]=b

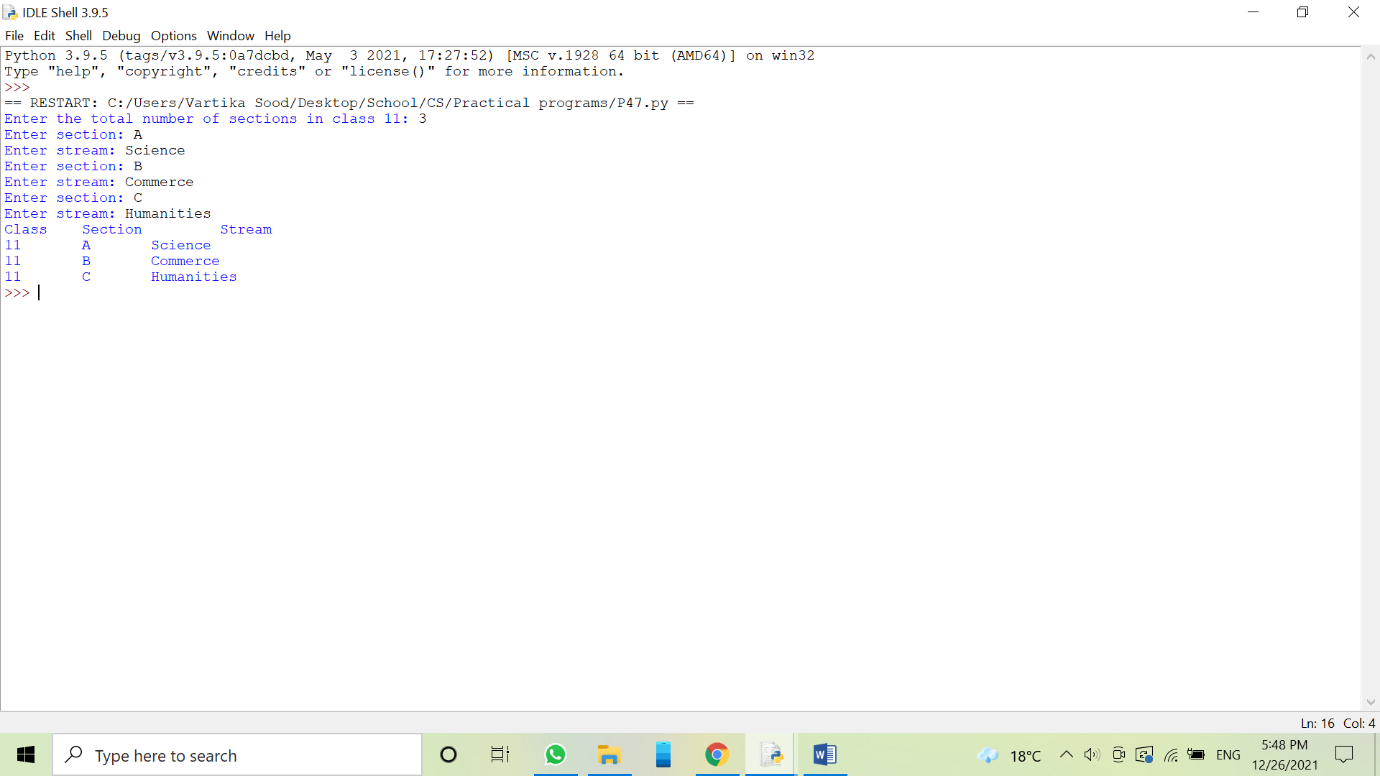
i=i+1

print("Class","\t","Section","\t","Stream")

for i in class11:

print("11","\t",i,"\t",class11[i])

**Output:**



**Program:**

Write a Python Program to count the frequency of elements using a dictionary.

**Source code:**

s=input("Enter a string: ")

d={}

for ch in s:

if ch in d:

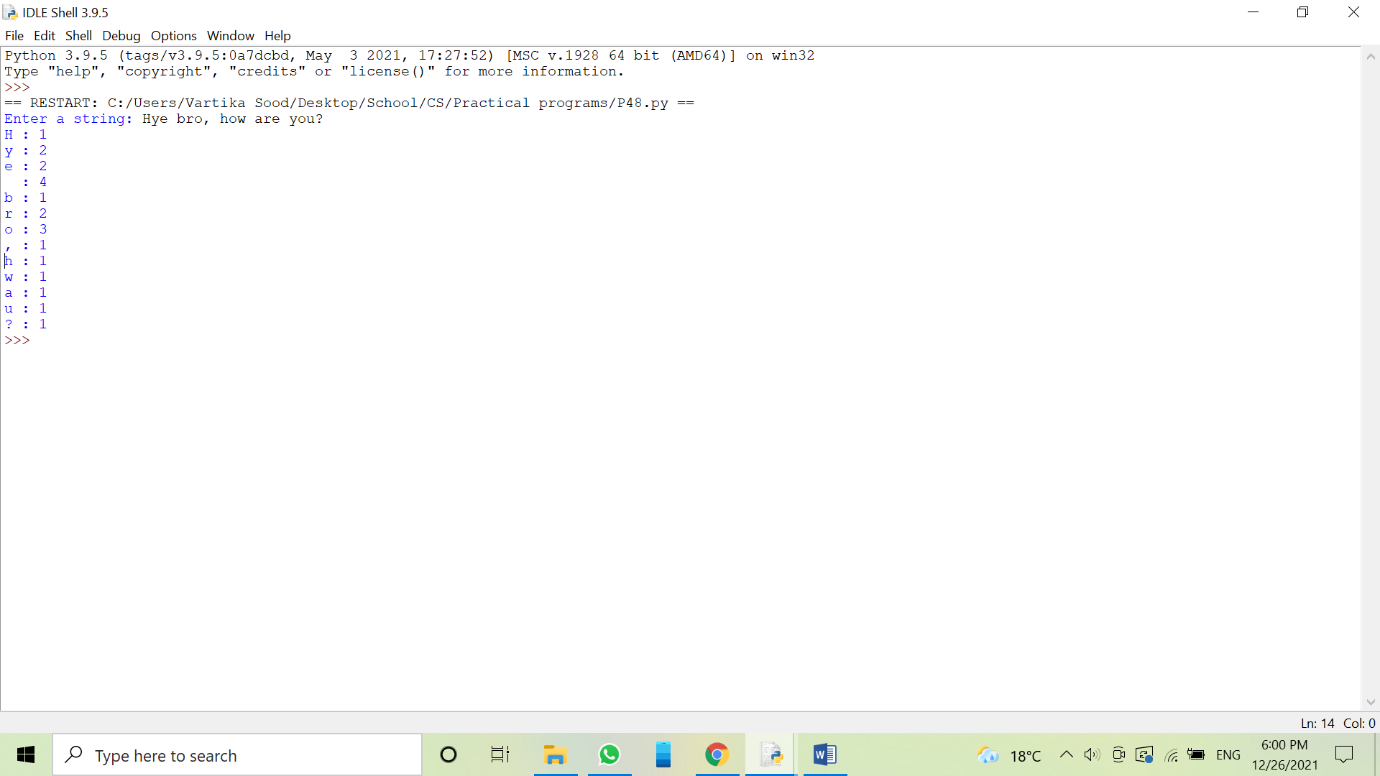
d[ch]+=1

else:

d[ch]=1

for k in d:

print(k,':',d[k])

**Output:**

**Program:**

Write a Python Program to iterate over dictionaries using for loops.

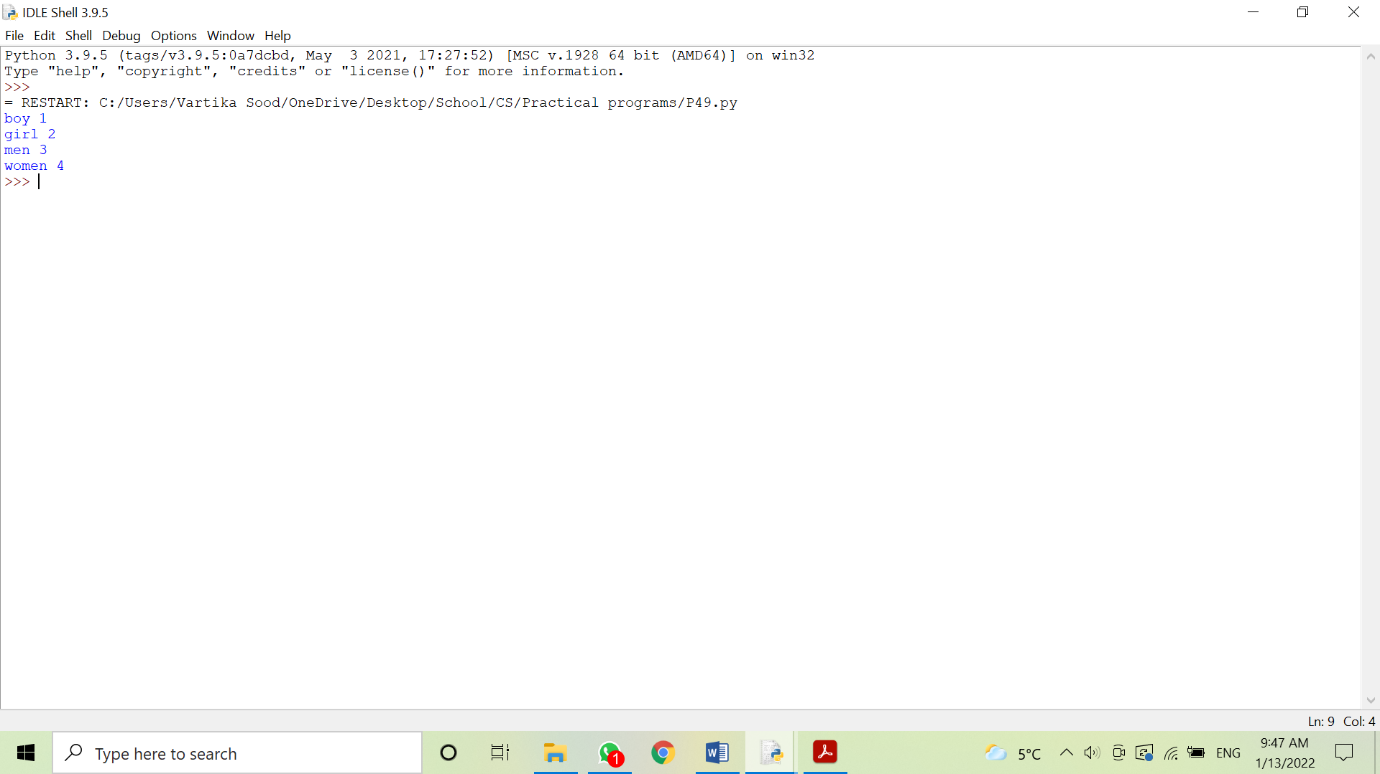
**Source code:**

d={'boy':1,'girl':2,'men':3,'women':4}

for key,value in d.items():

print(key,value)

**Output:**



**Program:**

Write a Python Program to get minimum and maximum value in dictionary.

**Source code:**

d={'a':5,'b':3,'c':1,'d':100}

print(d,"original dictionary")

d1=sorted(d)

min=d1[0]

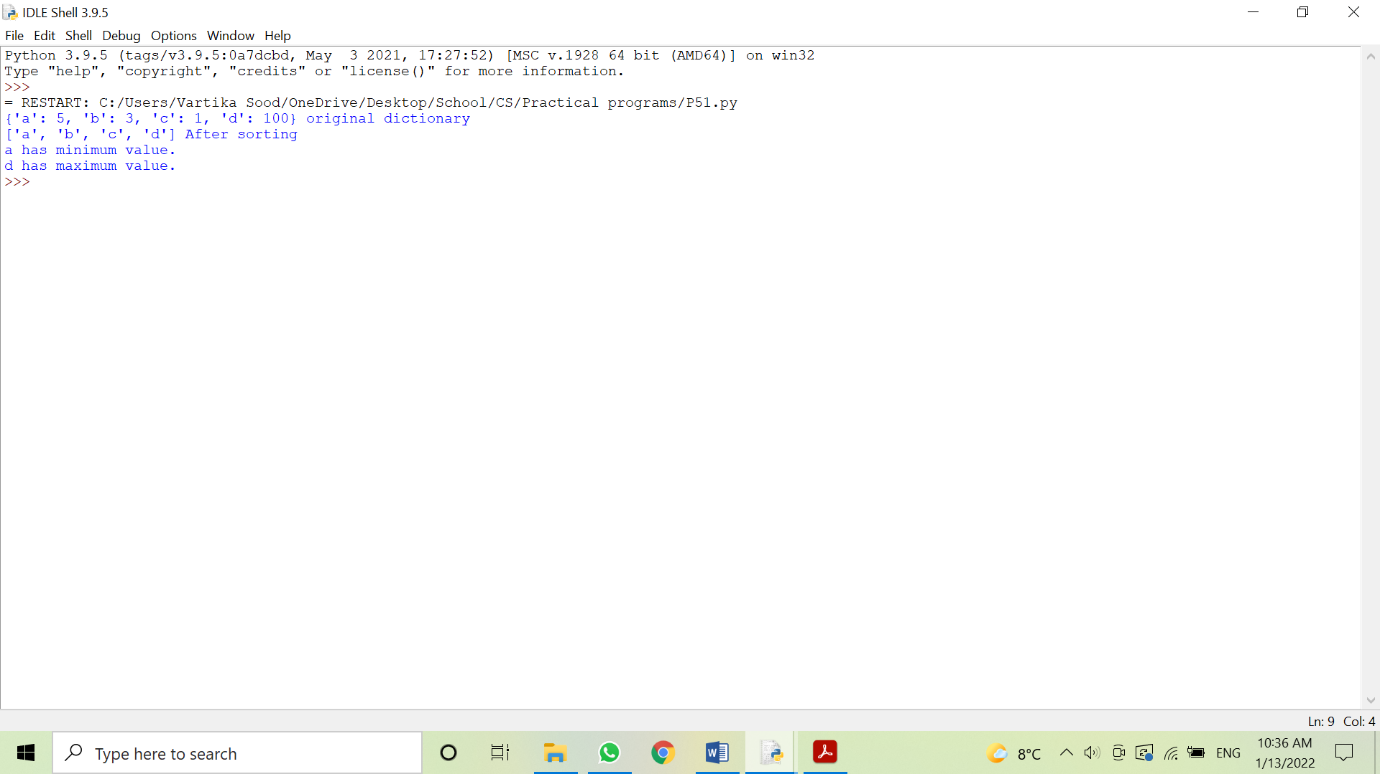
max=d1[-1]

print(d1,"After sorting")

print(min,'has minimum value.')

print(max,'has maximum value.')

**Output:**



**Program:**

Write a Python Program to concatenate 3 Python dictionaries.

**Source code:**

dict={}

d1={1:10,2:20}

d2={3:30,4:40}

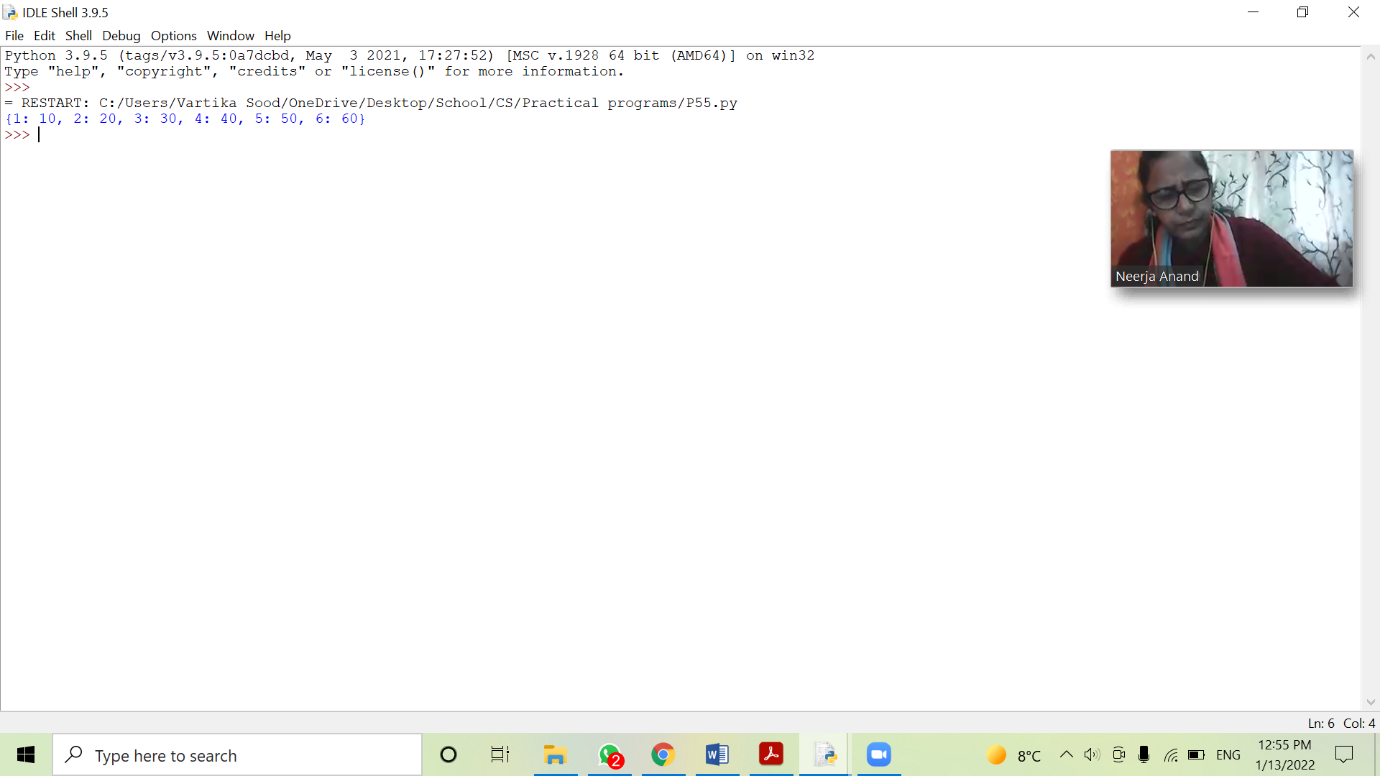
d3={5:50,6:60}

for d in (d1,d2,d3):

dict.update(d)

print(dict)

**Output:**



**Program:**

Write a Python Program to check if a given key exists in a dictionary or not.

**Source code:**

d={'a':1,'b':2,'c':3,'d':4}

print(d)

key=input('Enter a key to check: ')

if key in d.keys():

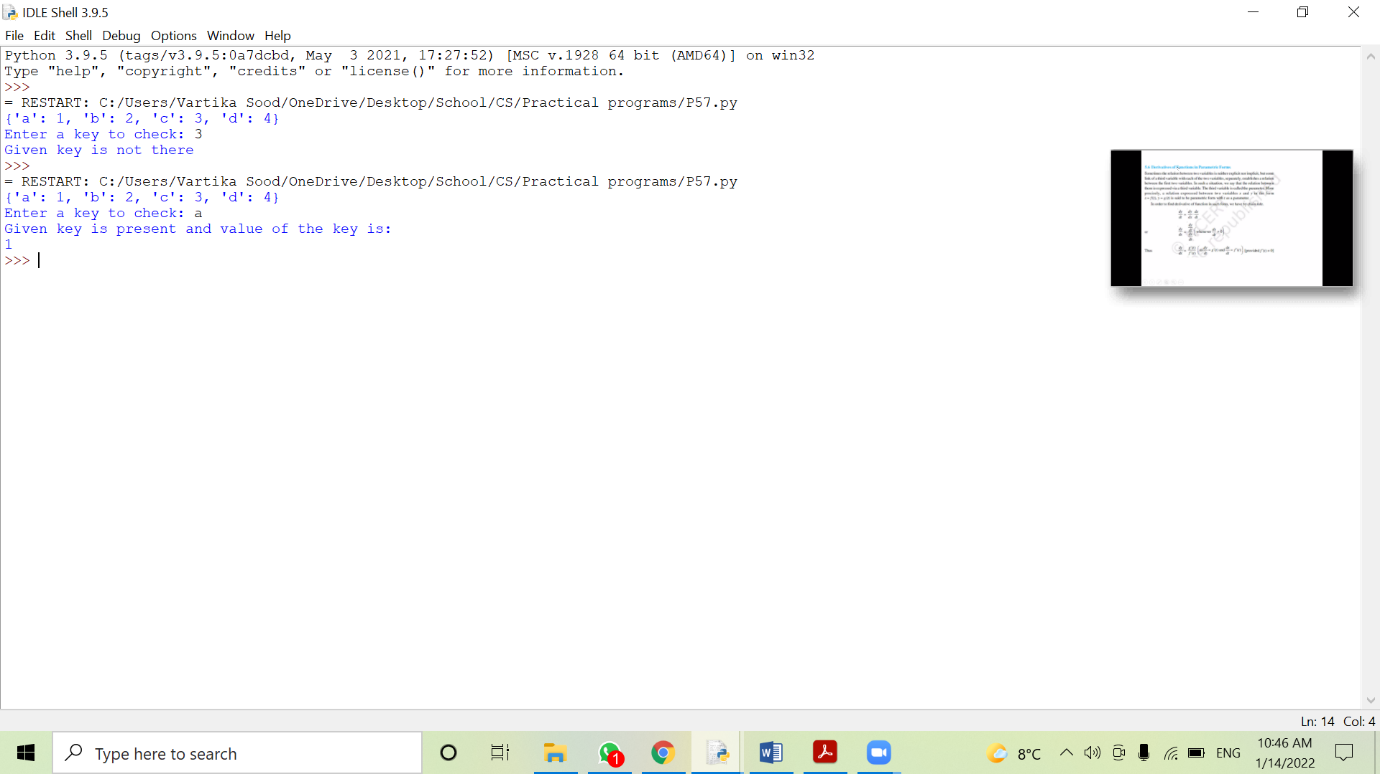
print("Given key is present and value of the key is: ")

print(d[key])

else:

print("Given key is not there")

**Output:**



**Program:**

Write a Python Program to create a dictionary with key as first character and value as words starting with that character.

**Source code:**

d={}

while True:

a=input("Enter: ")

l=a.split()

for word in l:

if (word[0] not in d.keys()):

d[word[0]]=[]

d[word[0]].append(word)

else:

if(word not in d[word[0]]):

d[word[0]].append(word)

for k,v in d.items():

print(k,":",v)

print(d)

**Output:**



**Program:**

Write a Python Program using dictionary to print the name and salary of employee.

**Source code:**

names=[]

dept=[]

salary=[]

details={'Employee Name:':names,'Department:':dept,'Salary:':salary}

rec=int(input('How many records are to be entered: '))

for i in range(rec):

n=input("Enter employee name: ")

names.append(n)

d=input('Enter employee department: ')

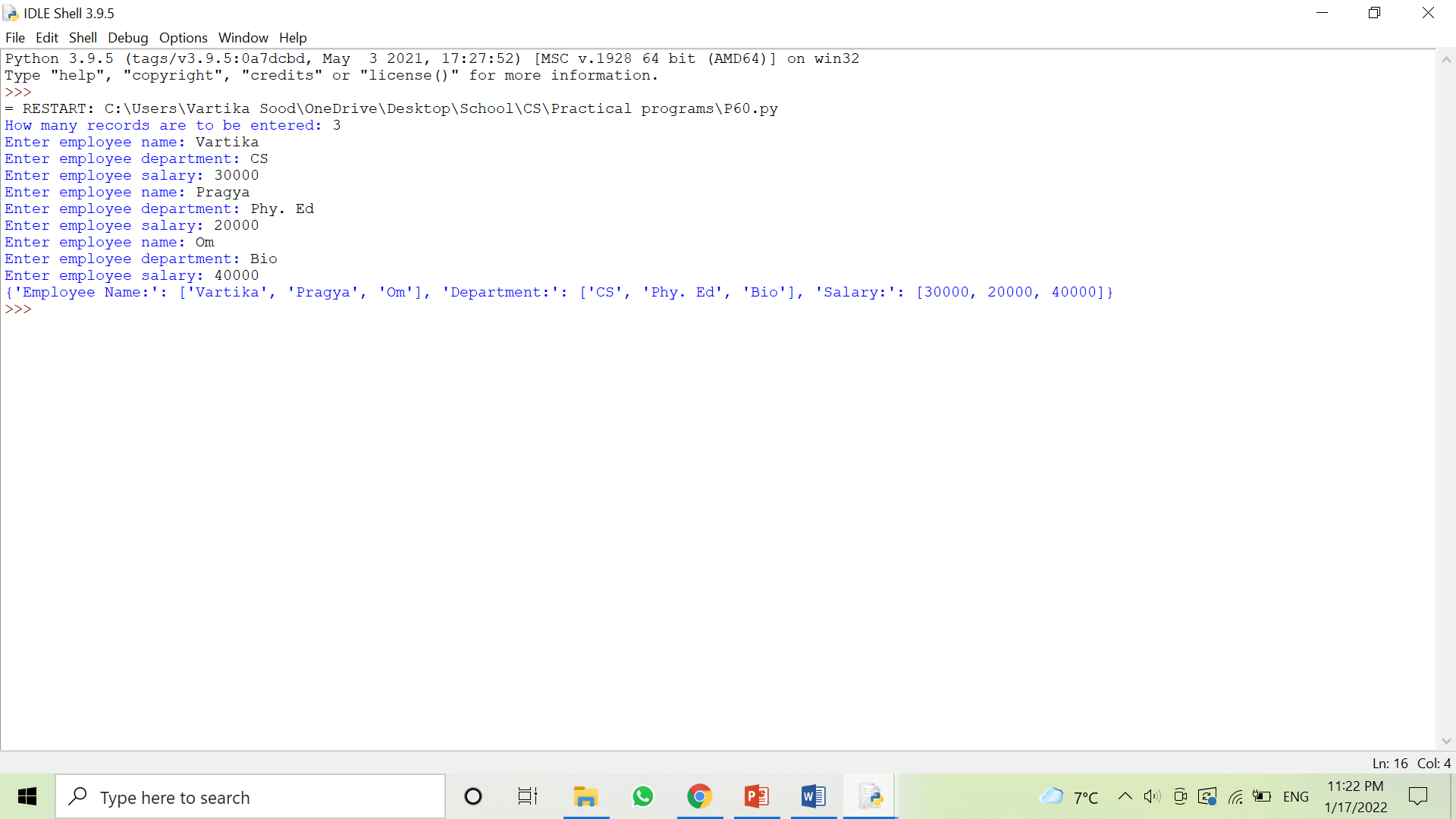
dept.append(d)

s=int(input('Enter employee salary: '))

salary.append(s)

print(details)

**Output:**



**Program:**

Write a Python Program to show the use of math module.

**Source code:**

#Math module

from math import \*

print('The square root of 46: ',sqrt(46))

print('The power function(4^3): ',pow(4,3) )

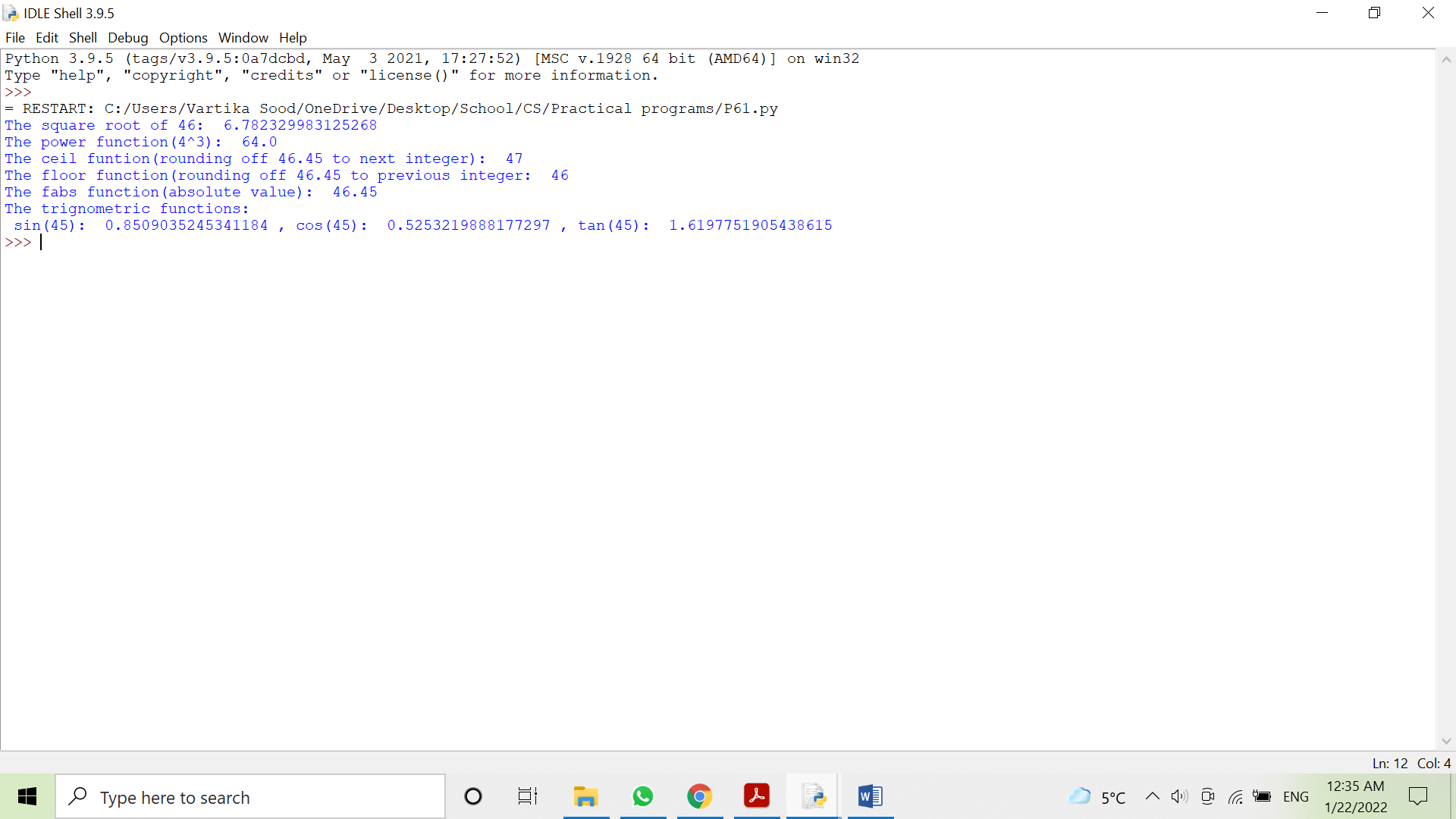
print('The ceil funtion(rounding off 46.45 to next integer): ',ceil(46.45))

print('The floor function(rounding off 46.45 to previous integer: ',floor(46.45))

print('The fabs function(absolute value): ',fabs(46.45))

print('The trignometric functions: ','\n','sin(45): ',sin(45),',','cos(45): ',cos(45),',','tan(45): ',tan(45))

**Output:**



**Program:**

Write a Python Program to make a guessing game using random module.

**Source code:**

import random

count=3

ans='y'

win=False

print("Guess what number computer generated between the range")

print("Total 3 chances are there ")

print("----------------------------------------------------------------")

a=int(input('Enter the lower range: '))

b=int(input('Enter the upper range: '))

while ans=='y':

num1=random.randint(a,b)

print("Chance Remaining :", count)

guess =int(input("Enter your answer :"))

if num1==guess:

print ("Congratulation! you guessed it right")

win=True

else:

print ("Wrong!")

count-=1

if count==0:

print ("Oops! You lost all your chances ")

print("Number was :",num1)

if win==True or count==0:

ans =input("Play Again?")

if ans=='y':

count=3

win=False

**Output:**



**Program:**

Write a Python Program to make a calculation game.

**Source code:**

import random

operators=['+','\*','-']

error=0

score=0

print("WELCOME TO MATH CALCULATION GAME ")

print("Rule: +4 for correct answer, -2 for wrong answer")

for i in range(5):

print("\*"\*50)

n1=random.randrange(1,100)

n2=random.randrange(1,100)

i =random.randrange(0,3)

op=operators[i]

result = 0

if op=='+':

result=n1+n2

elif op=='-':

if n1<n2:

n1, n2-n2, n1

result=n1-n2

elif op=='\*':

result=n1\*n2

print (n1,op,n2,'=')

ask=int(input())

if ask==result:

score+=4

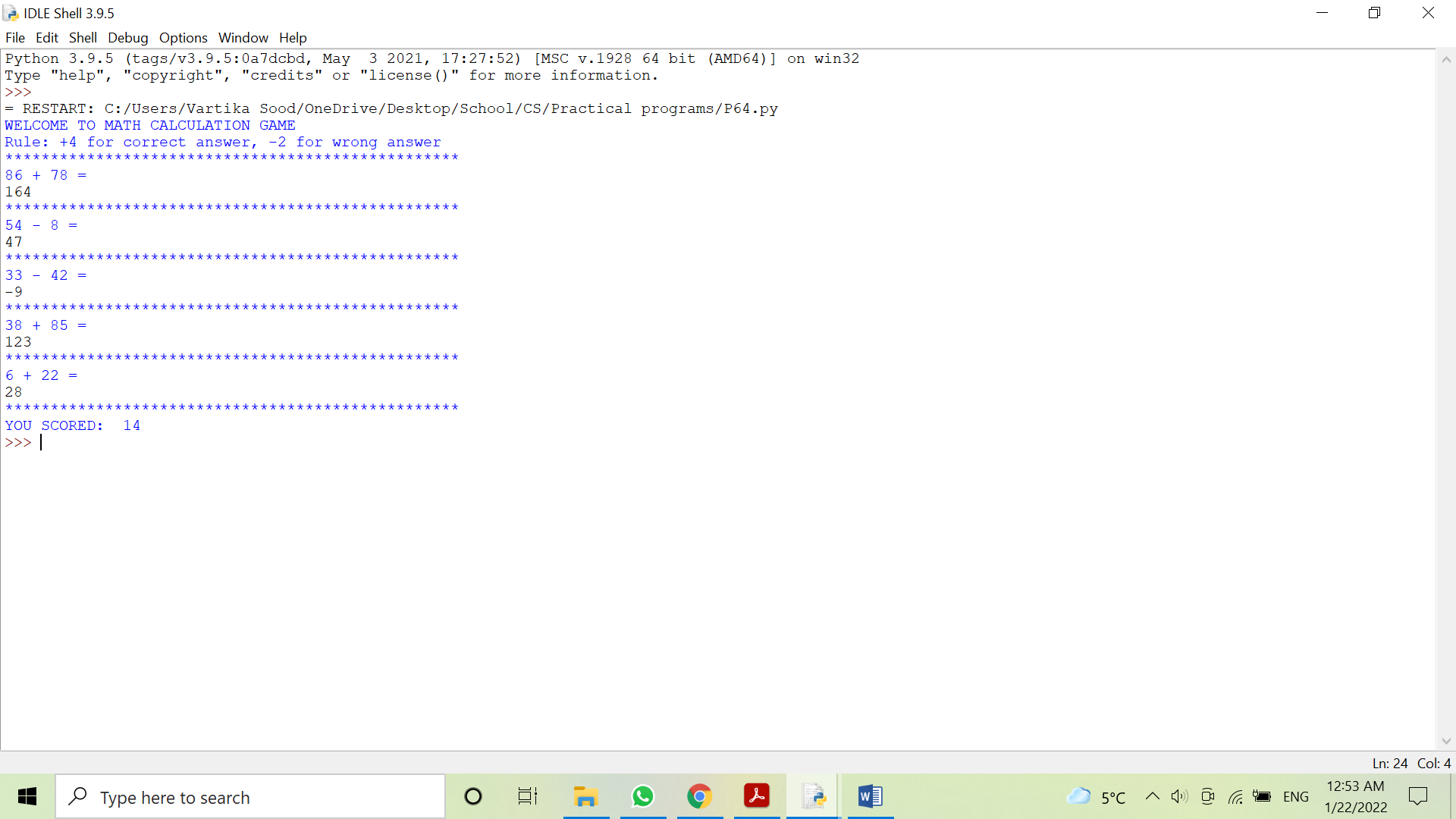
else:

score-=2

print('\*'\*50)

print("YOU SCORED: ",score,)

**Output:**



**Program:**

Write a Python Program to show the use of statistics module.

**Source code:**

from statistics import \*

l=[1,2,3,4,5,6,7,8,9]

print(l)

print('The mean of the numbers is : ',mean(l))

print('The mode of the numbers is : ',mode(l))

print('The median of the numbers is : ',median(l))

**Output:**



**FUNCTIONS**

**Program:**

Write a Python program to make a function to find the average of numbers.

**Source code:**

# 23/3/22

#Method 1:

def avg(n1,n2,n3,n4):

    av=(n1+n2+n3+n4)/4

    print(av)

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

b=int(input("Enter a number: "))

c=int(input("Enter a number: "))

d=int(input("Enter a number: "))

avg(a,b,c,d)

#Method 2:

def avg2(n1,n2,n3,n4):

    s=0

    l=4

    for k in n1,n2,n3,n4:

        if k=="":

            k=0

            l-=1

        s+=float(k)

    av=(s)/l

    print(av)

a=input("Enter a number: ")

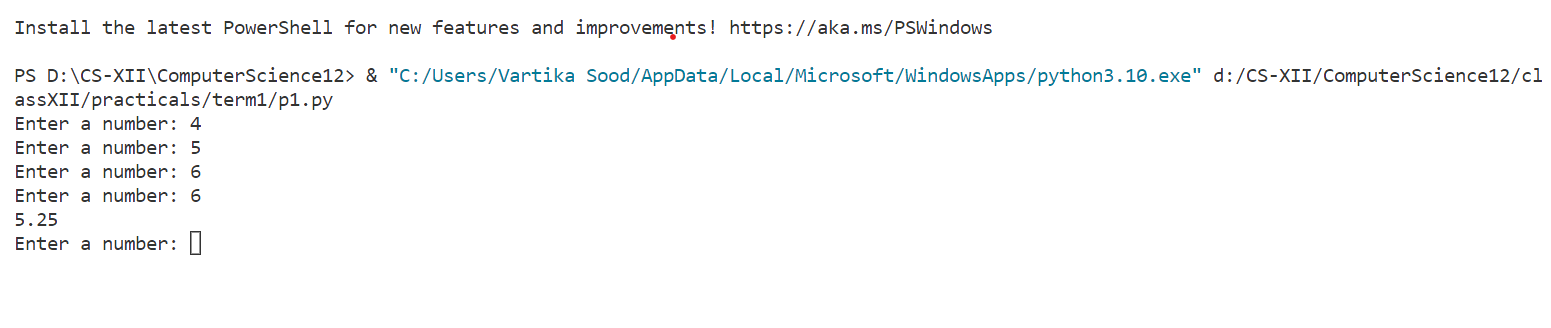
b=input("Enter a number: ")

c=input("Enter a number: ")

d=input("Enter a number: ")

avg2(a,b,c,d)

**Output:**

****

**Program:**

Write a Python program to make a function to find cube of a number given by the user.

**Source code:**

#23/3/22

#Method 1:

def cube(n):

    a=n\*n\*n

    return a

num=float(input("Enter a number: "))

print(cube(num))

#Method 2:

def cube2(n): return n\*\*3=

num2 = float(input("Enter a Number: "))

print(cube2(num2))

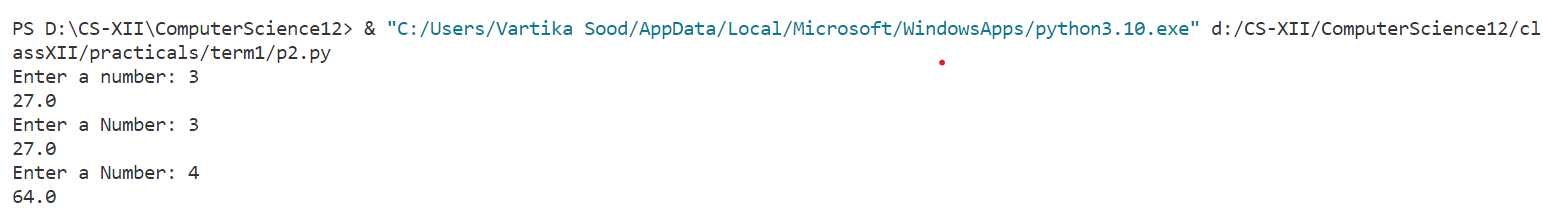
#Method 3:

cube3 = lambda n : n\*\*3

num3 = float(input("Enter a Number: "))

print(cube3(num3))

**Ouput:**



**Program:**

Write a Python program to make a function to find the area of the circle.

**Source code:**

#25/3/22

from math import \*

#Method 1:

def ar(rad):

    a=pi\*rad\*rad

    return a

r=float(input("Enter the radius: "))

print("The area is: ",ar(r))

#Method 2:

def ar2(rad):

    return pi\*(rad\*\*2)

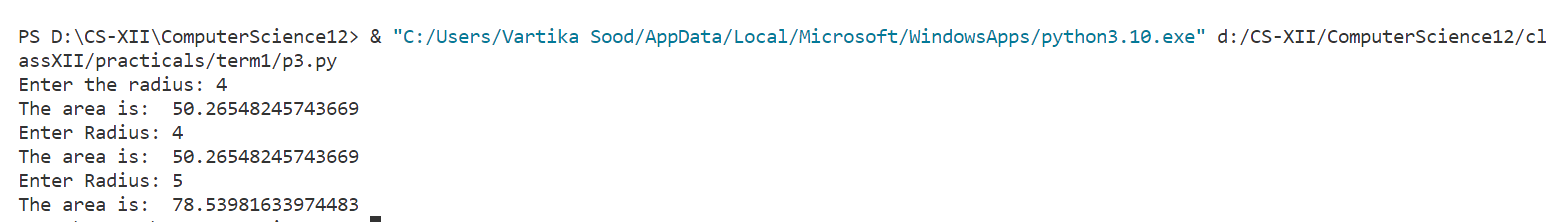
print(ar2(float(input("Enter Radius: "))))

#Method 3:

ar3 = lambda rad : pi\*(r\*\*3)

print(ar3(float(input("Enter Radius: "))))

**Output:**

****

**Program:**

Write a Python program to make a function to make a calculator.

**Source code:**

#28/3/22

from math import \*

def conNum(x):

    if float(x) == round(float(x)):

        return round(float(x))

    elif float(x) != round(float(x)):

        return round(float(x),3)

def simpleCalc():

    print("Welcome to Simple Calculator, + adds , - subtracts , x multiplies , / divides , ^ exponents, exit by typing 'out' in operation")

    cv = conNum(input("Enter number: "))

    while True:

        fun = input("Enter operation: ")

        if (fun == "+"):

            cv += conNum(input("Enter number: "))

            print(cv)

        if (fun == "-"):

            cv -= conNum(input("Enter number: "))

            print(cv)

        if (fun.lower() in ("x","\*") ):

            cv \*= conNum(input("Enter number: "))

            print(cv)

        if (fun == "/"):

            cv /= conNum(input("Enter number: "))

            print(cv)

        if (fun == "^"):

            v = conNum(input("Enter number: "))

            cv=pow(cv,v)

            print(cv)

        if (fun.lower() == "out"):

            break

def VolumeCalc():

    print("Welcome to Volume Calculator, write shape name to find volume ")

    shapeType = input("Enter Shape (Cube, Cuboid, Cone, Cylinder, Sphere, Hemisphere, Frustrum): ")

    if shapeType.lower() == "cube":

        print("The Volume of Cube is",

        conNum(input("Enter Side: "))\*\*3)

    if shapeType.lower() == "cuboid":

        print("The Volume of Cuboid is",

        conNum(input("Enter Length: "))\*conNum(input("Enter Breadth: "))\*conNum(input("Enter Height: ")))

    if shapeType.lower() == "cone":

        print("The Volume of Cone is",

        pi\*(conNum(input("Enter radius: "))\*\*2)\*conNum(input("Enter height: "))/3)

    if shapeType.lower() == "cylinder":

        print("The Volume of cylinder is",

        pi\*(conNum(input("Enter radius: "))\*\*2)\*conNum(input("Enter height: ")))

    if shapeType.lower() == "sphere":

        print("The Volume of Sphere is",

        4\*pi/3\*conNum(input("Enter radius: "))\*\*3)

    if shapeType.lower() == "hemisphere":

        print("The Volume of Hemisphere is",

        2\*pi/3\*conNum(input("Enter radius: "))\*\*3)

    if shapeType.lower() == "frustrum":

        print("The Volume of Frustrum is",fabs(

        (conNum(input("Enter radius 1: "))\*\*3 - conNum(input("Enter radius 2: "))\*\*3) \* pi / 3 \* conNum(input("Enter height: ")) ))

def SAcalc():

    print("Welcome to Surface Area Calculator, write shape name to find Surface Area ")

    shapeType = input("Enter Shape (Cube, Cuboid, Cone, Cylinder, Sphere, Hemisphere, Frustrum): ")

    if shapeType.lower() == "cube":

        print("The Surface Area of Cube is",

        6\*conNum(input("Enter Side: "))\*\*2)

    if shapeType.lower() == "cuboid":

        l=conNum(input("Enter Length: "))

        b=conNum(input("Enter Breadth: "))

        h=conNum(input("Enter Height: "))

        print("The Surface Area of Cuboid is",2\*(l\*b + b\*h + h\*l))

    if shapeType.lower() == "cone":

        r = conNum(input("Enter radius: "))

        h = conNum(input("Enter height: "))

        print("The Surface Area of Cone is",

        pi \* r \* (r + (r\*\*2 + h\*\*2)\*\*(1/2)   ))

    if shapeType.lower() == "cylinder":

        r = conNum(input("Enter radius: "))

        h = conNum(input("Enter height: "))

        print("The Surface Area of cylinder is",

        2\*pi\*r\*(r+h))

    if shapeType.lower() == "sphere":

        print("The Surface Area of Sphere is",

        4\*pi\*conNum(input("Enter radius: "))\*\*2)

    if shapeType.lower() == "hemisphere":

        print("The Surface Area of Hemisphere is",

        3\*pi\*conNum(input("Enter radius: "))\*\*2)

    if shapeType.lower() == "frustrum":

        print("The Surface Area of Frustrum is  ",

        (conNum(input("Enter radius 1: ")) + conNum(input("Enter radius 2: "))) \* pi \* (conNum(input("Enter height: ")) + 2) )

def TrigCalc():

    a = lambda x,y: x+" of "+str(y)+" is "+str(eval(x+"("+str(y)+")"))

    print("Welcome to trignometric calculator, What operation would you like to perform?")

    trigOp = input("Enter function(sin,cos,tan,asin,acos,atan): ").lower()

    if trigOp == "sin": print(a(trigOp , float(input("Enter Value: "))))

    elif trigOp == "cos": print(a(trigOp , float(input("Enter Value: "))))

    elif trigOp == "tan": print(a(trigOp , float(input("Enter Value: "))))

    elif trigOp == "asin": print(a(trigOp , float(input("Enter Value: "))))

    elif trigOp == "acos": print(a(trigOp , float(input("Enter Value: "))))

    elif trigOp == "atan": print(a(trigOp , float(input("Enter Value: "))))

def CashCalc():

    print("Welcome to Currency convertor: it converts the currencies to different forms")

    base = input("Enter a supported currency(USD,EUR,JPY,GBP,AUD,CAD,INR): ").upper()

    val = float(input("Enter amount:"))

    convertto = input("Enter 2nd supported currency(USD,EUR,JPY,GBP,AUD,CAD,INR): ").upper()

    if base=="USD":

        if convertto=="USD":print("Exchange Value is",val\*1)

        if convertto=="EUR":print("Exchange Value is",val\*0.93)

        if convertto=="JPY":print("Exchange Value is",val\*127.71)

        if convertto=="GBP":print("Exchange Value is",val\*0.78)

        if convertto=="AUD":print("Exchange Value is",val\*1.4)

        if convertto=="CAD":print("Exchange Value is",val\*1.27)

        if convertto=="INR":print("Exchange Value is",val\*76.69)

    if base=="EUR":

        if convertto=="USD":print("Exchange Value is",val\*1.07)

        if convertto=="EUR":print("Exchange Value is",val\*1)

        if convertto=="JPY":print("Exchange Value is",val\*136.97)

        if convertto=="GBP":print("Exchange Value is",val\*0.84)

        if convertto=="AUD":print("Exchange Value is",val\*1.5)

        if convertto=="CAD":print("Exchange Value is",val\*1.37)

        if convertto=="INR":print("Exchange Value is",val\*82.17)

    if base=="JPY":

        if convertto=="USD":print("Exchange Value is",val\*0.0078)

        if convertto=="EUR":print("Exchange Value is",val\*0.0073)

        if convertto=="JPY":print("Exchange Value is",val\*1)

        if convertto=="GBP":print("Exchange Value is",val\*0.0062)

        if convertto=="AUD":print("Exchange Value is",val\*0.011)

        if convertto=="CAD":print("Exchange Value is",val\*0.010)

        if convertto=="INR":print("Exchange Value is",val\*0.60)

    if base=="GBP":

        if convertto=="USD":print("Exchange Value is",val\*1.27)

        if convertto=="EUR":print("Exchange Value is",val\*1.19)

        if convertto=="JPY":print("Exchange Value is",val\*162.54)

        if convertto=="GBP":print("Exchange Value is",val\*1)

        if convertto=="AUD":print("Exchange Value is",val\*1.78)

        if convertto=="CAD":print("Exchange Value is",val\*1.62)

        if convertto=="INR":print("Exchange Value is",val\*97.62)

    if base=="AUD":

        if convertto=="USD":print("Exchange Value is",val\*0.71)

        if convertto=="EUR":print("Exchange Value is",val\*0.67)

        if convertto=="JPY":print("Exchange Value is",val\*91.2)

        if convertto=="GBP":print("Exchange Value is",val\*0.56)

        if convertto=="AUD":print("Exchange Value is",val\*1)

        if convertto=="CAD":print("Exchange Value is",val\*0.91)

        if convertto=="INR":print("Exchange Value is",val\*54.77)

    if base=="CAD":

        if convertto=="USD":print("Exchange Value is",val\*0.78)

        if convertto=="EUR":print("Exchange Value is",val\*0.73)

        if convertto=="JPY":print("Exchange Value is",val\*100.04)

        if convertto=="GBP":print("Exchange Value is",val\*0.62)

        if convertto=="AUD":print("Exchange Value is",val\*1.10)

        if convertto=="CAD":print("Exchange Value is",val\*1)

        if convertto=="INR":print("Exchange Value is",val\*60.08)

    if base=="INR":

        if convertto=="USD":print("Exchange Value is",val\*0.013)

        if convertto=="EUR":print("Exchange Value is",val\*0.012)

        if convertto=="JPY":print("Exchange Value is",val\*1.67)

        if convertto=="GBP":print("Exchange Value is",val\*0.010)

        if convertto=="AUD":print("Exchange Value is",val\*0.018)

        if convertto=="CAD":print("Exchange Value is",val\*0.017)

        if convertto=="INR":print("Exchange Value is",val\*1)

optionType = 0

print("Welcome to General Calculator, It has many options such as volume, area, trigonometry, currency, general.")

print("Select what you want to use")

while True:

    optionType = input("Enter your calculator type:\n   1)Volume\n   2)Surface Area\n   3)Trigonometry\n   4)Currency\n   5)General\n   6)Exit\n\n     ")

    if optionType.lower() == "volume" or optionType.lower() == "1":

        VolumeCalc()

    if optionType.lower() == "surface area" or optionType.lower() == "2":

        SAcalc()

    if optionType.lower() == "trigonometry" or optionType.lower() == "3":

        TrigCalc()

    if optionType.lower() == "currency" or optionType.lower() == "4":

        CashCalc()

    if optionType.lower() == "general" or optionType.lower() == "5":

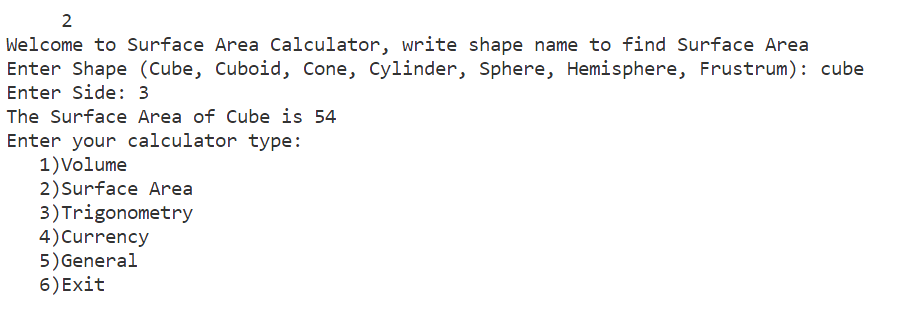
        simpleCalc()

    if optionType.lower() == "exit" or optionType.lower() == "6":

        break

**Output:**



****

**Program:**

Write a Python program to make a function to try different kinds of arguments.

**Source code:**

#4/4/22

def drawline(sym,t=20):

    for i in range(t):

        print(sym,end='')

    print()

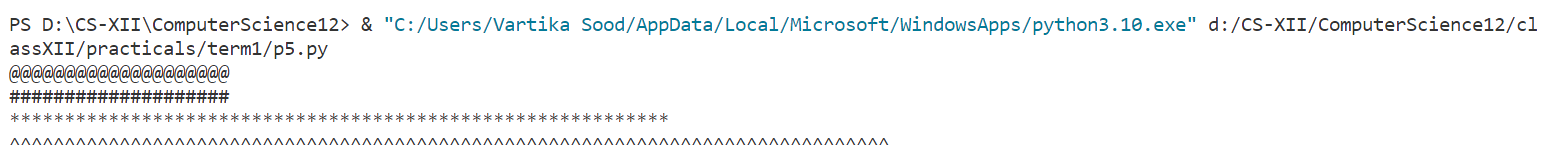
drawline(sym='@')

drawline('#')

drawline('\*',60)

drawline(t=80 ,sym='^')

**Output:**

****

**Program:**

Write a Python program to use different functions of random module.

**Source code:**

#8/4/22

import random

subj = ["Computer Science", "IP", "Physics", "Maths"]

#Method 1:

print(random.choice(subj))

#Method 2:

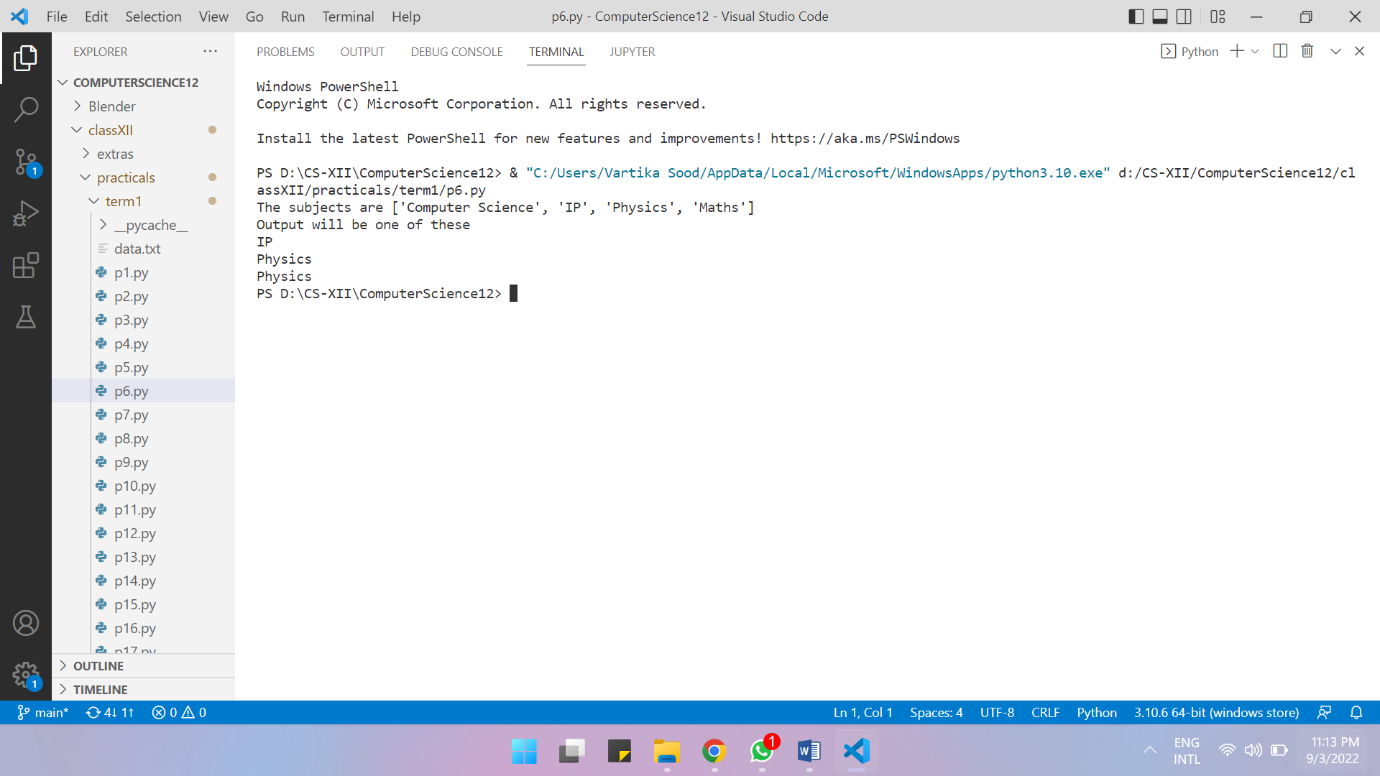
print(subj[random.randrange(3)])

#Method 3:

random.shuffle(subj)

print(subj[0])

**Output:**



**Program:**

Write a Python program to calculate sum of the digits of a random three-digit number.

**Source code:**

#8/4/22

from random import random

n=(random()\*900+100)//1

print('The number is: ',n)

s = 0

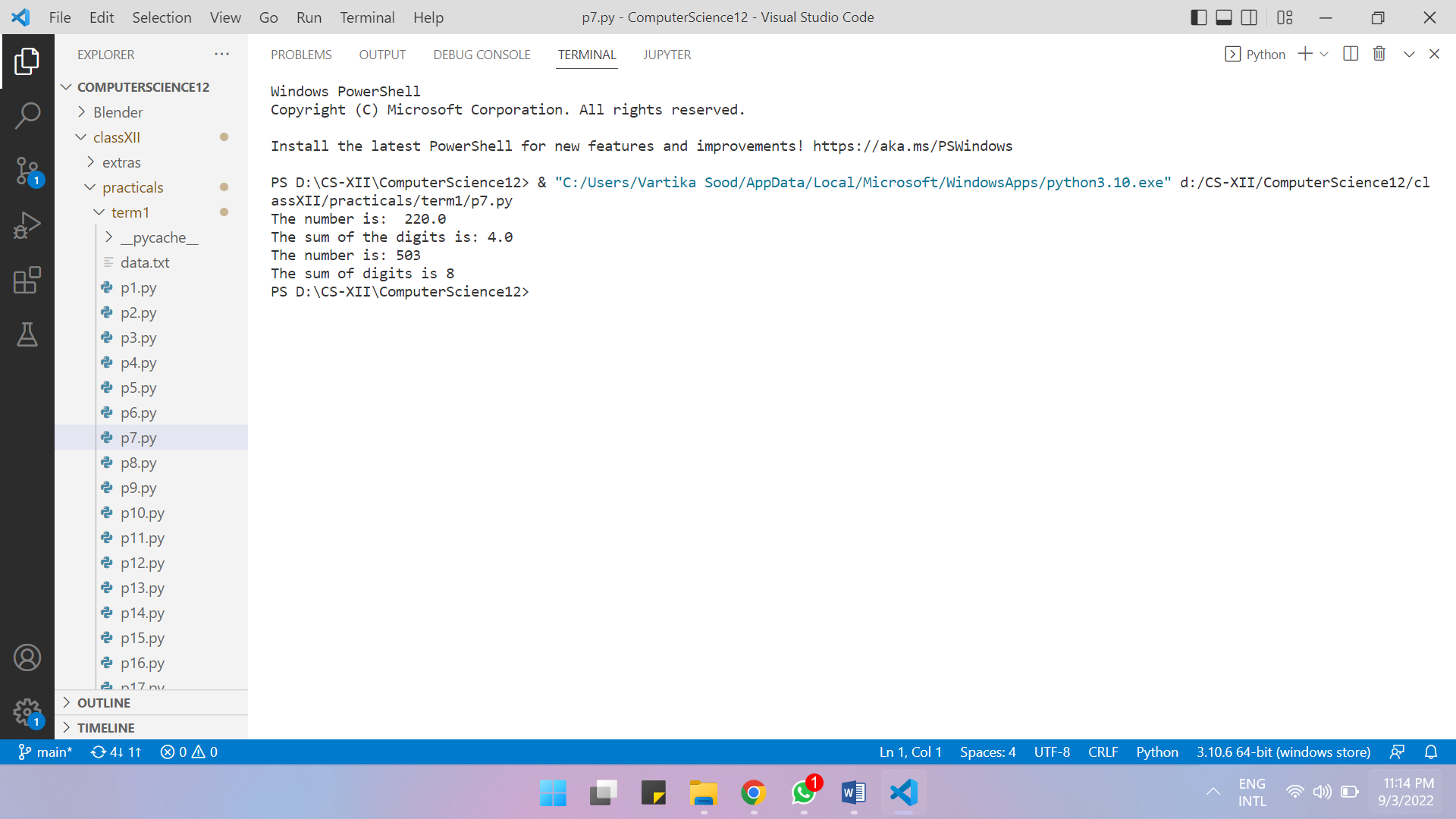
while n >0:

    s +=n%10

    n//=10

print('The sum of the digits is:',s)

**Output:**



**Program:**

Write a Python program that fills a list with numbers.

**Source code:**

#11/4/22

from random import randint

def fill\_list(L , num , min , max):

    for \_ in range(num): L.append(randint(min,max))

low , up , elem = int(input("Enter Minimum: ")) , int(input("Enter Maximum: ")) , int(input("Enter total terms: "))

a = []

b = []

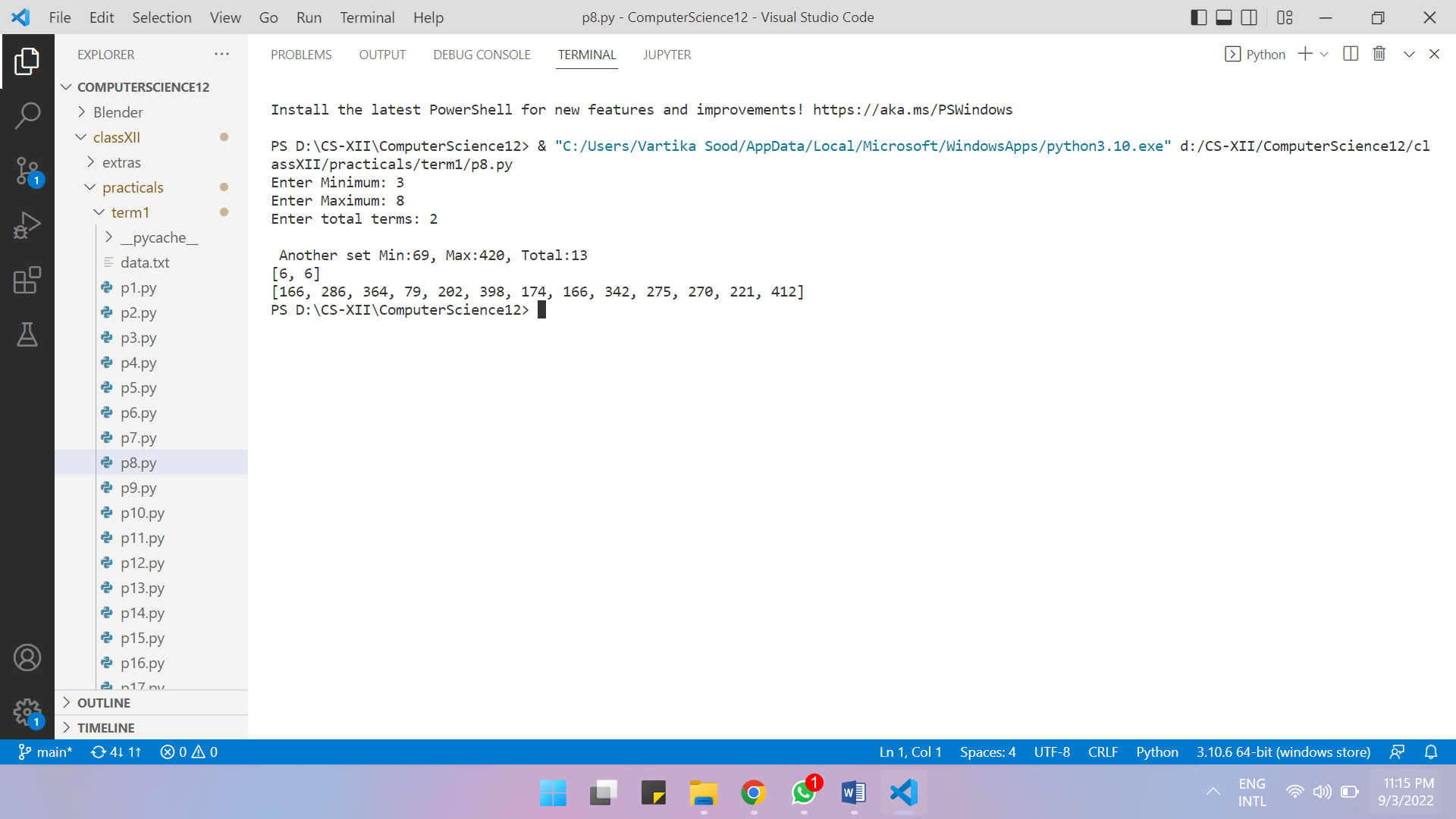
fill\_list(b,elem,low,up)

fill\_list(a , min=420 , max=4200 , num=69)

print(b)

print(a)

**Output:**



**Program:**

Write a Python program to perform binary search using randint().

**Source code:**

#18/4/22

from random import randint

def bin (l,el):

    mid=len(l)//2

    low=(0)

    high=len(l)-1

    while l[mid]!=el and low<=high:

        if el>l[mid]:

            low=mid+1

        else:

            high=mid-1

        mid=(high+low)//2

    if low>high:

        return None

    else:

        return mid

a=[]

for \_ in range(10):

    a.append(randint(1,20))

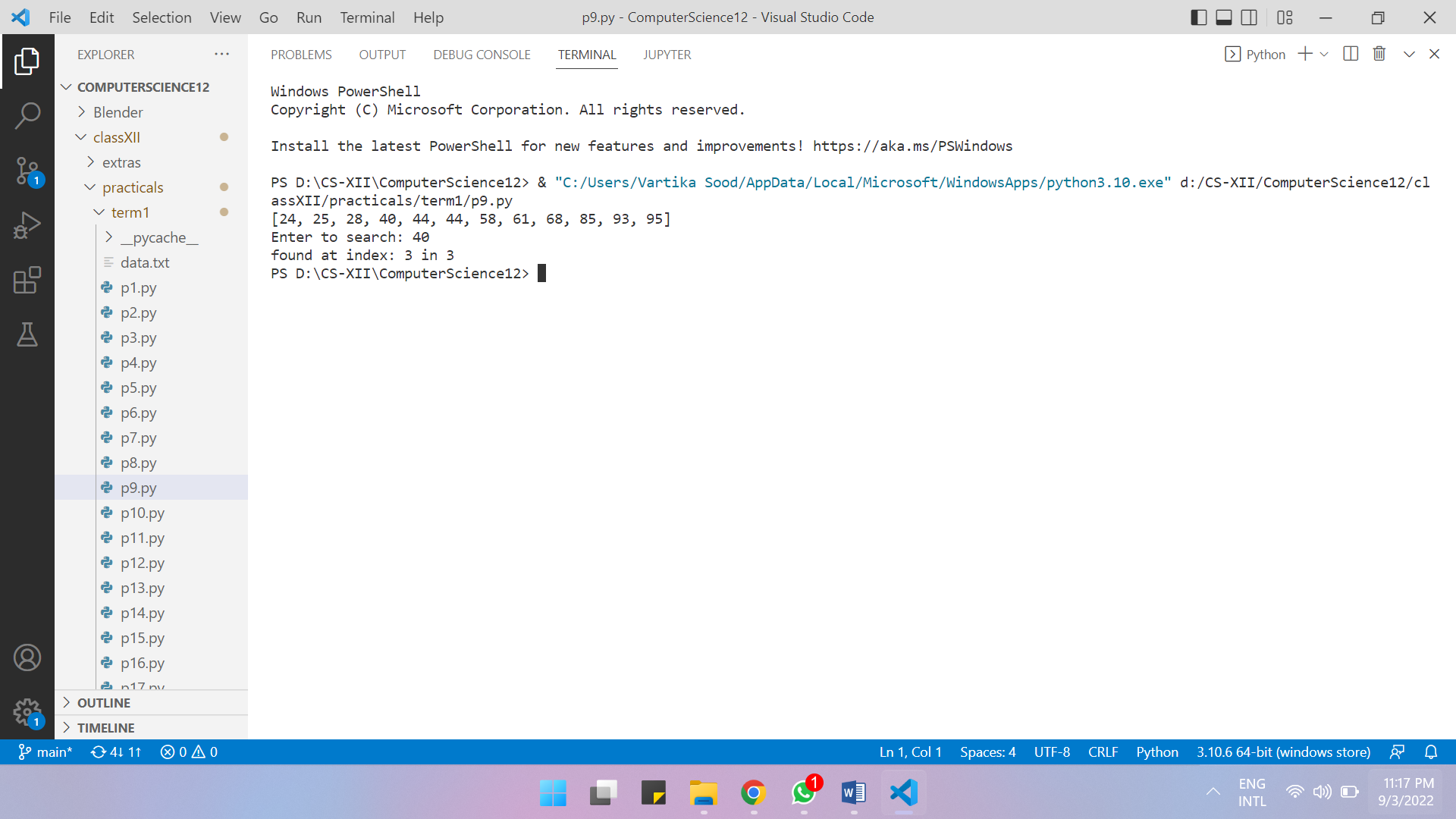
a.sort()

print(a)

v=int(input('Enter to search: '))

print("found: ",bin(a,v))

**Output:**



**Program:**

Write a Python program to make a lottery game.

**Source code:**

#19/4/22 lottery genera

from random import uniform

print("Lottery number is between 1-100")

a = uniform(0,100)

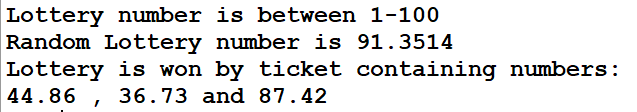
print("Random Lottery number is",a)

print("Lottery is won by ticket containing numbers: ")

a = lambda : uniform(1,100)//0.01 /100

print(a(),",",a(),"and",a())

**Output:**

****

**Program:**

Write a Python program to choose on random direction.

**Source code:**

#21/4/22

import random

direction = random.choice(["East", "West", "North","South"])

print("Randomly selected cardinal direction is",direction)

print("Random non cardinal Direction is",d()+"-"+d())

def d():

    dir = random.choice(["East", "West", "North","South"])

    return dir

**Output:**

****

**Program:**

Write a Python program to make a function to count the vowels in a string.

**Source code:**

#22/4/22

#Method 1:

def countvowel(s):

    c=0

    for ch in s:

        if ch in 'aeiouAEIOU':

            c+=1

    return c

sin=input('Enter a string: ')

count=countvowel(sin)

print('Total number of vowels in the string are: ',count)

#Method 2:

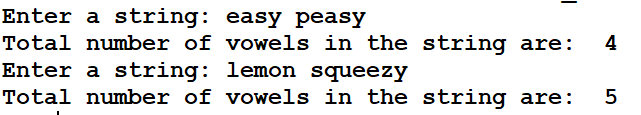
def vowelcount(st): return len(['' for k in st if st.lower() in "aeiou"])

a=input('Enter a string: ')

d=countvowel(a)

print('Total number of vowels in the string are: ',d)

**Output:**

****

**Program:**

Write a Python program to make a higher and lower number guessing game.

**Source code:**

#22/4/22

#Higher lower game

import random

initial = round(100000 \* random.random())

while True:

    print("number is: ",initial)

    print("Higher or lower\n")

    new = round(100000 \* random.random())

    input()

    if new>initial:

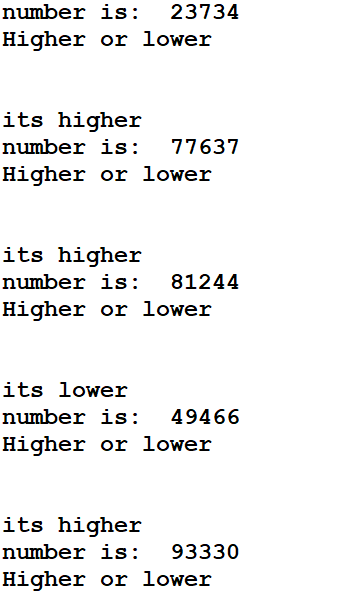
        print('its higher')

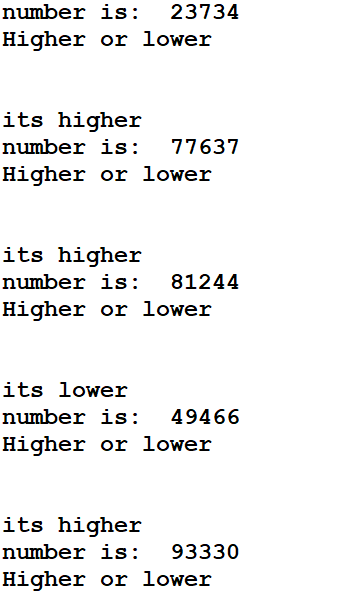
    if new<initial:

        print('its lower')

    initial = new

**Output:**

****

****

**Program:**

Write a Python program to count the frequency of a character in a string.

**Source code:**

#22/4/22

def cch(s,ch):

    c=0

    for i in s:

        if i==ch:

            c+=1

    return c

sin=input('Enter a string: ')

ch1=input('Enter the character to count: ')

f=cch(sin,ch1)

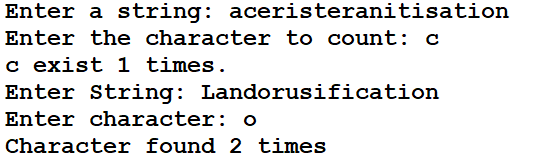
if f==0:

    print(ch1, "does not exist.")

else:

    print(ch1,'exist',f,'times.')

**Output:**



**Program:**

Write a Python program to make a function having tuple as the argument.

**Source code:**

#22/4/22

#passing immutable tuple to a function

def ttl(A):

    A=list(A)

    A[0]=A[0]\*2

    A[1]=A[1]+10

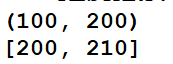
    print(A)

t=(100,200)

print(t)

ttl(t)

**Output:**



**Program:**

Write a Python program to determine whether the elements in the tuple are even or odd.

**Source code:**

#22/4/22

def countOddEven(t):

    odd = len(['' for k in t if k%2==1])

    return odd , len(t) - odd

store = tuple()

times = int(input("Number of elements: "))

for \_ in range(times):

    store+= (int(input("Enter Element "+str(\_+1)+": ")),)

a = countOddEven(store)

print("Odd numbers are",a[0] , "and Even numbers are",a[1])

**Output:**



**Program:**

Write a Python program to pass student record as a dictionary to a function and update their marks.

**Source code:**

#22/4/22

def marksu(s,nm):

    s['Marks']+=nm

    s["Status"]="Updated"

s1={'Name':'Akash','Marks':56,'Status':'Old'}

s2={'Name':'Chinmay','Marks':60,'Status':'Old'}

s3={'Name':'Chirag','Marks':50,'Status':'Old'}

print("Original data: ")

print(s1)

print(s2)

print(s3)

marksu(s1,70)

marksu(s2,80)

marksu(s3,75)

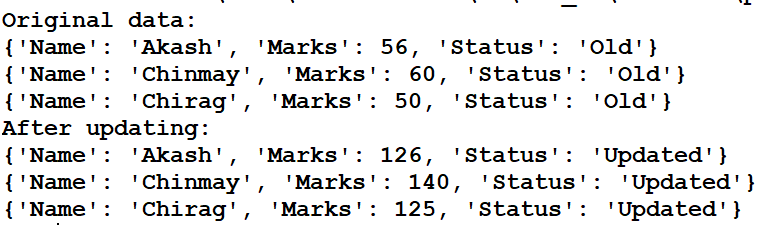
print("After updating: ")

print(s1)

print(s2)

print(s3)

**Output:**

****

**Program:**

Write a Python program to pass dictionary to a function with list of elements as keys and frequency of its occurrence as value and return as a dictionary.

**Source code:**

#23/4/22

#checks the number of times an element exists in the list.

def freq(l,d):

    for i in l:

        if i not in d:

            d[i]=1

        else:

            d[i]+=1

    return d

l1=[1,2,31,4,1,4,6,2,1,6,6,2]

d1={}

freq(l1,d1)

print(d1)

**Output:**

****

**Program:**

Write a python program to replace the alphabets with given alphabets.

**Source code:**

#4/6/22

name = "Madam"

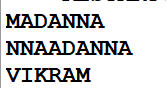
print(name.replace('m','nna').upper())

print(name.upper().replace("M",'NNA'))

name="Vikram"

print(name.replace('im','nt').upper())

**Output:**



**Program:**

List Mutation

**Source code:**

#6/5/22

GivenList1 = [458,646,64,385,48,364,77,62,43,59,78]

print("List Before Function is", GivenList1)

def OddEven1(Ls):

    for k in range(len(Ls)):

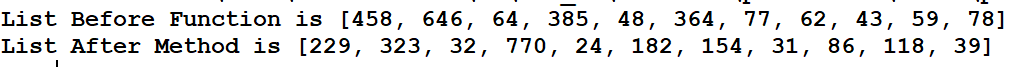
        if Ls[k] %2==0 : Ls[k]//=2

        else: Ls[k]\*=2

OddEven1(GivenList1)

print("List After Method is", GivenList1)

**Output:**



**Program:**

**Matrices and diagonals**

**Source code:**

#6/5/22

def MatrixPrintDiagonal(l):

    for i in range(len(l)):

        for j in range(len(l[i])):

            if i==j:

                print(l[i][j],end='\t')

            else:

                print('',end='\t')

        print()

m=[

    [1,2,3],

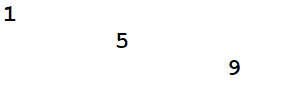
    [4,5,6],

    [7,8,9]

]

MatrixPrintDiagonal(m)

**Output:**



**Program:**

**Averages**

**Source code:**

#6/5/22

def avgs(l):

    s=0

    for i in l:

        s+=i

    avg=s/len(l)

    return s,avg

l1=[]

n=int(input('Enter how many numbers: '))

for i in range(n):

    num=int(input('Enter any number: '))

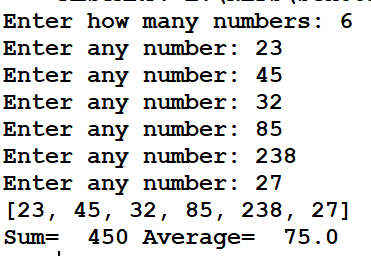
    l1.append(num)

print(l1)

savg=avgs(l1)

print('Sum= ',savg[0],'Average= ',savg[1])

**Output:**



**Program:**

Odd even counter

**Source code:**

#6/5/22

def OEcount(mytuple):

    even = len(["" for k in mytuple if k%2==0])

    odd = len(mytuple) - even

    return odd,even

n = int(input("Number of Numbers: "))

tup = ()

for \_ in range(n): tup += (int(input("Enter Number: ")),)

o,e = OEcount(tup)

print("Odd in tuple is",o,"and even in tuple is",e)

**Output:**



**Program:**

Swapcase and jumble

**Source code:**

#19/5/22

def j(s):

    sn1=''

    for i in range(len(s)):

        if sn1[i].islower: sn1+=s[i].upper()

        elif sn1[i].isupper: sn1+=s[i].lower()

        elif sn1[i].isdigit: sn1+="\*"

        else: sn1+='@'

    return sn1

print(“LumberiNATIONi5aster\*2001\*11\*09"))

**Output:**



**File Handling**

**Program:**

**Removing Eol and blankspaces**

**Source code:**

#6/6/22

import sd

myfile=open('p28.txt','r')

str1=' '

size=0

tsize=0

while str1:

    str1=myfile.readline()

    tsize=tsize+len(str1)

    size=size+len(str1.strip())

print('Total size: ',tsize)

print('Size after removing EOL and blank spaces: ',size)

myfile.close()

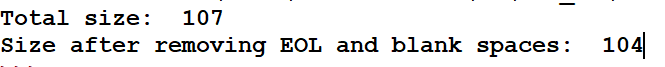
**Text file:**

Hye!

This is my first program in file handling.

I am practicing to make a project for the midterm project.

**Output:**



**Program:**

**File Handling**

**Source code:**

#6/6/22

import sd

#method 1 to read

myfile1 = open("p29.txt", "r")

for line in myfile1:

    print(line,end="")

print()

myfile1.close(

#method 2 to read

myfile1 = open("p29.txt", "r")

for line in myfile1.readlines():

    print(line.rstrip("\n"))

myfile1.close()

#method 3 to read

myfile1 = open("p29.txt", "r")

a = [line[:-1] for line in myfile1]

print(a)

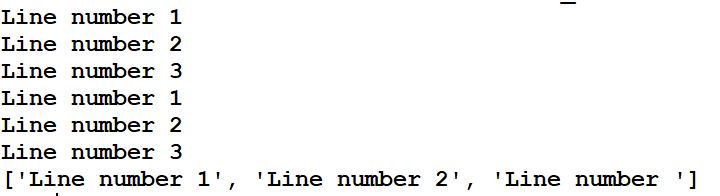
myfile1.close()  
**Text file:**

Line number 1

Line number 2

Line number 3

**Output:**



**Program:**

**Source code:**

#6/6/22

import sd

myfile=open('p30.txt','w')

myfile.writelines('Line number 4')

myfile.close()

print('Data saved in the file.')

**Text file before : (empty)**

**Output:**



**Text file after :**

Line number 4

**Program:**

**Copying data (#1)**

**Source code:**

#6/6/22

import sd

myfile = open("p31.txt" , 'r+')

data = [line for line in myfile]

[print(k.rstrip("\n")) for k in data]

myfile.seek(0)

myfile.writelines(data)

myfile.close()

**Text file:**

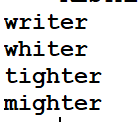
writer

whiter

tighter

mighter

**Output:**



**Program:**

**Storing Input data**

**Source code:**

#6/6/22

import sd

myfile=open('p32.txt','w')

for i in range(2):

    name=input('Enter a name to store in the file: ')

    myfile.write(name)

    myfile.write('\n')

myfile.close()

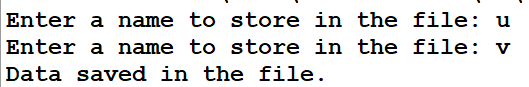
print('Data saved in the file.')

**Text file:**

u

v

**Output:**

****

**Program:**

**Saving File Data**

**Source code:**

#6/6/22

import sd

myfile=open('p33.txt','a')

for i in range(2):

    name=input('Enter a name to store in the file: ')

    myfile.write(name)

    myfile.write('\n')

myfile.close()

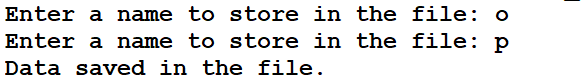
print('Data saved in the file.')

**Text file (before):**

v

u

**Output:**

****

**Text file (after):**

v

u

o

p

**Program:**

**Book Storage (Library System)**

**Source code:**

#7/6/22

import sd

myfile=open("p34.txt",'a')

ans='y'

while ans=='y':

    bn=int(input('Enter book number: '))

    bname=input('Enter book name: ')

    au=input('Enter book author: ')

    price=int(input('Enter book price: '))

    brec=str(bn)+','+bname+','+au+','+str(price)

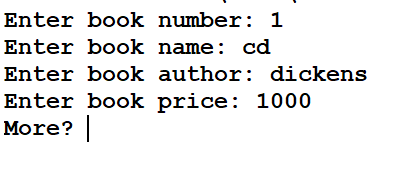
    myfile.write(brec)

    ans=input('More? ')

myfile.close()

**Text file:**

1,cd,dickens,1000

**Output:**

**Program:**

**Copy data (#2)**

**Source code:**

#06-07-2022

#file copy one from another

import sd

#method1

file1 = open("p35a.txt", 'r')

file2 = open("p35b.txt", 'w')

str = " "

while str:

    str = file1.readline()

    file2.write(str)

file1.close()

file2.close()

#method2

file1 = open("p35a.txt", 'r')

file2 = open("p35b.txt", 'w')

[file2.write(line) for line in file1.readlines()]

file1.close()

file2.close()  
**Text file(before):**

ace of hearts

king of diamonds

queen of spades

jack of all trades

seven of heavens

**Text file (After):**

ace of hearts

king of diamonds

queen of spades

jack of all trades

seven of heavens

**Program:**

**Append Data**

**Source code:**

#7/6/22

import sd

myf1=open('p34.txt','r')

myf2=open('p36.txt','a')

s=' '

while s:

    s=myf1.readline()

print(s)

    myf2.write(s)

myf1.close()

myf2.close()

**Output:**

****

**Text file (write):**

1,cd,dickens,1000

**Program:**

**Flushing data to load to disk from memory.**

**Source code:**

#7/6/22

import sd

myfile=open('p37.txt','w+')

myfile.writelines('Hye')

myfile.flush()

input("Next: ")

myfile.writelines('\nThe return of Hi')

myfile.flush()

input("Next: ")

myfile.writelines('\nHi, the wrath of Hello')

myfile.flush()

input("Next: ")

myfile.writelines('\nThe last Hi')

myfile.close()

print('Data saved in the file.')

**Text file:**

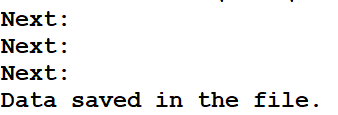
Hye

The return of Hi

Hi, the wrath of Hell

The last Hi

**Output:**



**Program:**

**Using Seek to write at specific location**

**Source code:**

#07-07-2022

#line cursor setter

import sd

def lineset(filen,Lnum,pos):

    chars = [len(line) for line in filen.readlines()]

with open('p38.txt','r+') as file1:

    file1.seek(12)

    file1.write("1")

**Text file:**

Abcdefghijkl1no

pqrstuvwxyz

**Output:**

Abcdefghijkl1no

pqrstuvwxyz

**Program:**

**Pickling and binary data  
Source code:**

#8/7/22

import sd

from pickle import \*

mf=open('p39.txt','r+b')

val=load(mf)

print(val)

d={1:'p',2:'b',3:'c',4:'d'}

mf.seek(0)

dump(d,mf)

mf.close()

**Text file:**

��}�(K�p�K�b�K�c�K�d�u.

**Output:**



**Program:**

**Writing binary datas**

**Source code:**

#8/7/22

import sd

from pickle import \*

with open('p40.txt', 'w+b') as f:

    dump({1:'p',2:'b',3:'c',4:'d'},f)

    f.seek(0)

    data = load(f)

    print(data)

**Text file:**

��}�(K�p�K�b�K�c�K�d�u.

**Output:**



**Program:**

**White space cleaner program**

**Source code:**

#12/7/22

import sd

f1 = open("p41a.txt", 'r')

f2 = open("p41b.txt", 'w')

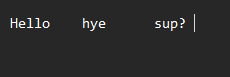
for i in f1:

    f2.write(' '.join(i.split()))

f1.close()

f2.close()

**Text file (before):**



**Text file (after):**

****

**Program:**

**Copying data via pickling**

**Source code:**

#12/7/22

import sd, pickle

with open("p42a.txt" , "r+b") as f1:

    with open("p42b.txt" , 'r+b') as f2:

        output = ""

        for word in str(pickle.load(f1)).split() : output+= word+" "

        pickle.dump(output,f2)

with open("p42a.txt" , "r+b") as f1:

    with open("p42b.txt" , 'r+b') as f2:

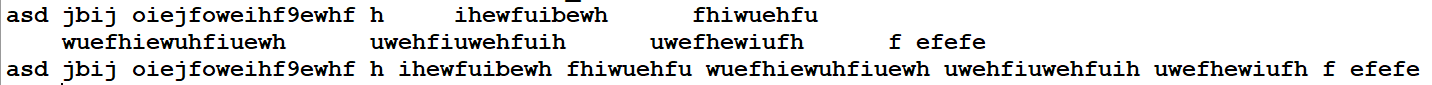
        print(pickle.load(f1))

        print(pickle.load(f2))  
**Text file:**

�����asd jbij oiejfoweihf9ewhf h     ihewfuibewh      fhiwuehfu

    wuefhiewuhfiuewh      uwehfiuwehfuih      uwefhewiufh      f efefe    �.

**Output:**



**Program:**

**Formatting and locating by specific keyword**

**Source code:**

#12/7/22

import sd

f1 = open("p43a.txt", 'r')

f2 = open("p43b.txt", 'w')

str = " "

while str:

    str = f1.readline().strip('\n')

    w = str.split("~")

    if w[0] == "A":

        f2.write(str+"\n")

print(str)

f1.close()

f2.close()

**Text file:**

A~ps

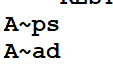
a~vs

A~ad

a~om

a~uk

**Output:**



**Program:**

**Counting ‘to’ amd ‘the’ in sentence (activity)**

**Source code:**

#12/7/22

#to and the counter

import sd

with open("p44a.txt", 'r+') as f1:

    data = [k.rstrip("\n") for k in f1.readlines()]

    ans = len(["" for k in data for word in k.split() if word.lower() in ("the","to")])

    print(ans)

**Text file:**

This notice is to inform the organisation that the last day for paying the rent is tomorrow.

**Output:**



**Program:**

**Making a data storage system with use of classes**

**Source code:**

#14-07-2022

#address locationalistation

#project uses custom class student and allows entry and removal of data from the records

#import of libraries

import sd,pickle

#defining required classes

class Student:

    def \_\_init\_\_(self, name, addm, cls , sec):

        self.name = name

        self.addm = addm

        self.cls = cls

        self.sec = sec

    def \_\_dir\_\_(self):

        output , c = [] , 1

        for val in (self.name, self.addm, self.cls , self.sec):

            output.append(str(c)+str(val))

            c+=1

        return output

#defining useful functions

#Import data: parameters: File name , return list of records

def importData(fname):

    #reads data, if cant read data, then assumes file DNE, and creates file, if creation fails, then assumes empty or corrupt file and writes blank list to file, returns blank, or data

    try:

        with open(fname,"rb") as f1:

            data = pickle.load(f1)

            return data

    except:

        try:

            with open(fname,"xb") as f1:

                pass

        except:

            with open(fname,'wb') as f1:

                pickle.dump(list(),f1)

        return []

def updateRecords(data, fname):

    #If write successful, returns true else false

    try:

        with open(fname, "wb") as f1:

            pickle.dump(data,f1)

            return True

    except: return False

#working Output Formats

def dataFormat(data):

    #formats data to present in console

    return ["%-15s|%9s|%5s|%-2s"%tuple([x[1:] for x in dir(k)]) for k in data]

#defining useful global variables

Fn = "p45.txt"

#main code

records = importData(Fn)

count = int(input("Number of Students: "))

for \_ in range(count):

    a = Student(input("Name: "),

                int(input("Admission Number: ")),

                int(input("Class: ")),

                input("Section: "))

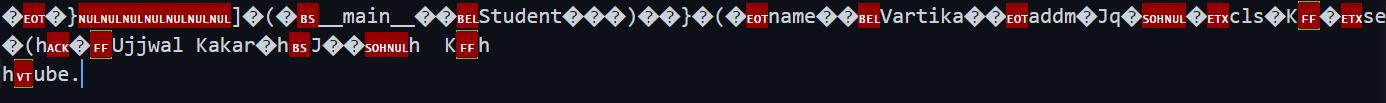
    records.append(a)

if updateRecords(records,Fn):

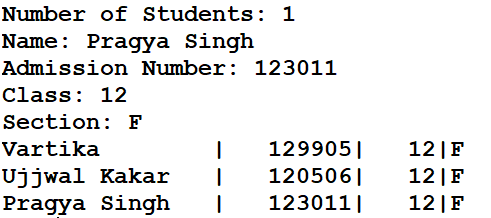
    for val in dataFormat(importData(Fn)):

        print(val)

**Text file:**



**Output:**



**Program:**

**Calcute total number of captal letters in sentence.**

**Source code:**

#13/7/22

import sd

f1 = open("p46.txt", 'w')

s=f1.readline()

c=0

for i in s:

    if i.isupper():

        c+=1

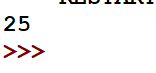
print(c)

f1.close(0)

**Text file:**

Line has been written here. THIS IS ALL CAPS. All caps are rude. DONT BE RUDE!

**Output:**



**Program:**

**Classes to simplify data structure in file handling**

**Source code:**

#25/7/22

import sd

from pickle import \*

class student:

    def \_\_init\_\_(self, name, cls, per , admno):

        self.name = name

        self.cls = cls

        self.per = per

        self.admno = admno

class Collection:

    def \_\_init\_\_(self,dirpath):

        self.dirpath = dirpath

        def filecheck(location):

            try: a = load(open(location,'rb'))

            except: a = open(location,'wb') ; dump([],a) ; print("Exception occurs")

        filecheck(dirpath)

        #basis filerefresh

        with open(dirpath,"rb") as f1:

            self.data = load(f1)

            self.totalnum = len(self.data)

    def filerefresh(self):

        with open(self.dirpath,"rb") as f1:

            self.data = load(f1)

            self.totalnum = len(self.data)

    def add(self,info):

        self.data+=[info]

        with open(self.dirpath,'wb') as f1:

            dump(self.data,f1)

        self.filerefresh()

    def remove(self,name = "", admno = ""):

        if name == "" and admno=="":

            return False

        elif name == "":

            for k in self.data:

                if k.admno == admno:

                    self.data.remove(k)

        else:

            for k in self.data:

                if k.name == name:

                    self.data.remove(k)

        with open(self.dirpath,'wb') as f1:

            dump(self.data,f1)

    def edit(self, Gadmno):

        if Gadmno in [k.admno for k in self.data]:

            k = [stu for stu in self.data if stu.admno == Gadmno][0]

            k.name,k.cls,k.per = input("Enter Name: "), input("Enter Class: "),input("Enter Percentage: ")

            self.filerefresh()

            return True

        else:

            return False

students = Collection('p48.txt')

students.add(student("Vartika",12,99.999,1))

print("Added Student")

print([a.name for a in students.data])

students.edit(1)

print("Edited Rn.1")

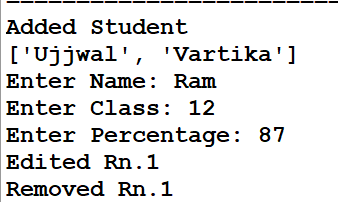
students.remove(admno=1)

print("Removed Rn.1")

**Text file:**



**Output:**



**Program:**

**Using CSV and formatting to make a table**

**Source code:**

#26/7/22

import sd,csv

with open('p50.csv') as c:

    mr=csv.reader(c)

    print('%10s'%'EMP. NO.','%20s'%'EMP. NAME','%10s'%'SALARY',)

    for i in mr:

        print('%10s'%i[0],'%20s'%i[1],'%10s'%i[2])

**Text file:**

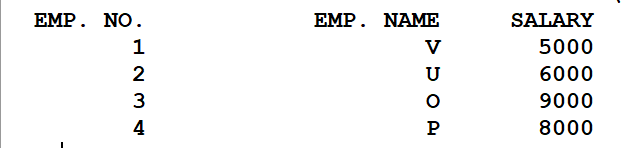
1,V,5000

2,U,6000

3,O,9000

4,P,8000

**Output:**



**Program:**

**CSV to MySQL type formatting of table with max length in mind**

**Source code:**

#27/7/2022

#csv working

import sd , csv

with open("p51.csv",'r') as f1:

    dat = csv.reader(f1)

    dat = [line for line in dat]

    maxl = max([len(line) for line in dat])

    for line in dat:

        while len(line)!=maxl:

            line.append('')

    print(dat)

    maxlist = []

    for times in range(len(line)):

        maxval = 0

        for line in dat:

            if len(line[times]) > maxval:

                maxval = len(line[times])

        maxlist.append(maxval)

    strR = "|"

    for length in maxlist:

        strR = strR + r"%-"+str(length)+"s|"

    print("\_"\*(sum(maxlist)+len(maxlist)+1))

    for line in dat:

        # print(line)

        print(str(strR)%tuple(line))

    print("‾"\*(sum(maxlist)+len(maxlist)))

**Text file:**

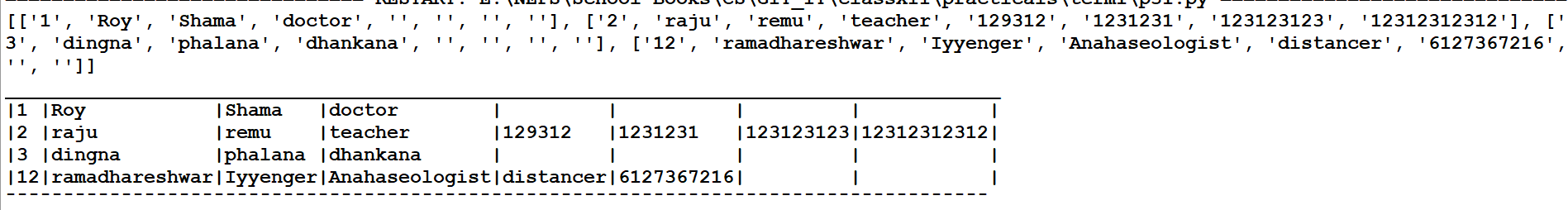
1,Roy,Shama,doctor

2,raju,remu,teacher,129312,1231231,123123123,12312312312

3,dingna,phalana,dhankana

12,ramadhareshwar,Iyyenger,Anahaseologist,distancer,6127367216

**Output:**

****

**Program:**

**Listing from CSVs**

**Source code:**

#27/7/22

import sd,csv

with open('p50.csv') as c:

    mr=csv.reader(c)

    co=0

    s=0

    print('%10s'%'EMP. NO.','%20s'%'EMP. NAME','%10s'%'SALARY',)

    for i in mr:

        print('%10s'%i[0],'%20s'%i[1],'%10s'%i[2])

        s+=int(i[2])

        if int(i[2])>8000:

            co+=1

    print("Sum of salaries- ",s)

    print('Number of employees with salary more than 7000- ',co)

**Text file:**

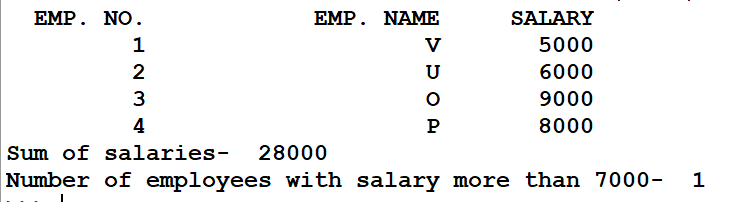
1,V,5000

2,U,6000

3,O,9000

4,P,8000

**Output:**



**Stacks**

**Program:**

Stacks Application

**Source code:**

#10/8/22

#functions

def emp(S):

    if len(S)==0:

        return True

    else:

        return False

def push(S,a):

    S.append(a)

    top=len(S)-1

def pop(S):

    if emp(S):

        return "Underflow"

    else:

        val=S.pop()

        if len(S)==0:

            top=None

        else:

            top=len(S)-1

        return val

def peek(S):

    if emp(S):

        return 'Underflow'

    else:

        top=len(s)-1

        return S[top]

def show(S):

    if emp(S):

        print("Sorry, empty stack")

    else:

        t=len(S)-1

        print('(Top)',end='')

        while(t>=0):

            print((S[t]))\

                t-=1

        print()

#main program

S=[]

top=None

while True:

    print("STACKS FUNCTIONS")

    print("1) PUSH")

    print("2) POP")

    print("3) PEEK")

    print("4) SHOW STACK")

    print("5) EXIT")

    n=int(input("Enter your choice:" ))\

    if ch==1:

        v=int(input("Enter item to add: "))

        push(S,v)

    elif ch==2:

        v=pop(S)

        if v=='Underflow':

            print('Stack empty')

        else:

            print("Deleted item: ",v)

    elif ch==3:

        v=peek(S)

        if v=='Underflow':

            print('Stack empty')

        else:

            print("Top item: ",v)

    elif ch==4:

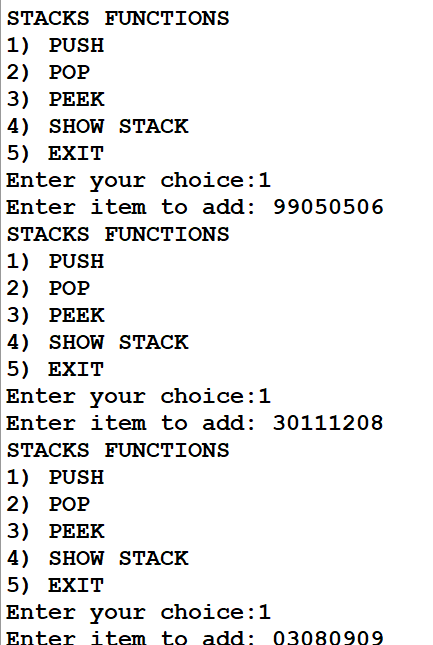
        show(S)

    elif ch==5:

        print("Bye")

        break

**Output:**





**Program:**

Playing Cards Peeker

**Source code:**

#Stacks peeking through a deck of cards

import random as r

stack = []

L = 0

top=None

types = ["Spades","Clubs","Hearts","Diamonds"]

numbers = ["Ace","Two","Three","Four","Five","Six","Seven","Eight","Nine","Ten","Jack","Queen","King"]

deck = [a+" of "+b for a in numbers for b in types]

#adding

for card in deck:

stack.append(card)

L+=1

top = L-1

r.shuffle(stack)

#peeking

print("The current card is \"%-12s\""%stack[top])

while True:

if input("Do you want to see next card? (Y/N)").lower() == "n":

break

#removal

stack.pop(top)

L-=1

top = L-1

print("The new card is \"%-12s\""%stack[top])

**Output:**

