|  |  |
| --- | --- |
| ***Index*** | |
| **1.** | **write a program that reads a line and a substring. It should then display the no of occurrences of the given substring in the line.** |
| **2.** | **Write a Python program to remove the nth index character from a nonempty string.** |
| **3.** | **Python program to find uncommon words from two Strings.** |
| **4.** | **Reverse a string using stack(in list form).** |
| **5.** | **Sum of series (1) + (1+2) +(1+2+3)+........upto N terms.** |
| **6.** | **Sum of series (1) + x/2! + x^2/4! + x^3/6!.upto N terms.** |
| **7.** | **Sum of series 1^2 + 2^2 + 3^2 +........upto N terms.** |
| **8.** | **Implementation of insertion sort.** |
| **9.** | **Implementation of bubble sort.** |
| **10.** | **Write a program to read the contents of both the files created in the above programs and merge the contents into “merge.txt”. Avoid using the close() function to close the files.** |
| **11.** | **Write a program to count a total number of lines and count the total number of lines starting with ‘A’, ‘B’, and ‘C’. (Consider the merge.txt file).** |
| **12.** | **Count the uppercase characters in a file.** |
| **13.** | **How to create a table in MySQL from Python. Let’s create table ‘Laptop’ under the ‘Electronics’ database.** |
| **14.** | **Insert multiple rows into MySQL table.** |
| **15.** | **Update mySQL table records using python.** |

**Programs**

**#1. write a program that reads a line and a substring. It should then display the no of occurrences of the given substring in the line.**

line = input("Enter line ")

substr = input("Enter substring ")

lenline = len(line)

lensub = len(substr)

start=count=0

end = lenline

while True:

pos = line.find(substr,start,end)

if pos !=-1:

count +=1

start = pos + lensub

else :

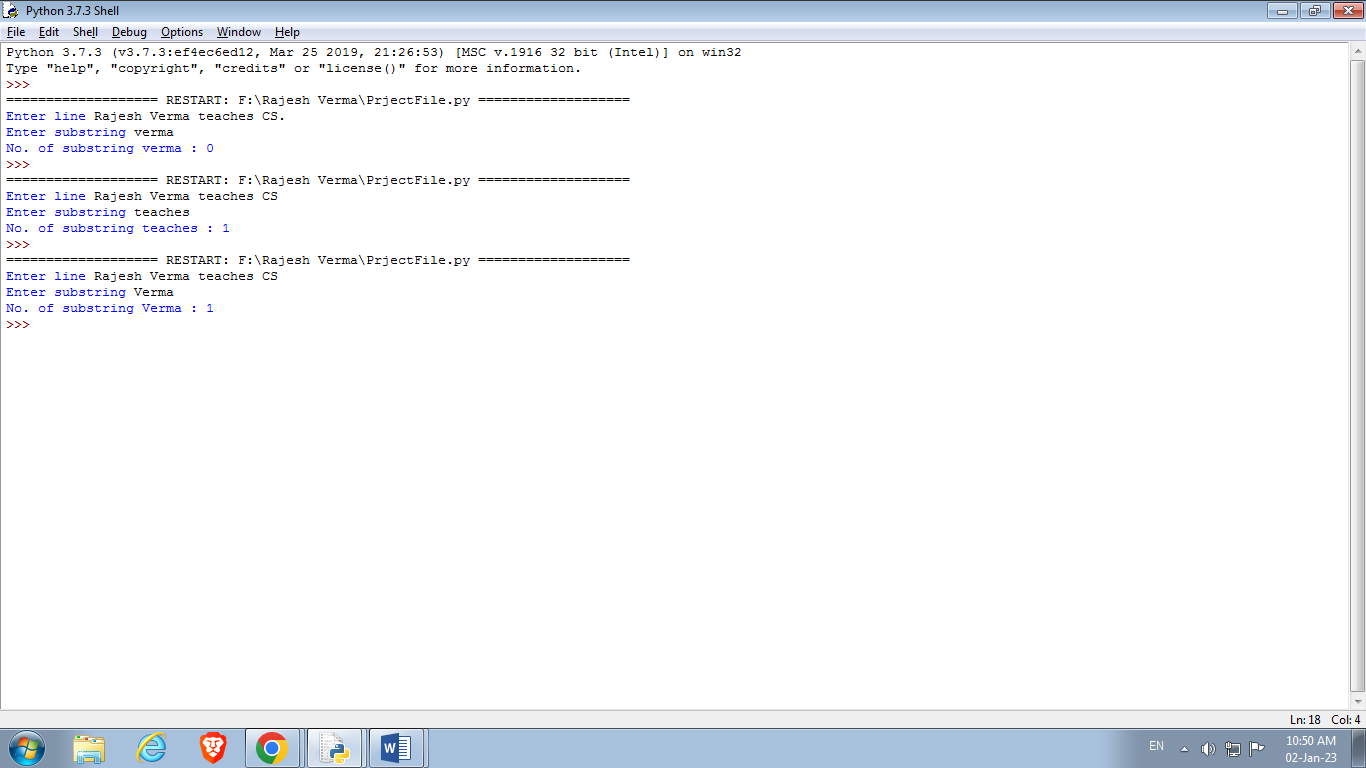
break

if start >= lenline :

break

print("No. of substring "+substr, ":",str(count))

**Output:**



**#2.Write a Python program to remove the nth index character from a nonempty string.**

str1 = input("Enter text ")

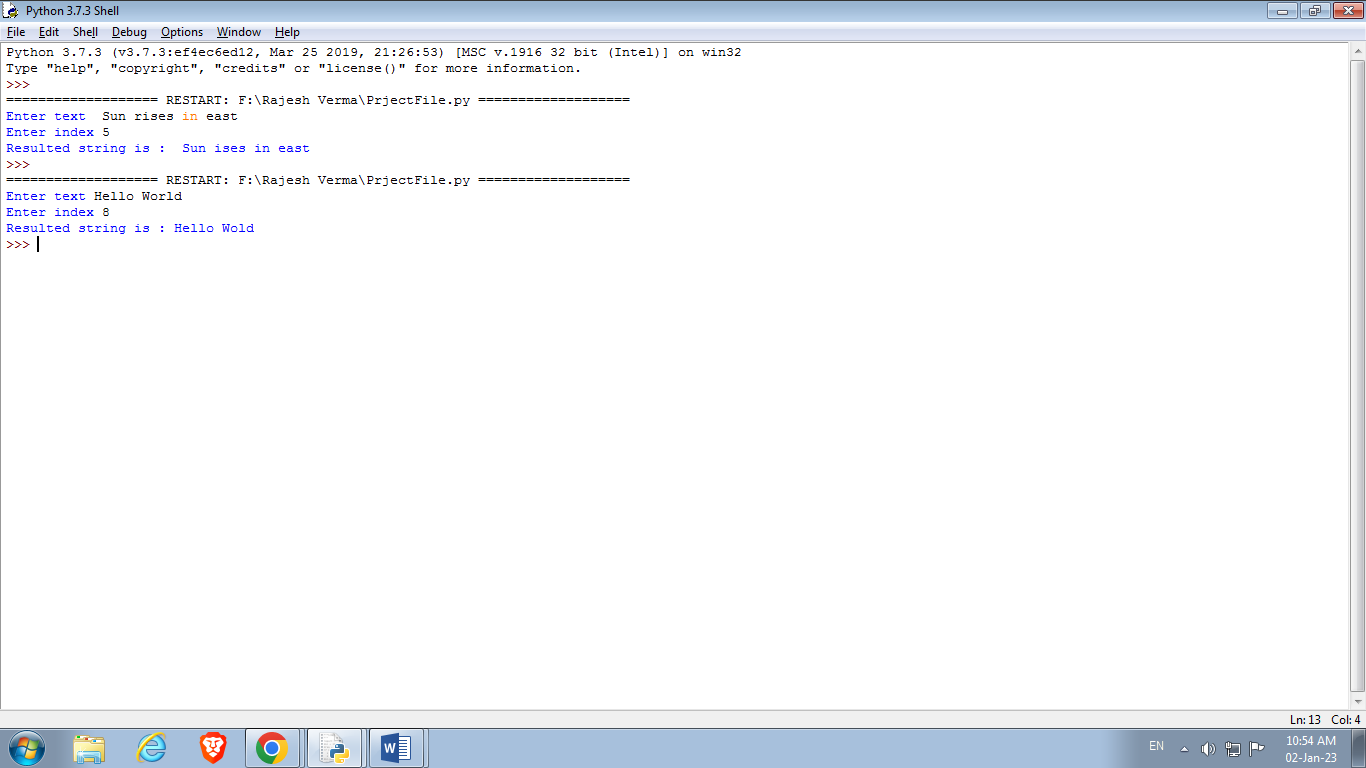
n = int(input("Enter index "))

first\_part = str1[:n]

last\_part = str1[n+1:]

print('Resulted string is : '+first\_part + last\_part)

**Output:**



**#3.Python program to find uncommon words from two Strings**

A=input('Enter first String ')

B=input('Enter Second String ')

A=A.split()

B=B.split()

x=[]

for i in A:

if i not in B:

x.append(i)

for i in B:

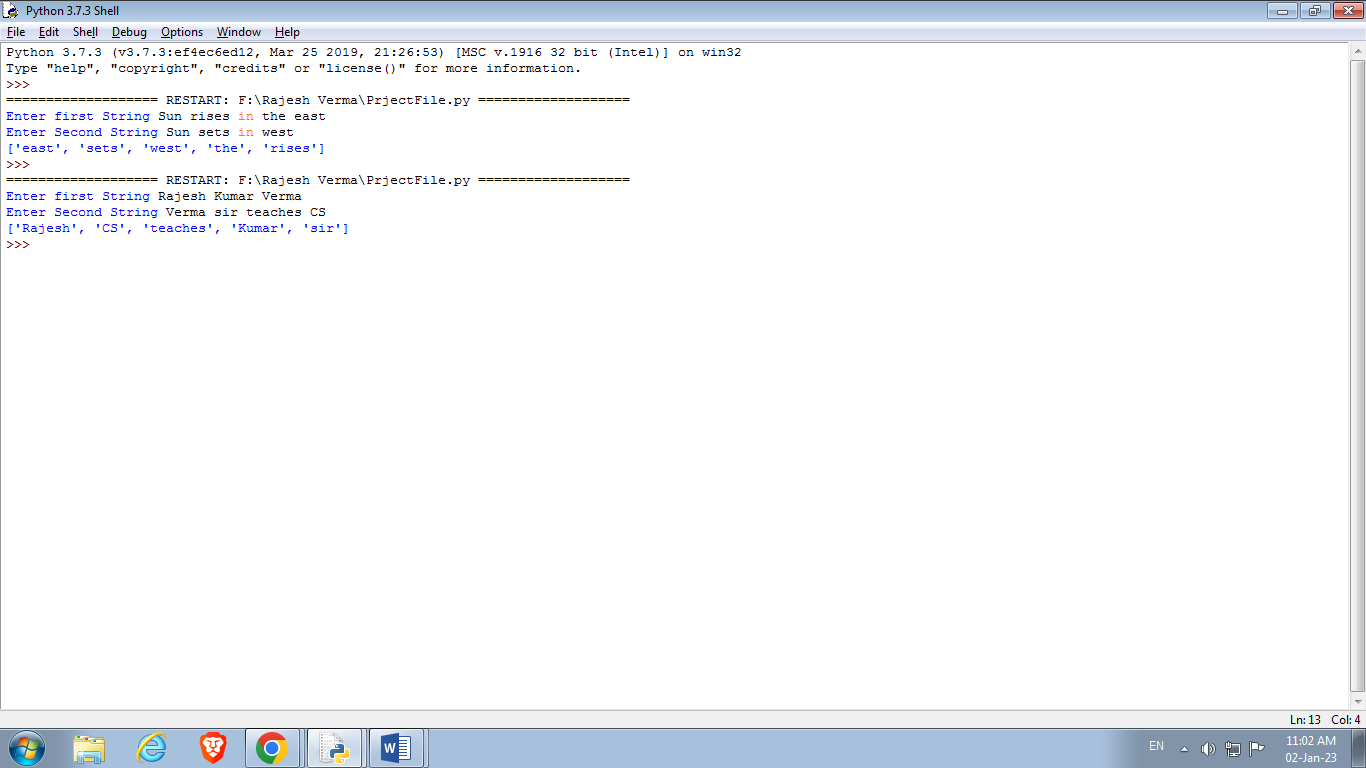
if i not in A:

x.append(i)

x=list(set(x))

print(x)

**Output:**



**#4.Reverse a string using stack(as list)**

def isEmpty( stk ):

if stk == []:

return True

else :

return False

def Push(stk,item) :

stk.append(item)

top = len(stk)-1

def Pop(stk) :

if isEmpty(stk):

return 'Underflow'

else :

item = stk.pop()

if len( stk ) == 0:

top =None

else :

top = len( stk )-1

return item

Stack=[]

str2=''

str1 = input('Enter string ')

for i in str1:

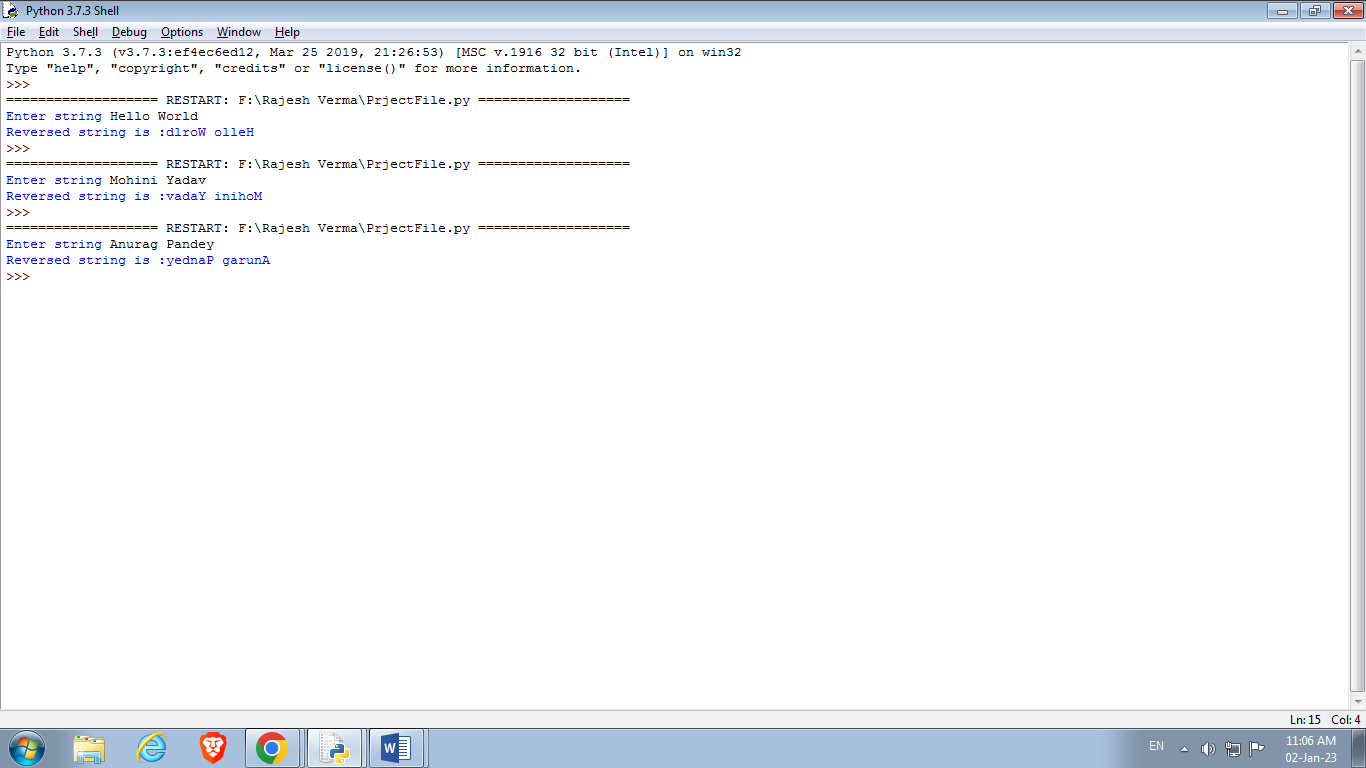
Push(Stack,i)

for j in range(len(Stack)):

str2+=Pop(Stack)

print('Reversed string is :'+str2)

**Output:**



**#5. Sum of series (1) + (1+2) +(1+2+3)+........upto N terms.**

def series1(n):

sum = sum1 = 0

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

sum = sum1 = 0

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

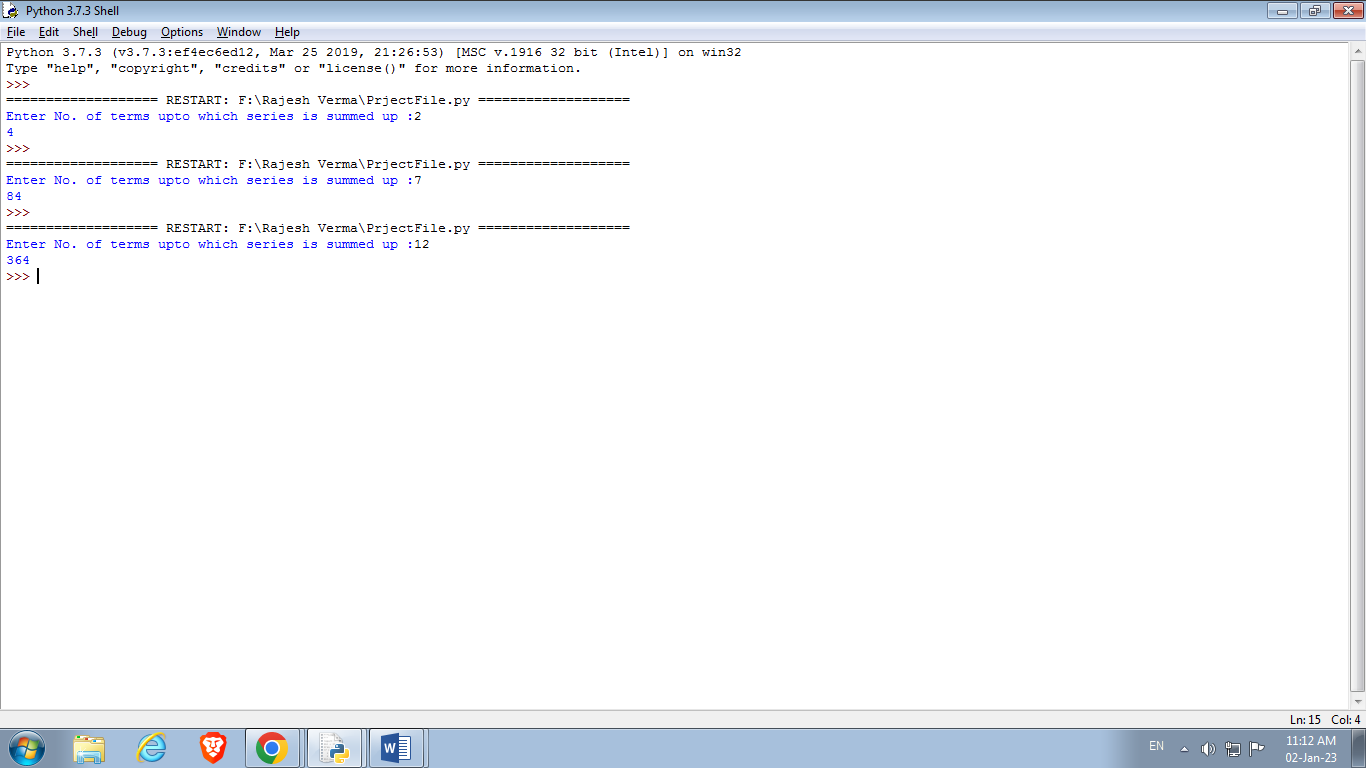
sum += j

sum1 += sum

return sum1

print(series1(int(input('Enter No. of terms upto which series is summed up :'))))

**Output :**



**# 6. Sum of series (1) + x/2! + x^2/4! + x^3/6!.upto N terms.**

def series2(x,n):

total = 1

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

num = i \* 2

fact = 1

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

fact = fact \* j

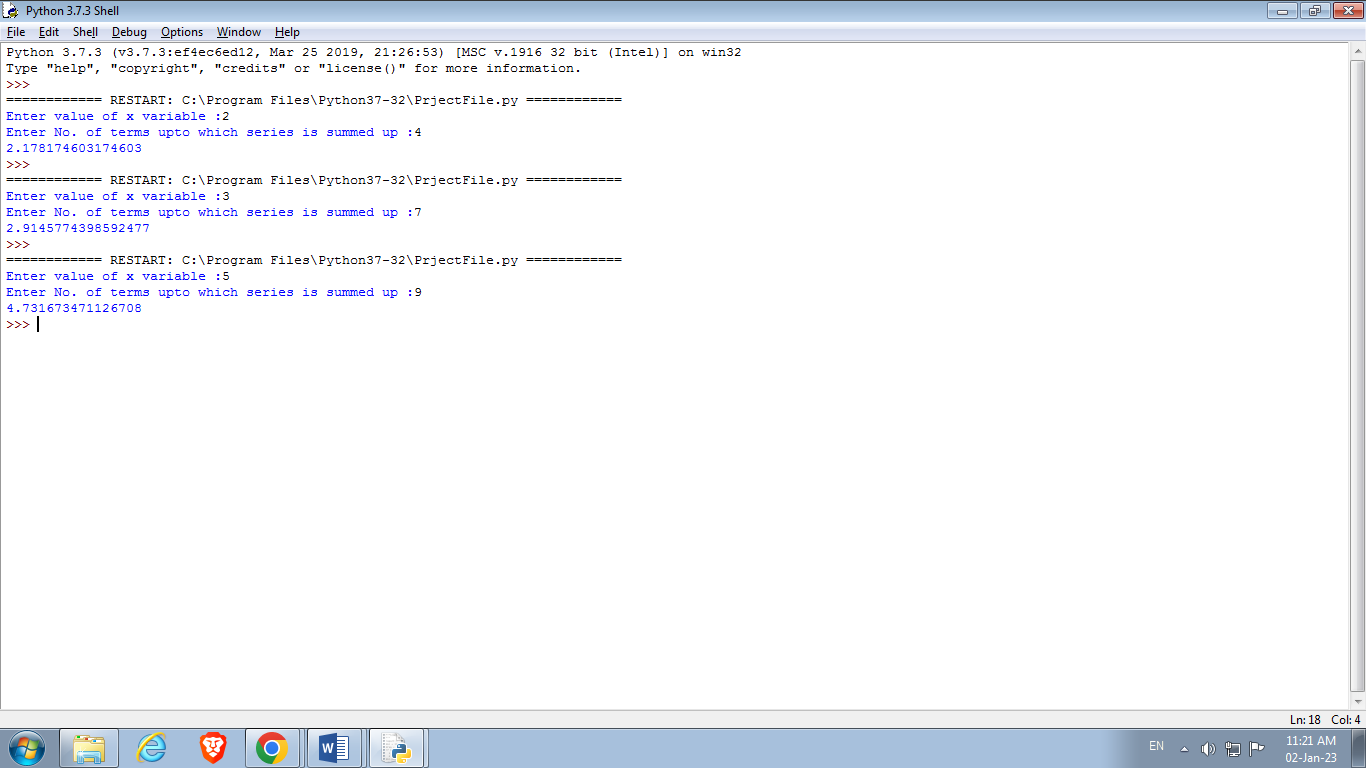
fraction = (x\*\*i)/fact

total = total + fraction

return total

print(series2(int(input('Enter value of x variable :')),int(input('Enter No. of terms upto which series is summed up :'))))

**Output:**



**# 7. Sum of series 1^2 + 2^2 + 3^2 +........upto N terms.**

def series3(n):

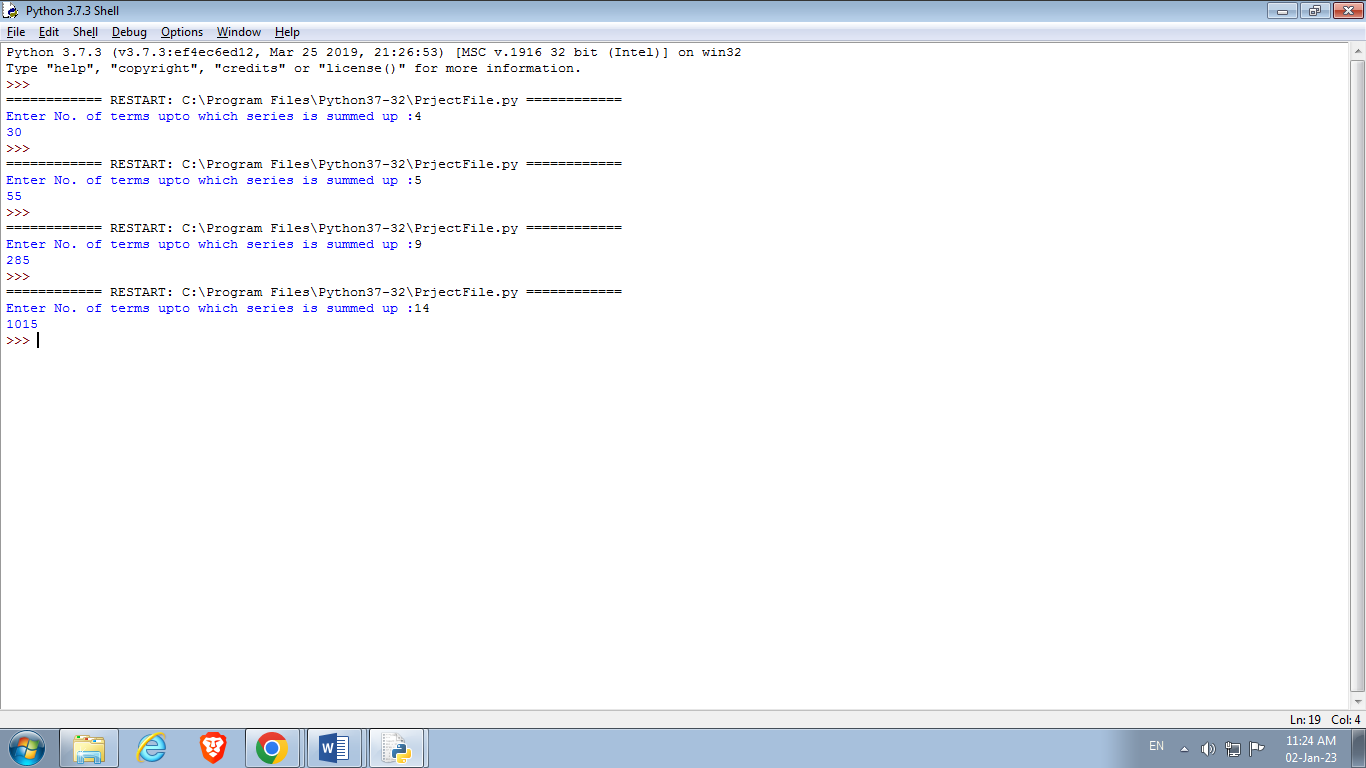
sum=0

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

sum += i\*\*2

return sum

print(series3(int(input('Enter No. of terms upto which series is summed up :'))))

**output:**

**#8.Implementation of insertion sort**

def Insertionsort(n):

alist=[]

for i in range(0, n):

if i==0:

ele = int(input('Enter ' +str(i+1) +'st element: '))

elif i==1:

ele = int(input('Enter ' +str(i+1) +'nd element: '))

elif i==2:

ele = int(input('Enter ' +str(i+1) +'rd element: '))

else :

ele = int(input('Enter ' +str(i+1) +'th element: '))

alist.append(ele) # adding the element

print('List before sorting :',alist)

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

key = alist[i]

j = i-1

while j>=0 and key < alist[j]:

alist[j+1] = alist[j]

j = j-1

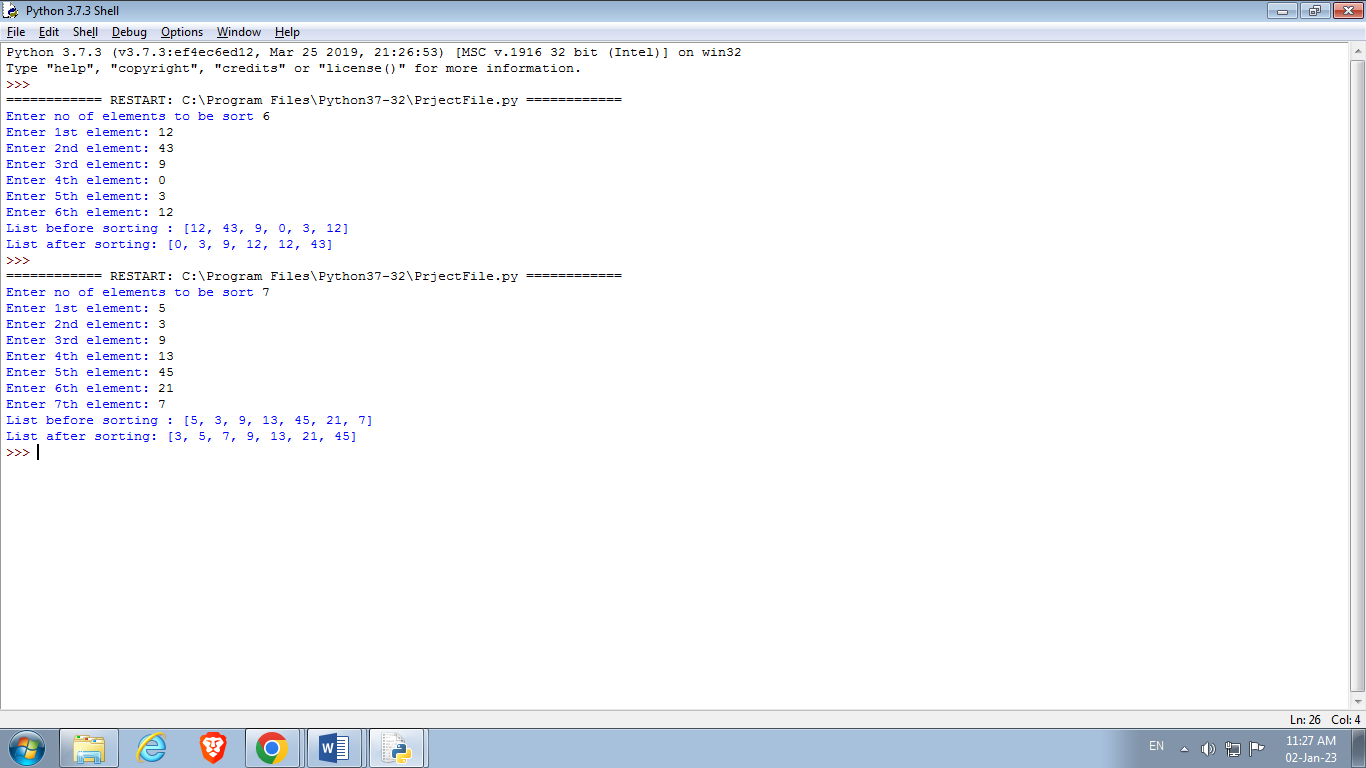
else:

alist[j+1]= key

print('List after sorting:',alist)

Insertionsort(int(input('Enter no of elements to be sort ')))

**Output:**



**#9.Implementation of bubble sort**

def bubblesort(n):

alist=[]

for i in range(0, n):

if i==0:

ele = int(input('Enter ' +str(i+1) +'st element: '))

elif i==1:

ele = int(input('Enter ' +str(i+1) +'nd element: '))

elif i==2:

ele = int(input('Enter ' +str(i+1) +'rd element: '))

else :

ele = int(input('Enter ' +str(i+1) +'th element: '))

alist.append(ele) # adding the element

print('List before sorting :',alist)

n=len(alist)

for i in range(n):

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

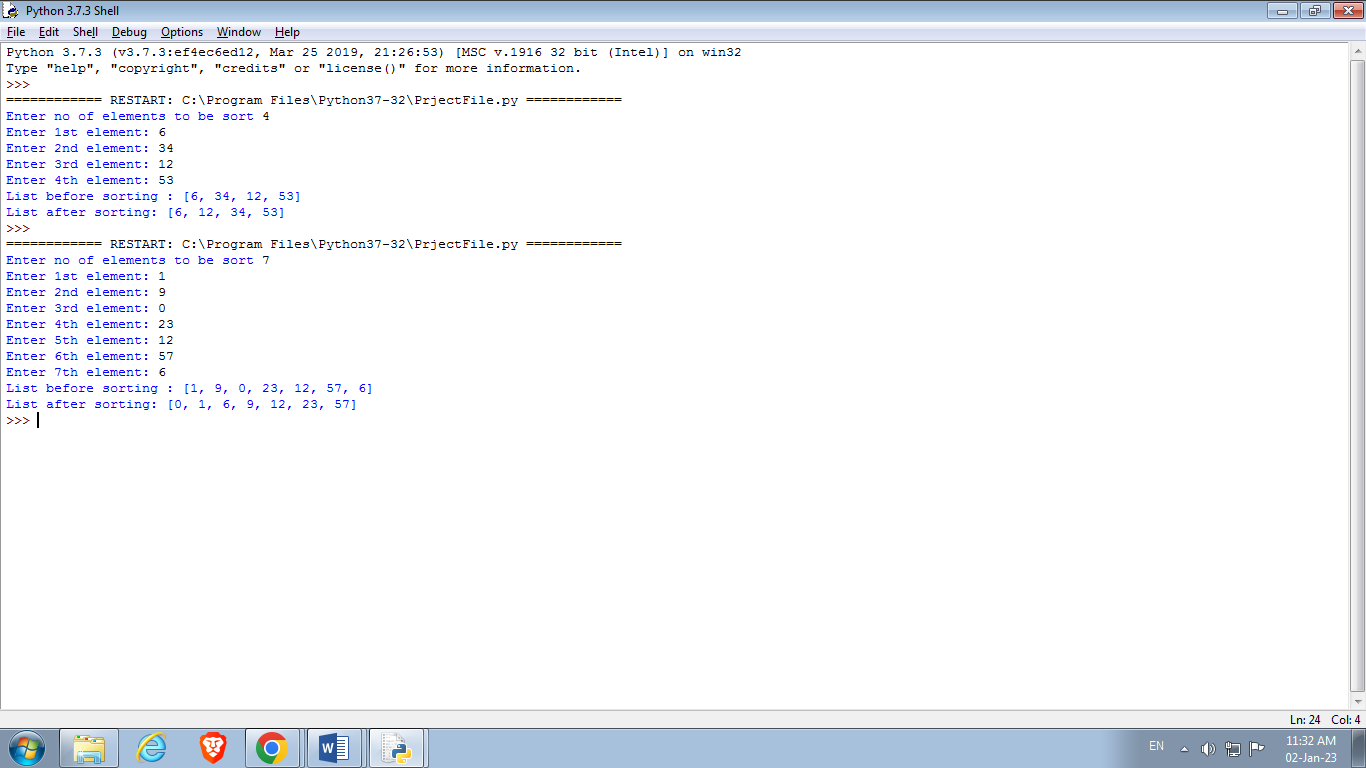
if alist[j]>alist[j+1]:

alist[j],alist[j+1]=alist[j+1],alist[j]

print('List after sorting:',alist)

bubblesort(int(input('Enter no of elements to be sort ')))

**Output:**



**10.Write a program to read the contents of both the files created in the above programs and merge the contents into “merge.txt”. Avoid using the close() function to close the files.**

def mergeFiles():

with open("MyFile.txt","r") as f1:

data=f1.read()

with open("intro.txt","r") as f2:

data1=f2.read()

with open("merge.txt","w") as f3:

f3.write(data)

f3.write(data1)

with open("merge.txt","w") as f4:

print(f4.read())

mergeFiles()

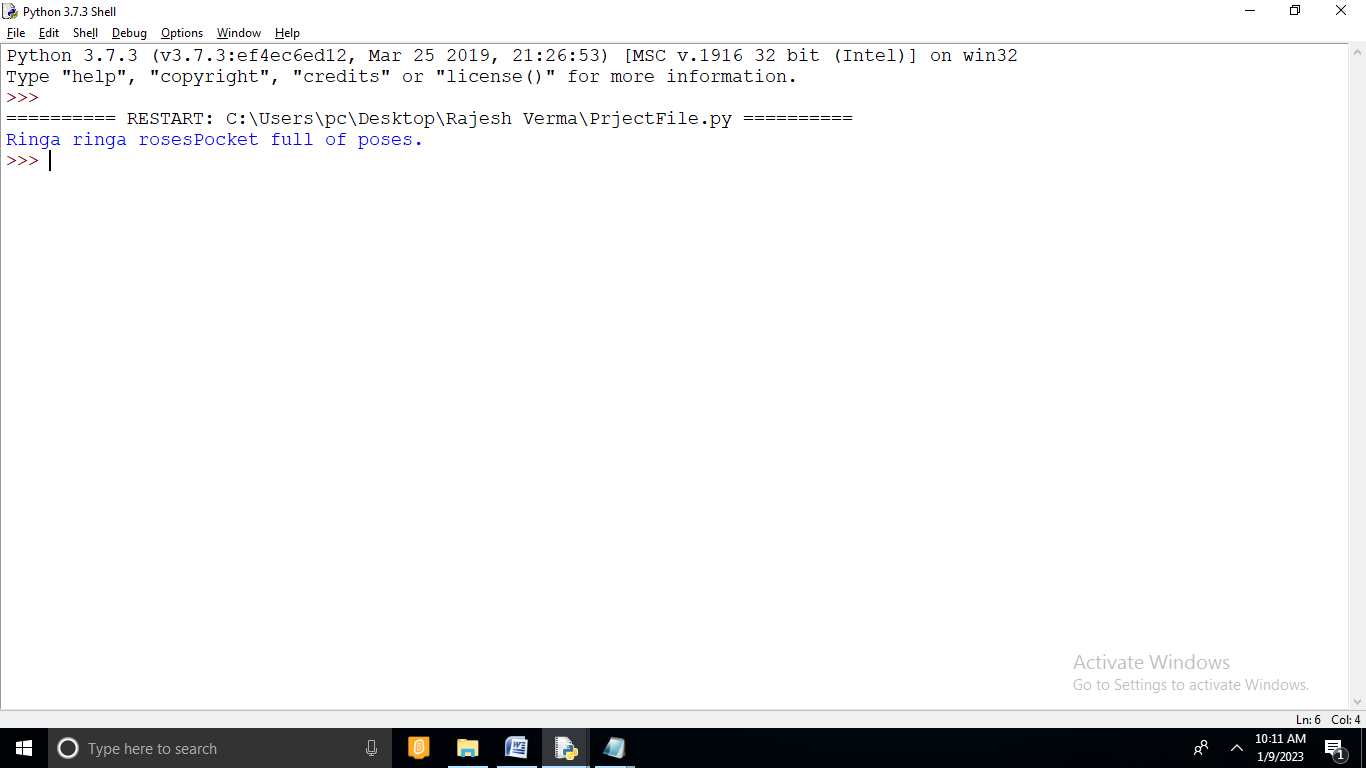
**Note: Text in Myfile.txt**

Ringa ringa roses

**Text in intro.txt**

Pocket full of poses.

**Output in merge.txt file:**



**11. Write a program to count a total number of lines and count the total number of lines starting with ‘A’, ‘B’, and ‘C’. (Consider the merge.txt file)**

def checkChar():

with open(r"merge.txt","r") as f1:

data=f1.readlines()

cnt\_lines=0

cnt\_A=0

cnt\_B=0

cnt\_C=0

for lines in data:

cnt\_lines+=1

if lines[0]=='A':

cnt\_A+=1

if lines[0]=='B':

cnt\_B+=1

if lines[0]=='C':

cnt\_C+=1

print("Total Number of lines are:",cnt\_lines)

print("Total Number of lines strating with A are:",cnt\_A)

print("Total Number of lines strating with B are:",cnt\_B)

print("Total Number of lines strating with C are:",cnt\_C)

checkChar()

**Note: Content of Merge.txt**

Baa, baa black sheep

Have you any wool

Yes sir, yes sir

Three bags full.

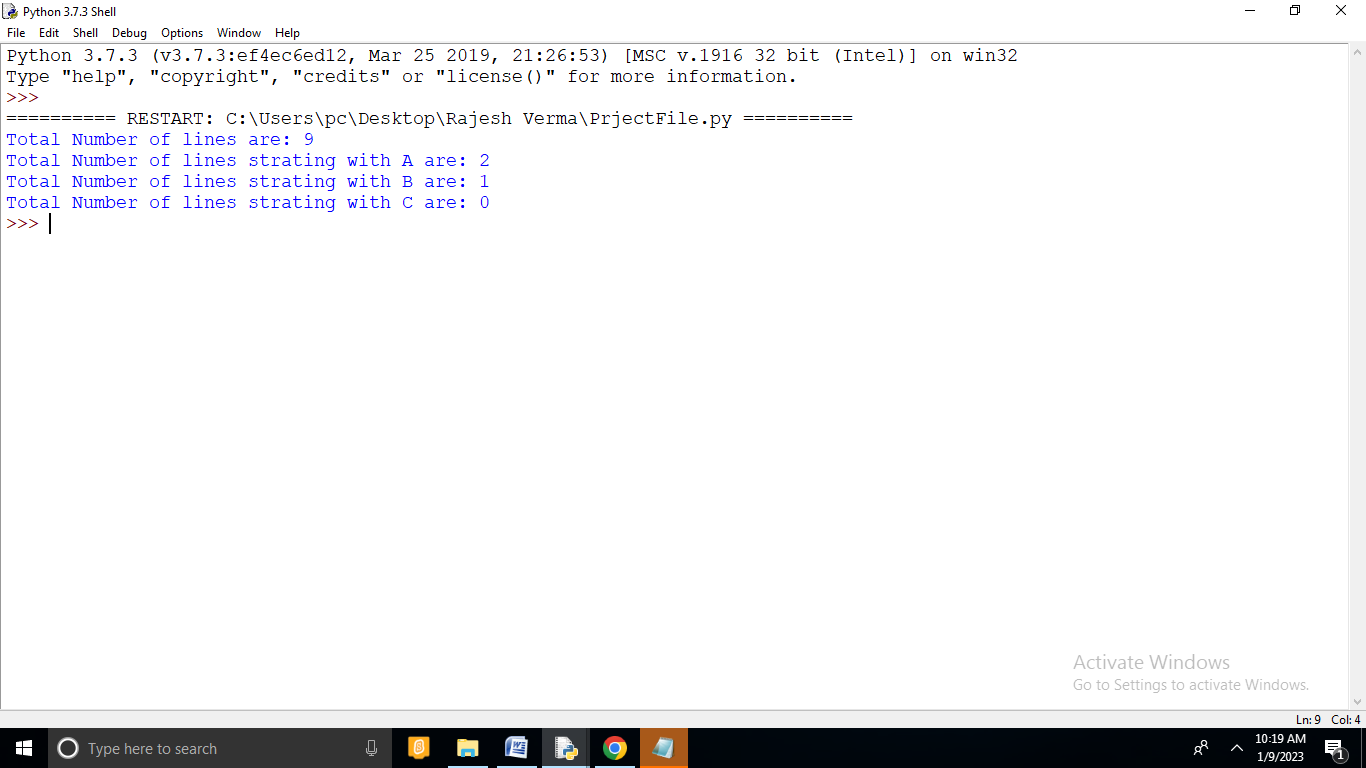
One for my master

And one for my dame

And one for the little boy

Who lives down the lane.

**Output:**

****

**#12 : count the uppercase characters in a file**

upperCount =0

F=open(r"merge.txt","r")

while(True):

data=F.read(1)

if(data==""):

break

if (ord(data) >= 65 and ord(data) <= 90):

upperCount = upperCount +1

print(data,end='')

print("Total Upper Case:",upperCount)

F.close()

**Note: Content of Merge.txt**

Baa, baa black sheep

Have you any wool

Yes sir, yes sir

Three bags full.

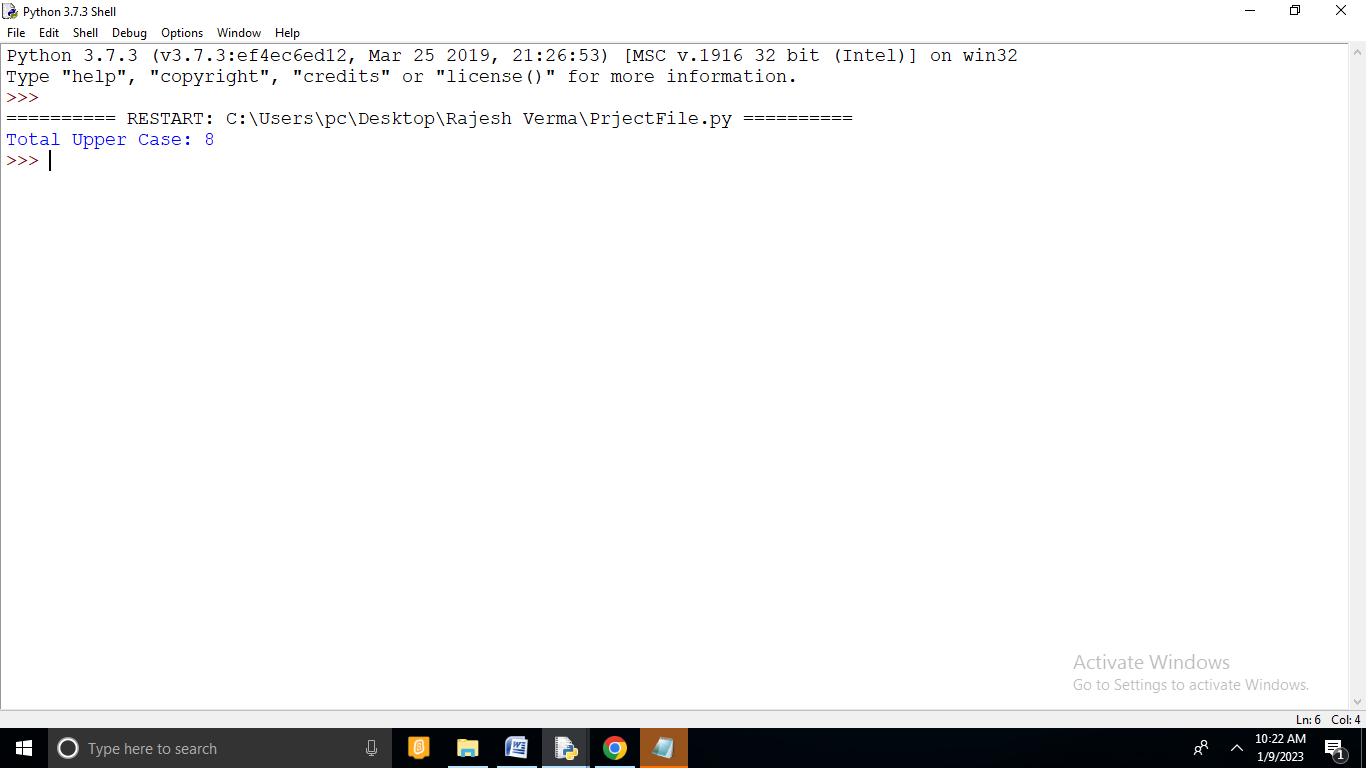
One for my master

And one for my dame

And one for the little boy

Who lives down the lane.

**Output:**

****

**#13.How to create a table in MySQL from Python. Let’s create table ‘Laptop’ under the ‘Electronics’ database.**

import mysql

import mysql.connector

try:

connection = mysql.connector.connect(host='localhost',database='Electronics', user='root',password='1234')

mySql\_Create\_Table\_Query = """CREATE TABLE Laptop (

Id int(11) NOT NULL,

Name varchar(250) NOT NULL,

Price float NOT NULL,

Purchase\_date Date NOT NULL,

PRIMARY KEY (Id)) """

cursor = connection.cursor()

result = cursor.execute(mySql\_Create\_Table\_Query)

print("Laptop Table created successfully ")

except mysql.connector.Error as error:

print("Failed to create table in MySQL: {}".format(error))

finally:

if connection.is\_connected():

cursor.close()

connection.close()

print("MySQL connection is closed")

**Output:**

Laptop Table created successfully. MySQL connection is closed.

## #14. Insert multiple rows into MySQL table.

import mysql.connector

try:

connection = mysql.connector.connect(host='localhost',

database='Electronics',

user='root',

password='1234')

mySql\_insert\_query = """INSERT INTO Laptop (Id, Name, Price, Purchase\_date)

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

records\_to\_insert = [(4, 'HP Pavilion Power', 1999, '2019-01-11'),

(5, 'MSI WS75 9TL-496', 5799, '2019-02-27'),

(6, 'Microsoft Surface', 2330, '2019-07-23')]

cursor = connection.cursor()

cursor.executemany(mySql\_insert\_query, records\_to\_insert)

connection.commit()

print(cursor.rowcount, "Record inserted successfully into Laptop table")

except mysql.connector.Error as error:

print("Failed to insert record into MySQL table {}".format(error))

finally:

if connection.is\_connected():

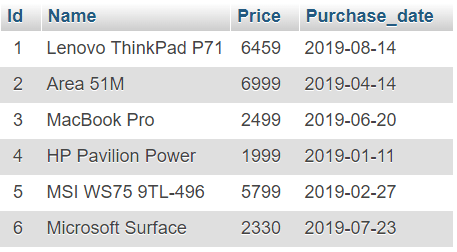
cursor.close()

connection.close()

print("MySQL connection is closed")

**Output:**

3 Record inserted successfully into Laptop table MySQL connection is closed



MySQL Laptop table after inserting multiple rows

**#15. Update mySQL table records using python.**

import mysql.connector

try:

connection = mysql.connector.connect(host='localhost',

database='electronics',

user='root',

password='1234’)

cursor = connection.cursor()

print("Before updating a record ")

sql\_select\_query = """select \* from Laptop where id = 1"""

cursor.execute(sql\_select\_query)

record = cursor.fetchone()

print(record)

# Update single record now

sql\_update\_query = """Update Laptop set Price = 7000 where id = 1"""

cursor.execute(sql\_update\_query)

connection.commit()

print("Record Updated successfully ")

print("After updating record ")

cursor.execute(sql\_select\_query)

record = cursor.fetchone()

print(record)

except mysql.connector.Error as error:

print("Failed to update table record: {}".format(error))

finally:

if connection.is\_connected():

connection.close()

print("MySQL connection is closed")

**Output:**

Before updating a row

(1, 'Lenovo ThinkPad P71', 6459.0, datetime.date(2019, 8, 14))

Record Updated successfully

After updating row

(1, 'Lenovo ThinkPad P71', 7000.0, datetime.date(2019, 8, 14))

MySQL connection is closed

