**INTRODUCTION**

**BRIEF ON RELEVANCE OF PYTHON:**

Python is a general purpose and high level programming language. You can use Python for

developing desktop GUI applications, websites and web applications.The simple syntax rules of the

programming language further makes it easier for you to keep the code base readable and

application maintainable. There are also a number of reasons why you should prefer Python

to other programming languages.

● Python is used in machine learning & artificial intelligence, fields at the cutting-edge

of tech

●

Python is simply structured and easy to learn.

● Python programming is versatile in terms of platform and purpose.

● Many Open Source Frameworks and Tools.

● python has broad standard library.

● Readable and Maintainable Code.

● Compatible with Major Platforms and Systems.

● it can be used for web development,networking games,data proccessing,business

application.

.

.

**HOW TO INSTALL AND EXECUTE PYTHON:**

**Step 1: Select Version of Python to Install :**

The installation procedure involves downloading the official Python .exe installer and

running it on your system.

The version you need depends on what you want to do in Python.

**Step 2: Download Python Executable Installer :**

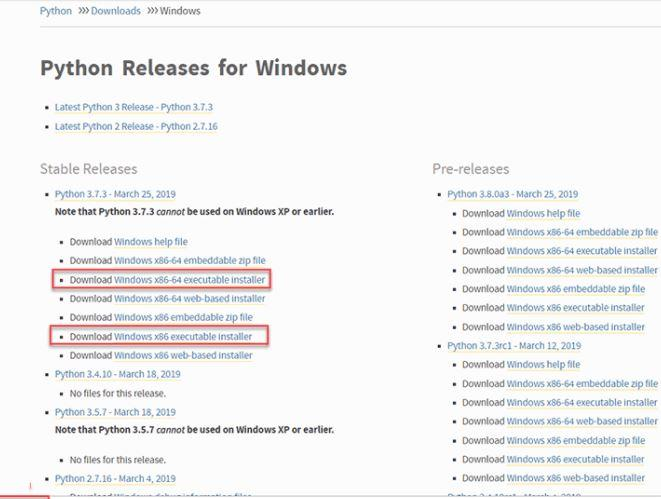
1. Open your web browser and navigate to the Downloads for Windows section of the

official Python website.

2. Search for your desired version of Python.

3. Select a link to download either the Windows x86-64 executable installer or Windows

x86 executable installer. The download is approximately 25MB.



**Step 3: Run Executable Installer :**

Run the Python Installer once downloaded.

2. Make sure you select the Install launcher for all users and Add Python 3.7 to PATH

checkboxes. The latter places the interpreter in the execution path. For older versions of

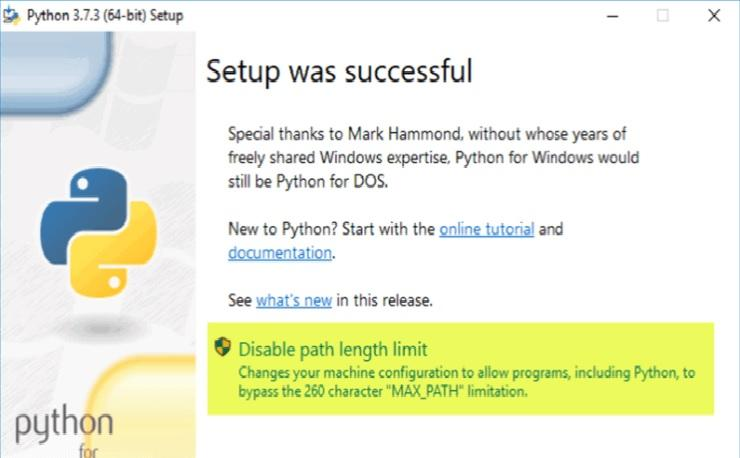
Python that do not support the Add Python to Path checkbox

3. Select Install Now – the recommended installation options.



For all recent versions of Python, the recommended installation options include Pip and

IDLE



**Step 4: Verify Python Was Installed On Windows :**

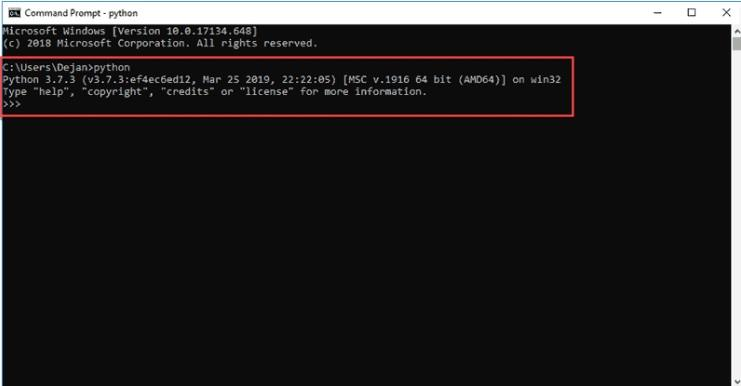
1. Navigate to the directory in which Python was installed on the system. In our case, it

is C:\Users\Username\AppData\Local\Programs\Python\Python37 since we have

installed the latest version.

2. Double-click python.exe.

3. The output should be similar to what you can see below:



**Step 5: Verify Pip Was Installed :**

Pip is a powerful package management system for Python software packages. Thus, make

sure that you have it installed.

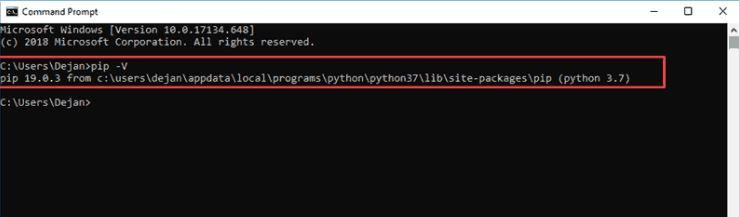
To verify whether Pip was installed:

1. Open the Start menu and type "cmd."

2. Select the Command Prompt application.

3. Enter pip -V in the console. If Pip was installed successfully, you should see the

following outp



Pip has not been installed yet if you get the following output:

'pip' is not recognized as an internal or external command,

Operable program or batch file.

**DATA TYPE AND OPERATORS IN PYTHON PROGRAMMING :**

**Data types** :The classification of data items or to put the data value into some sort of data

category is called Data Types.

1. Python Numbers :

● Integers:

There is no maximum limit on the value of an integer. The integer can be of any length

without any limitation which can go up to the maximum available memory of the system.

eg:0,1,2,3....

● Floating Point Number:

The difference between floating points and integers is decimal points. Floating point number

can be represented as “1.0”, and integer can be represented as “1”. It is accurate up to 15

decimal places.

● Complex Number:

“x + yj” is the written form of the complex number. Here y is the imaginary part and x is the

real part.

2. Python List :

An ordered sequence of items is called List

eg: a = [5,9.9,’list’]

3. Python Tuple :

A Tuple is a sequence of items that are in order, and it is not possible to modify the Tuples.

eg: t = (6,’tuple’,4+2r)

4. Python Strings :

A String is a sequence of Unicode characters.

eg:s = “Python String”

5. Python Set :

The Collection of Unique items that are not in order is called Set. Braces {} are used to

defined set and a comma is used to separate values.

eg:>>> a = {4,5,5,6,6,6}

>>> a

{4, 5, 6}

6. Python Dictionary :

Dictionary is a type of python data type in which collections are unordered, and values are in

pairs called key-value pairs. This type of data type is useful when there is a high volume of

data.

eg:>>> d = {3:’key’,4:’value’}

7. Boolean Type :

There can be only two types of value in the Boolean data type of Python, and that is True or

False.

It can look like this:

>>> type(True)

<class ‘bool’>

>>> type(False)

<class ‘bool’>

The true value in the Boolean context is called “truthy”, and for false value in the Boolean

context, it is called “falsy”. Truthy is defined by the objects in boolean, which is equal to True,

and in the same way, Falsy is defined by the objects equal to falsy. One can also evaluate

Non-Boolean objects in a Boolean context.

**python operators :**

Operators are used to perform operations on variables and values.

● Arithmetic operators

● Assignment operators

● Comparison operators

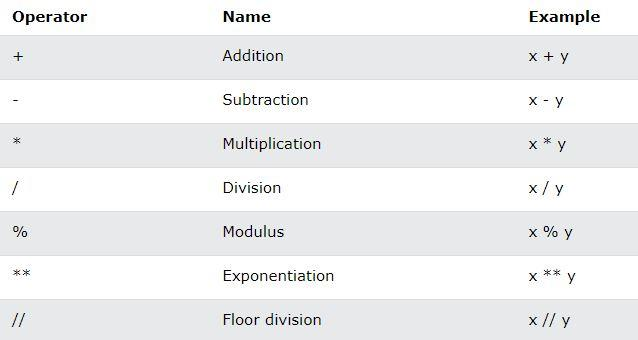
● Logical operators

● Identity operators

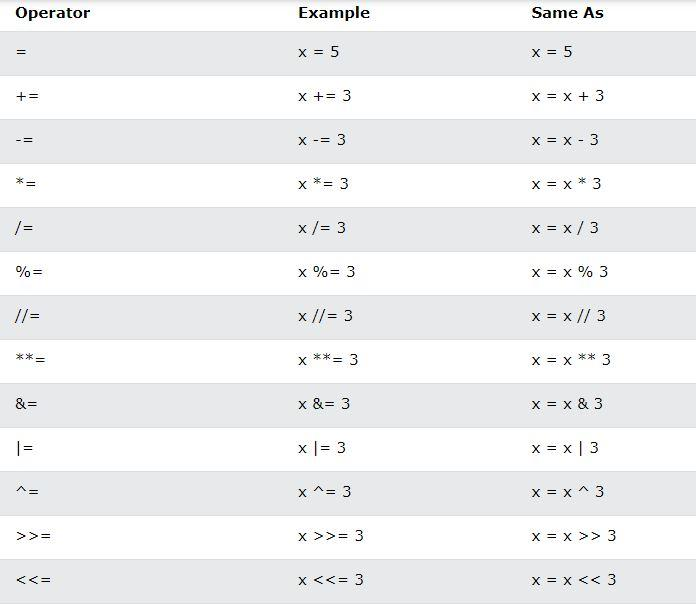
● Membership operators

● Bitwise operators

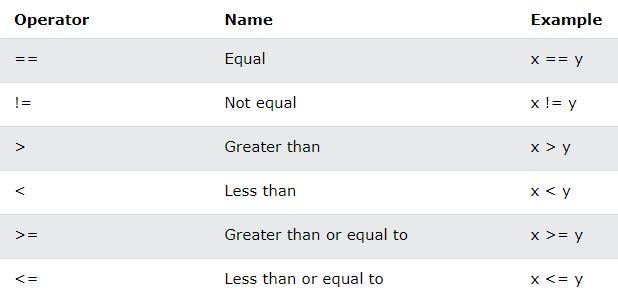
Arithmetic operators :



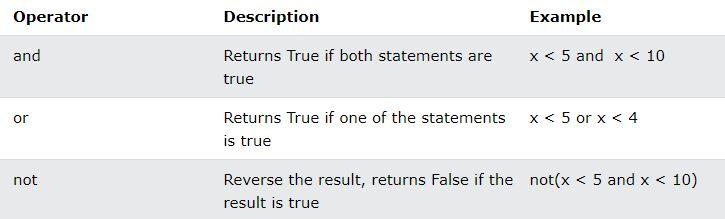
Assignment Operators :



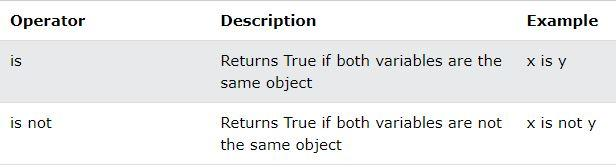
Comparison Operators :



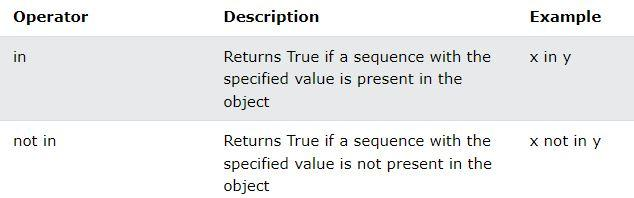
Logical Operators :



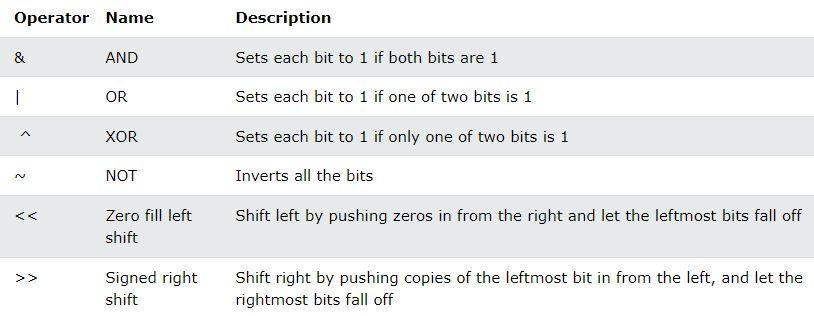
Identity Operators :



Membership Operators :

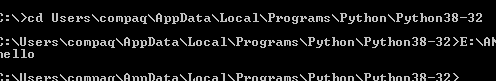


Bitwise Operators:



1.print Hello

 print("Hello")



2.Program to add two numbers by taking default values.

a=5

print("first number is:")

print(a)

b=3

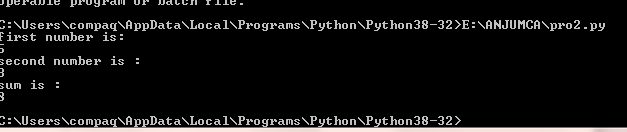
print("second number is :")

print(b)

sum=a+b

print("sum is :")

print(sum)



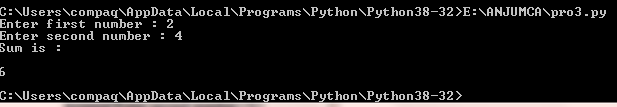
3. Program to add two numbers by user interactive input.

a=input('Enter first number : ')

b=input('Enter second number : ')

sum=int(a)+int(b)

print("Sum is :\n") print(sum)



4.Program to swap two numbers using temporary variable.

x=input("Enter the value for x :")

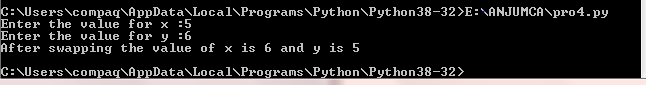
y=input("Enter the value for y :")

temp=x

x=y

y=temp

print("After swapping the value of x is {} and y is {}".format(x,y))



5. Program to swap to numbers without temporary variable.

a=input("Enter first number : ")

b=input("Enter second number : ")

sum=int(a)+int(b)

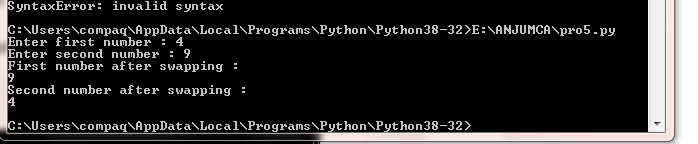
a=int(sum)-int(a)

print("First number after swapping : ")

print(a) b=int(sum)-int(b)

print("Second number after swapping : ")

print(b)



6.Program to compute the roots of quadratic equation.

import cmath

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

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

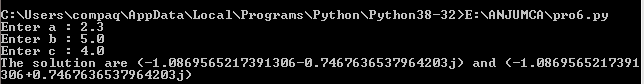
c=float(input("Enter c : "))

d=(b\*\*2)-(4\*a\*c)

sol1=(-b-cmath.sqrt(d))/(2\*a)

sol2=(-b+cmath.sqrt(d))/(2\*a)

print('The solution are {0} and {1}'. format(sol1,sol2))



7.Program to compute the area of a triangle

a=float(input("Enter first side : "))

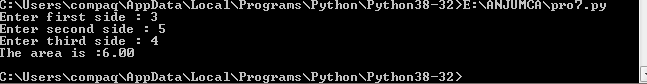
b=float(input("Enter second side : "))

c=float(input("Enter third side : "))

s=(a+b+c)/2

area=(s\*(s-a)\*(s-b)\*(s-c))\*\*0.5

print("The area is :%0.2f"%area)



8.Program to check leap year or not.

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

if(year%4==0):

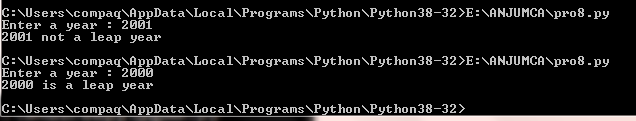
if(year%100==0):

if(year%400==0):

print("{0} is a leap year".format(year))

else:

print("{0} not a leap year".format(year))



**9.Simulate calculator**

print("1.Addition")

print("2.Substraction")

print("3.Multiplication")

print("4.Division")

choice=int(input("Select choice"))

if choice==1:

a=float(input("Enter first num"))

b=float(input("Enter second num"))

c=a+b

print("sum is=",c)

elif choice==2:

a=float(input("Enter first num"))

b=float(input("Enter second num"))

c=a-b

print("difference is=",c)

elif choice==3:

a=float(input("Enter first num"))

b=float(input("Enter second num"))

c=a\*b

print("product is=",c)

elif choice==4:

a=float(input("Enter first num"))

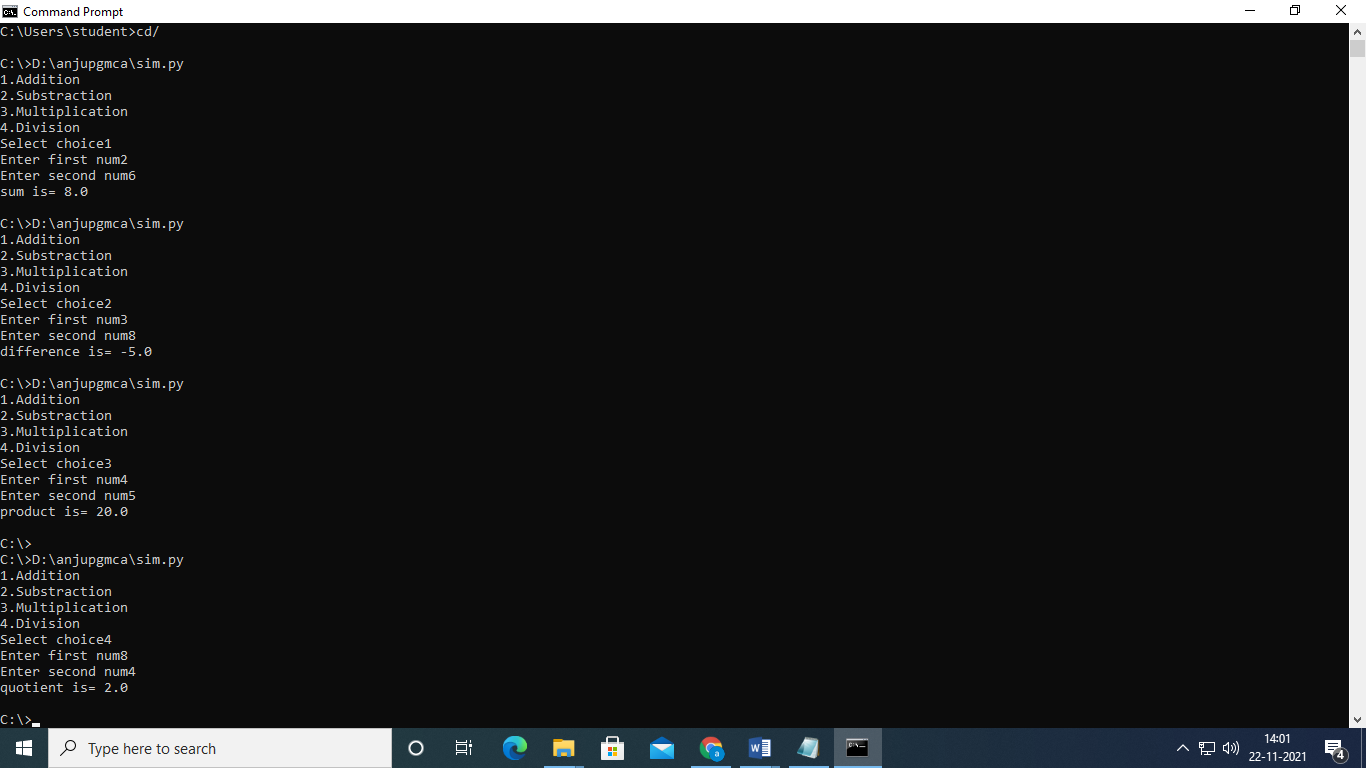
b=float(input("Enter second num"))

c=a/b

print("quotient is=",c)

else:

print("Wrong choice")



 10. PROGRAM TO CHECK WHETHER ARMSTRONG OR NOT

n=int(input("Enter the number"))

temp1=n

temp2=n

s=0

while(n!=0):

       s=s+1

       n=n//10

print("no of digits is ",s)

res=0

while(temp2!=0):

      r=temp2%10

      res=res+r\*\*s

      temp2=temp2//10

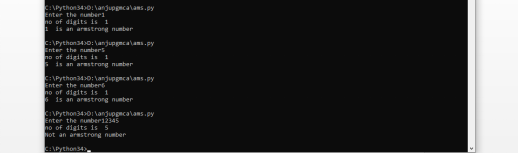
if(res==temp1):

        print(res," is an armstrong number")

else:

        print("Not an armstrong number")

output



 11.PROGRAM TO DISPLAY MULTIPLICATION TABLE

num=int(input("MULTIPLCATION TABLE OF:"))

for i in range(1,11):

         print(num,"\*",i,"=",num\*i)

output



  12.  PROGRAM TO CHECK WHETHER PERFECT NUMBER OR NOT

n=int(input("enter a number"))

sum=0

for i in range(1,n):

         if(n%i==0):

                print(i,"is a factor of",n)

                sum=sum+i

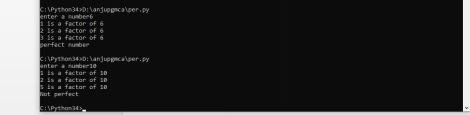
if(sum==n):

      print("perfect number")

else:

      print("Not perfect")

output



     13.  PROGRAM TO CHECK WHETHER PRIME OR NOT

num=int(input("enter a number"))

#prime number are greater than 1

if(num>1):

      for i in range(2,num):

               if(num%i==0):

                     print("not prime\n")

                     print(i,"times",num//i,"is",num)

                     break

#if break is not used else below will also be executed

      else:

                       print("it is a prime number")

output



    14.  PROGRAM TO PRINT PRIME IN AN INTERVAL

   low=int(input("Enter LOWER RANGE:"))

high=int(input("Enter HIGHER RANGE:"))

for num in range(low,high+1):

        if(num>1):

               for i in range(2,num):

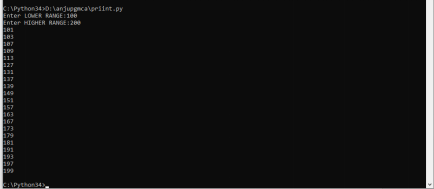
                       if((num%i)==0):

                           break

               else:

                           print(num)

output



15. PROGRAM TO CONVERT A BINARY  NUMBER TO DECIMAL

def conv(n):

       s=0

       i=0

       while(n>0):

             r=n%10

             s=s+r\*(2\*\*i)

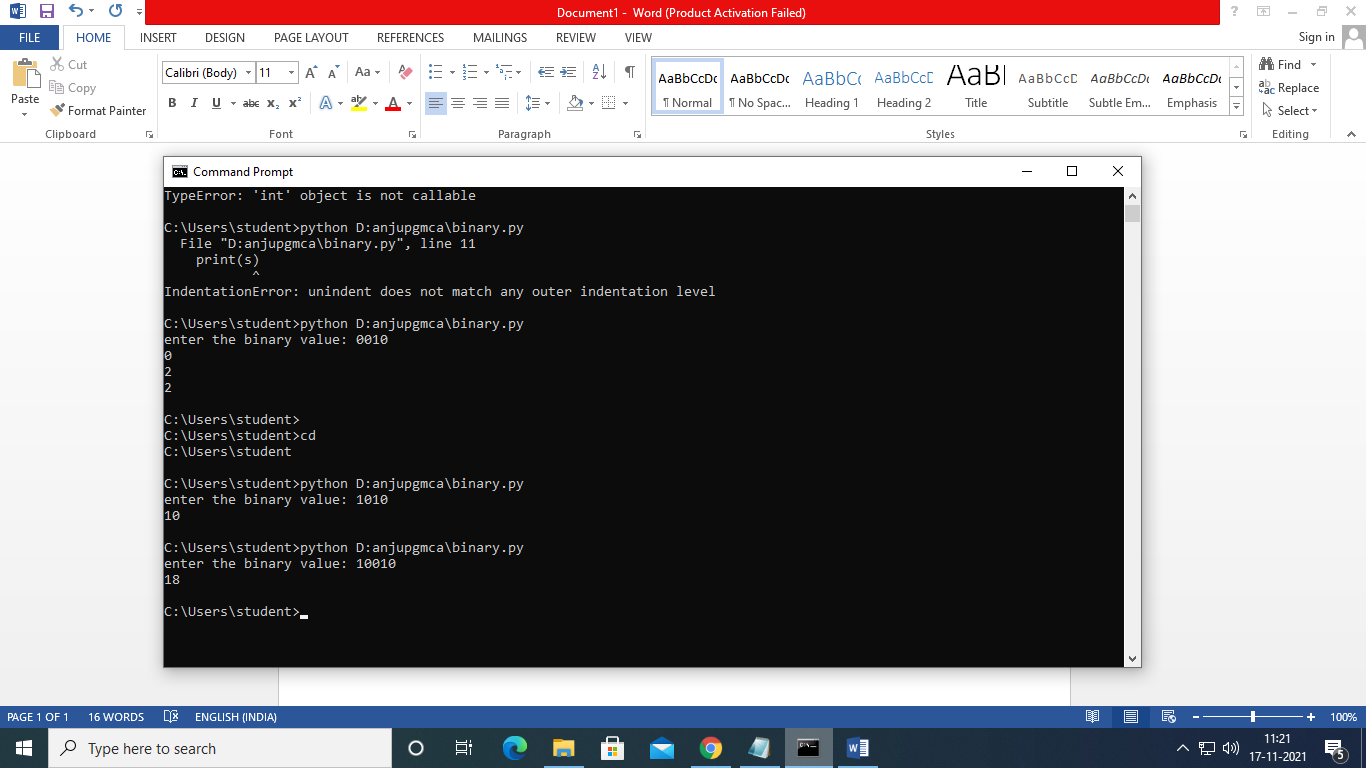
             i=i+1

             n=n//10

       print(s)

f=int(input("enter the binary value: "))

conv(f)



**16.PROGRAM TO FIND SUM TILL A NUMBER USING FUNCTION**

def sumn(num):

sum=0

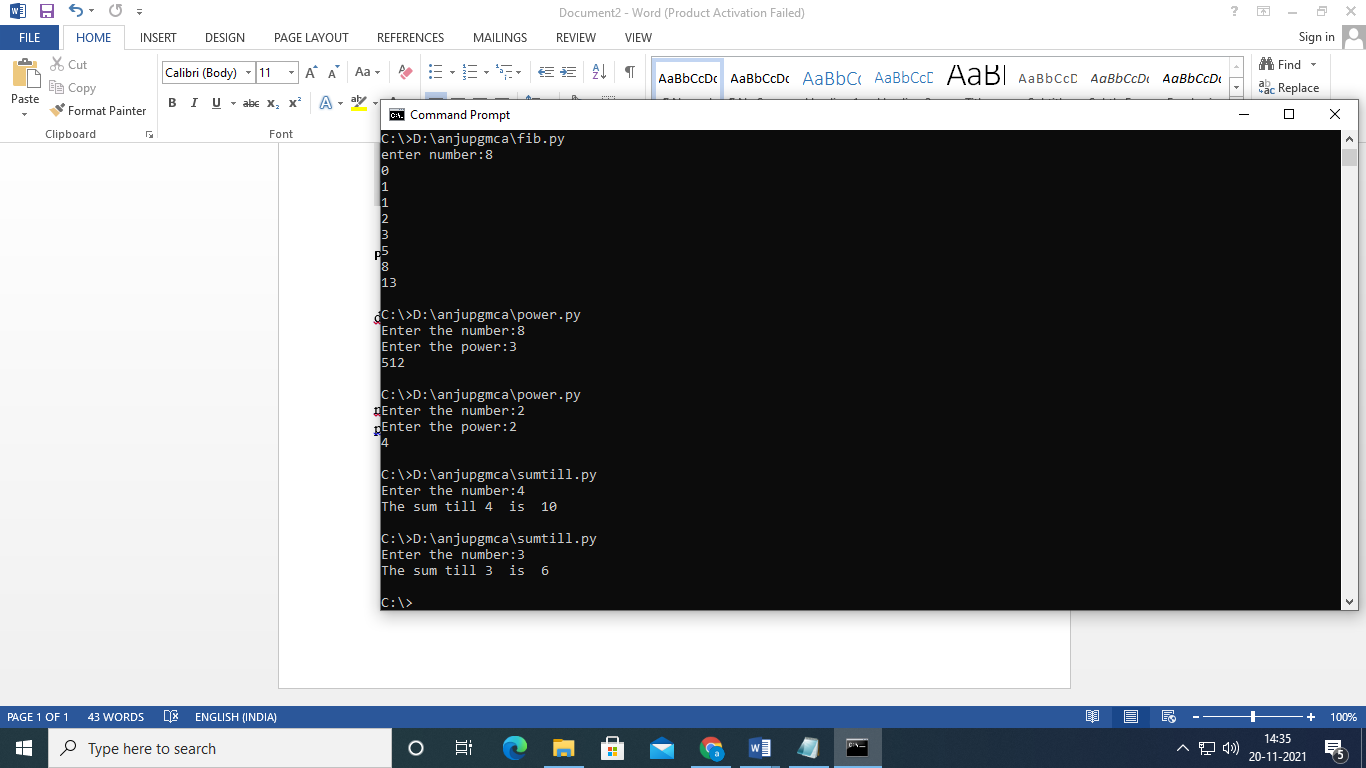
for i in range(1,num+1):

  sum=sum+i

return sum

num=int(input("Enter the number:"))

print("The sum till",num," is ",sumn(num))



**17.PROGRAM TO FIND FIBONACCI OF A NUMBER USING FUNCTION**

**def fibo(num):**

**a=0**

**b=1**

**for i in range(2,num):**

**c=a+b**

**print(c)**

**a=b**

**b=c**

**num=int(input("Enter the number:"))**

**if(num==1):**

**print(0)**

**elif(num==2):**

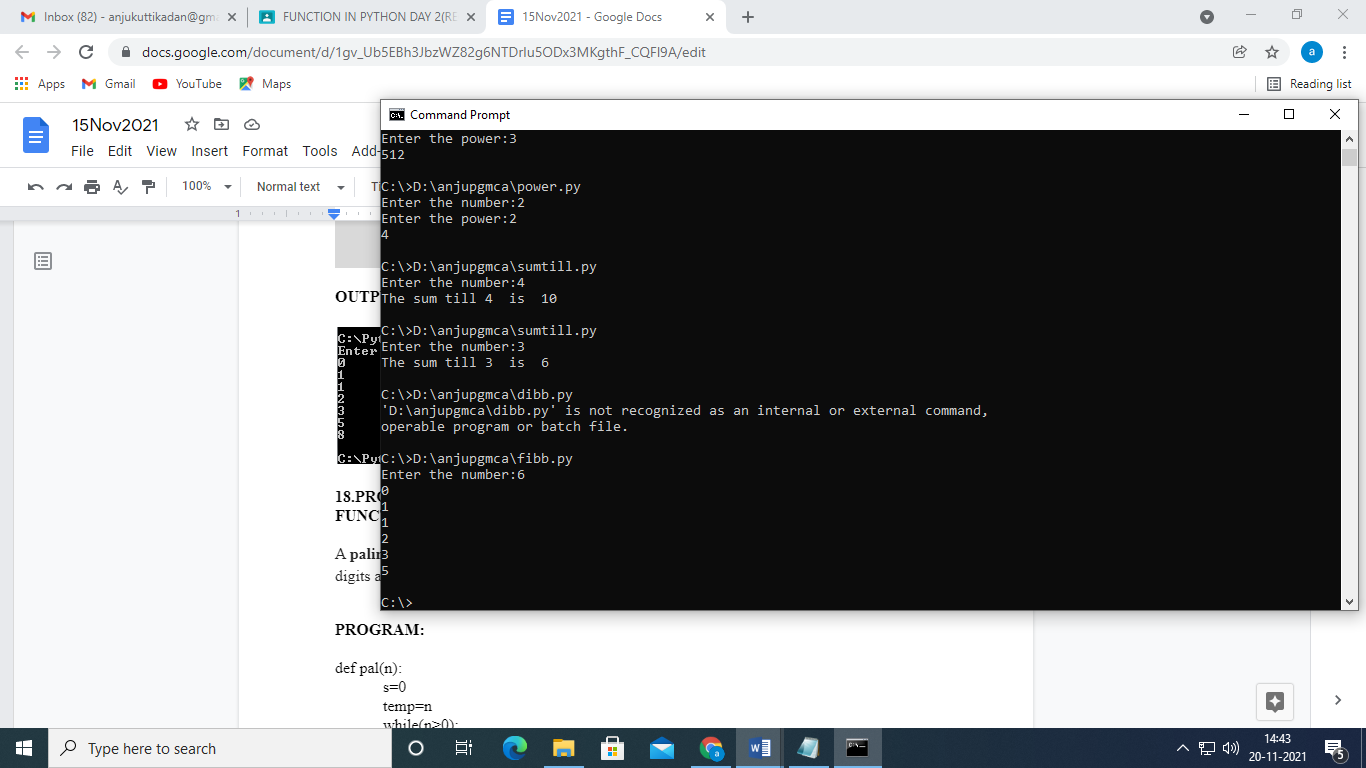
**print(0,1)**

**else:**

**print(0)**

**print(1)**

**fibo(num)**



**18.PROGRAM TO FIND R A NUMBER IS PALINDROME OR NOT USING FUNCTION**

def pal(n):

s=0

temp=n

while(n>0):

r=n%10

s=s\*10+r

n=n//10

print("The reverse is",s)

if(temp==s):

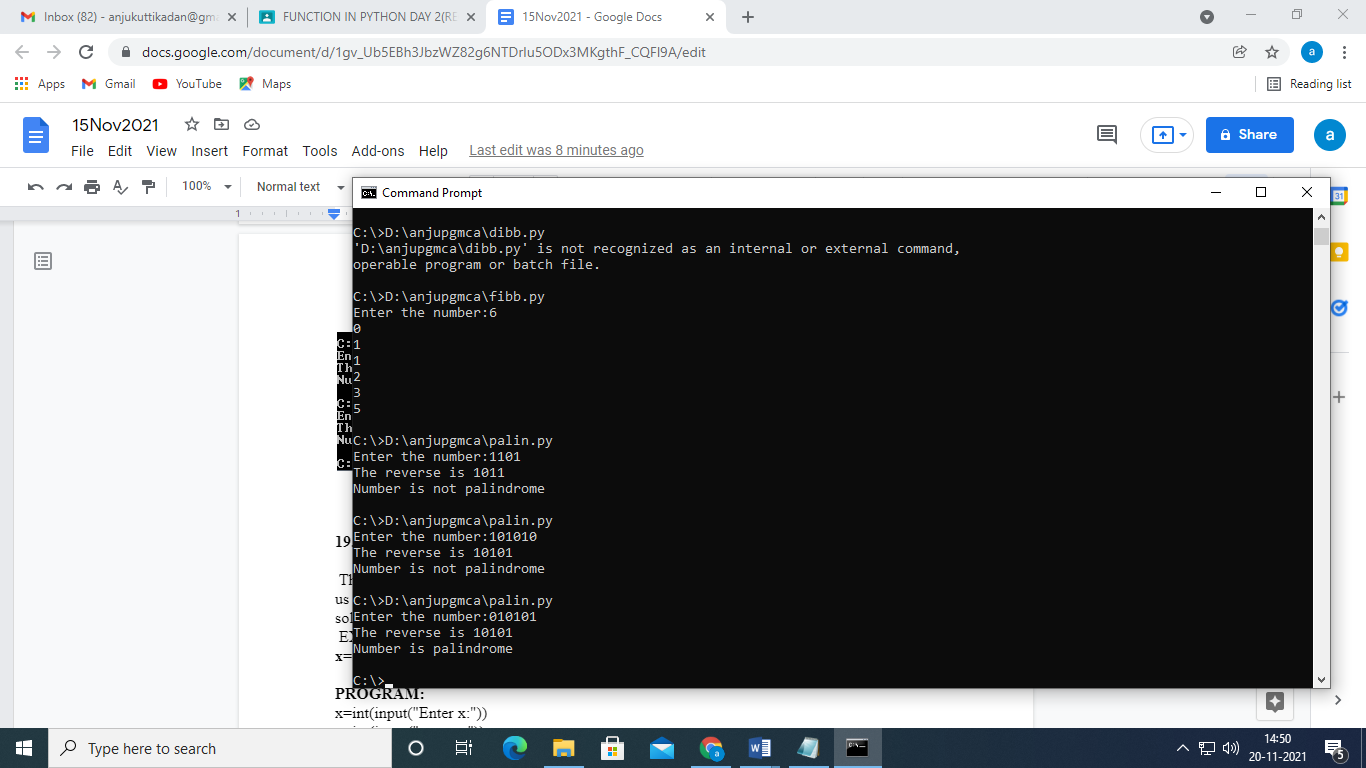
print("Number is palindrome")

else:

print("Number is not palindrome")

f=int(input("Enter the number:"))

pal(f)



**19.PROGRAM TO COMPUTE EXPONENTIAL SERIES**

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

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

s=1

def fac(n):

f=1

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

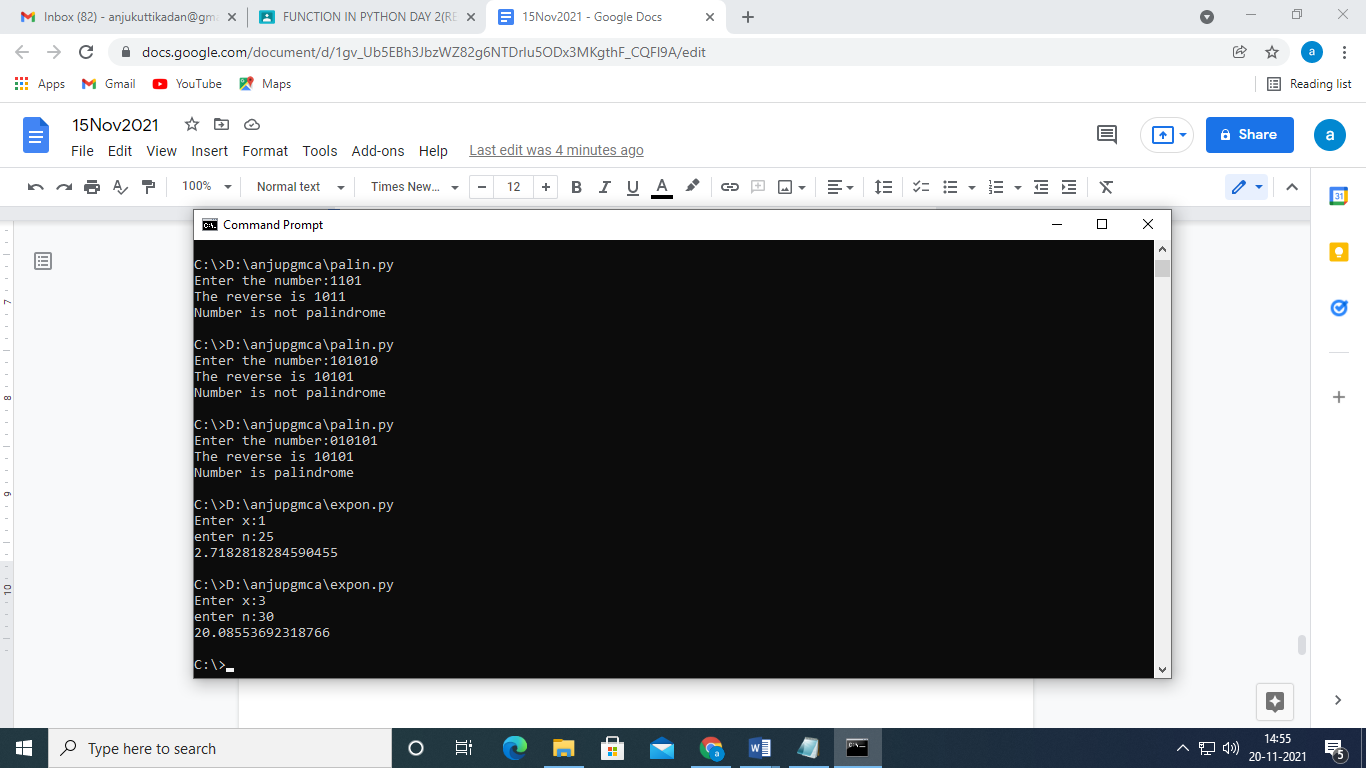
f=f\*i

return(f)

for i in range(1,n):

s=s+(x\*\*i)/(fac(i))

print(s)



**20. PROGRAM TO IMPLEMENT A SIMPLE CALCULATOR USING FUNCTIONS**

def add(x, y):

   """This function adds two numbers"""

   return x + y

def subtract(x, y):

   """This function subtracts two numbers"""

   return x - y

def multiply(x, y):

   """This function multiplies two numbers"""

   return x \* y

def divide(x, y):

   """This function divides two numbers"""

   return x / y

def pow(x, y):

   """This function computes power of the number"""

   return x \*\*y

def mod(x, y):

   """This function computes modulus operator"""

   return x % y

# take input from the user

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

print("5.Power")

print("6.Modulus")

choice = input("Enter choice(1/2/3/4/5/6):")

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

if choice == '1':

   print(num1,"+",num2,"=", add(num1,num2))

elif choice == '2':

   print(num1,"-",num2,"=", subtract(num1,num2))

elif choice == '3':

   print(num1,"\*",num2,"=", multiply(num1,num2))

elif choice == '4':

   print(num1,"/",num2,"=", divide(num1,num2))

elif choice == '5':

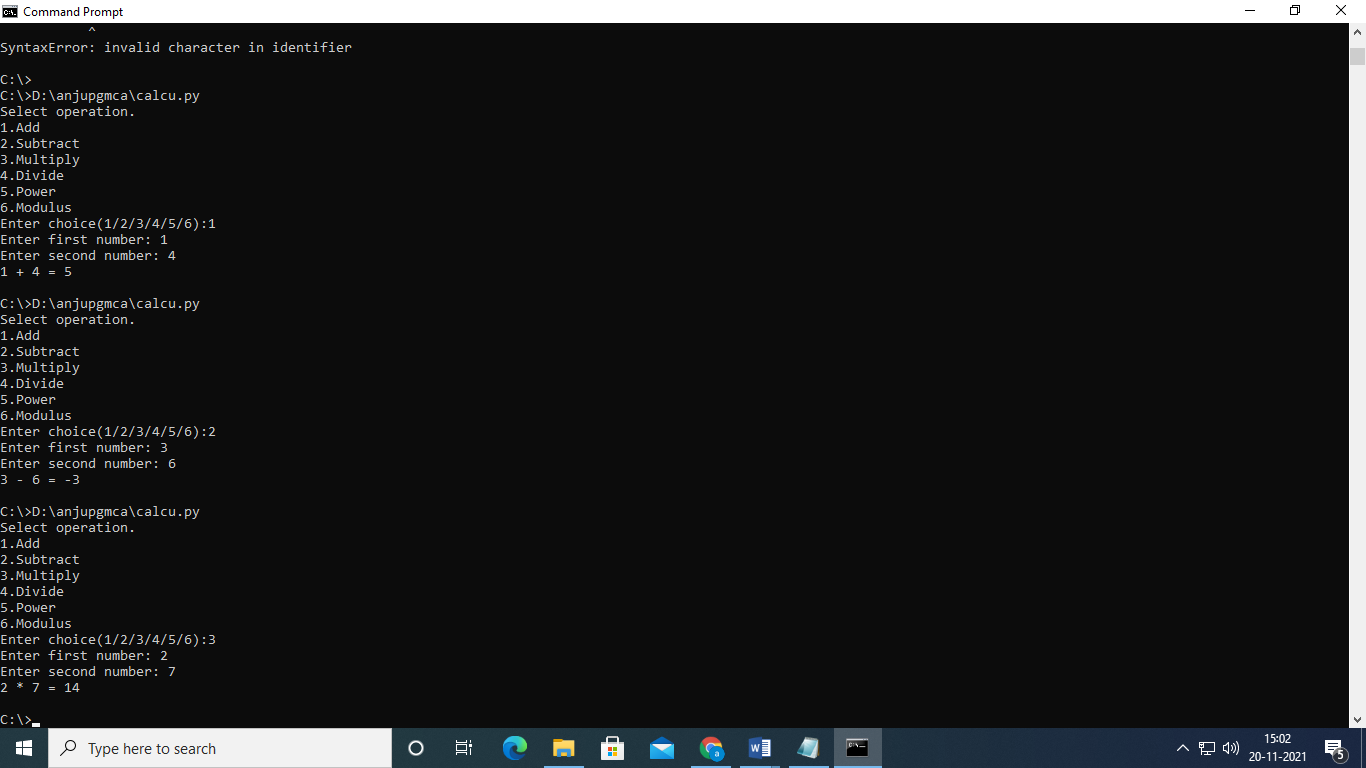
   print(num1,"^",num2,"=", pow(num1,num2))

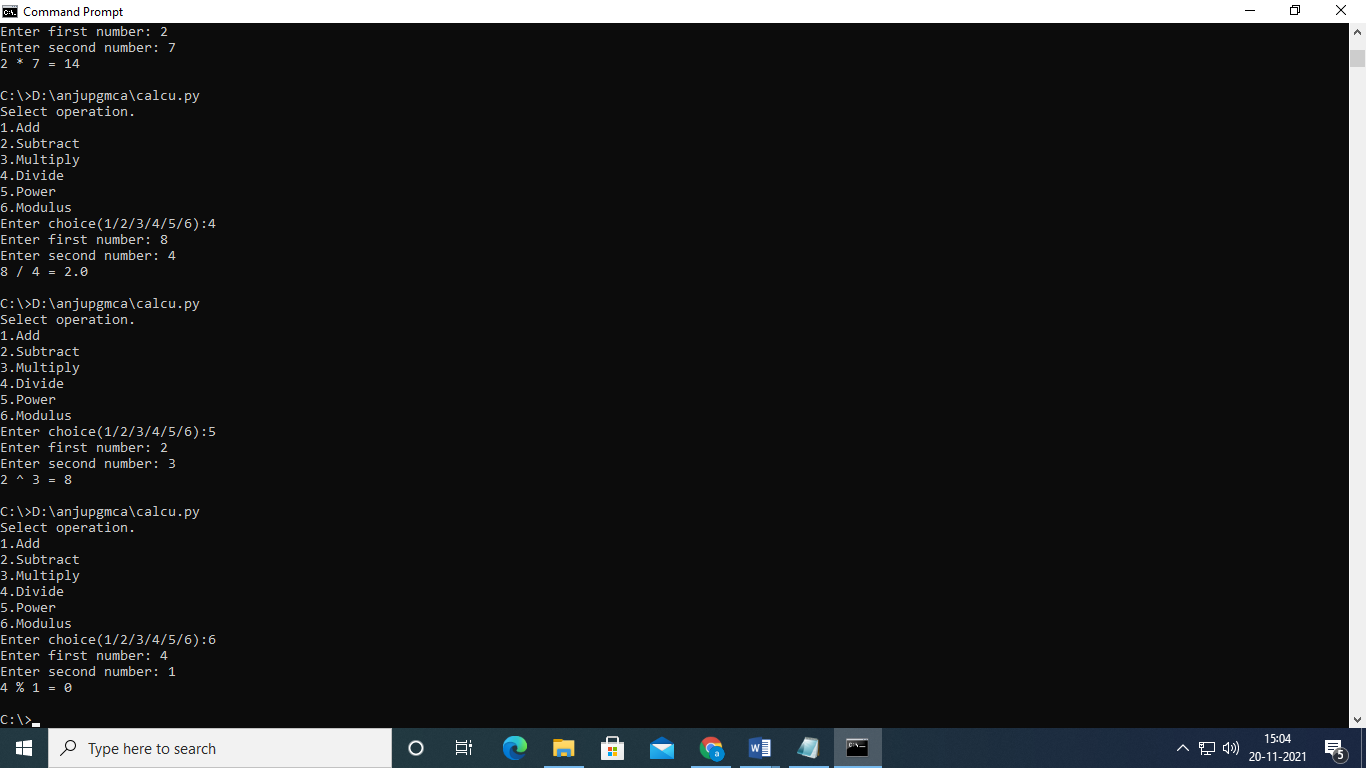
elif choice == '6':

   print(num1,"%",num2,"=", mod(num1,num2))

else:

   print("Invalid input")





21.PROGRAM TO FIND FACTORIAL OF A NUMBER USING RECURSION

def rec(n):

if(n==1):

return 1

else:

return n\*rec(n-1)

num=int(input("Enter the number:"))

if(num<0):

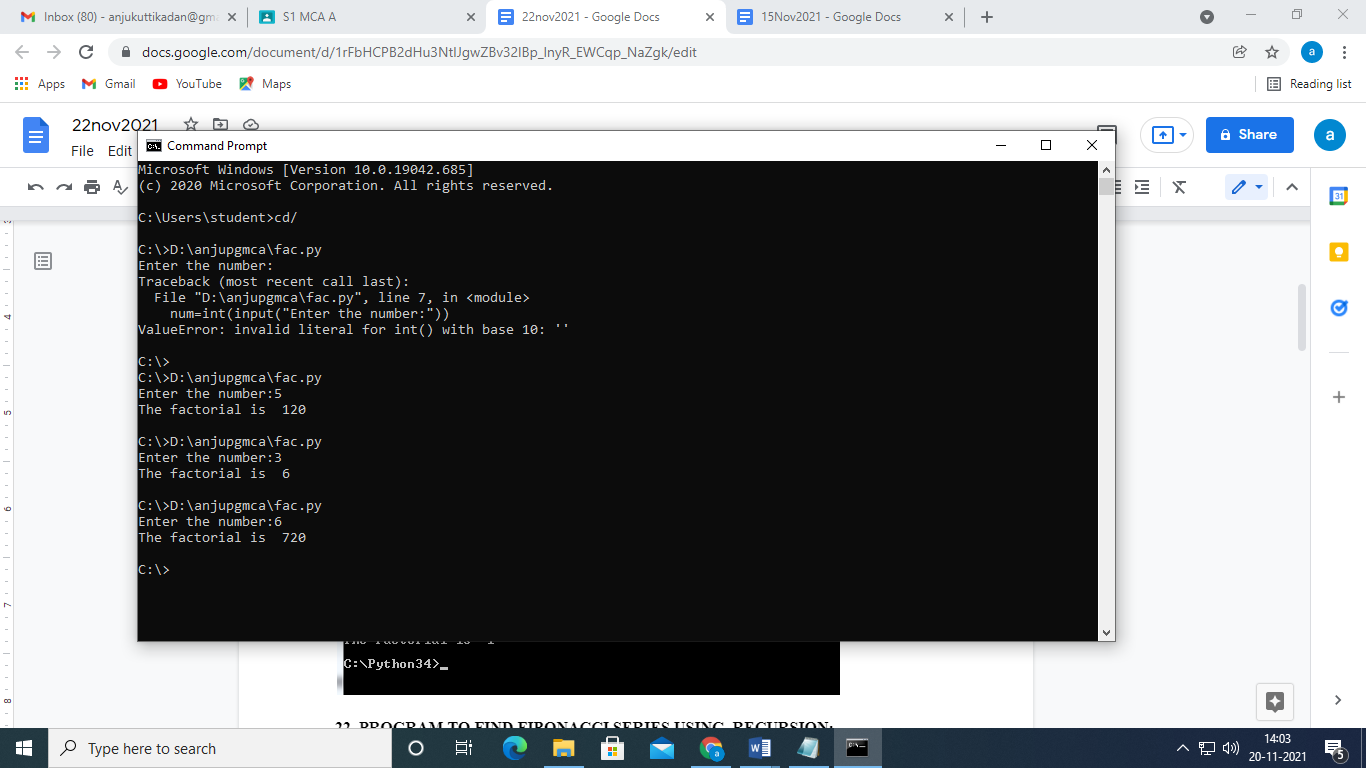
print("Factorial does not exist")

elif(num==0):

print("Factorial is 1")

else:

print("The factorial is ",rec(num))



22.PROGRAM TO PRINT FIBONACCI SERIES USING RECURSION

def fib(n):

if n<=1:

return(n)

elif n>0:

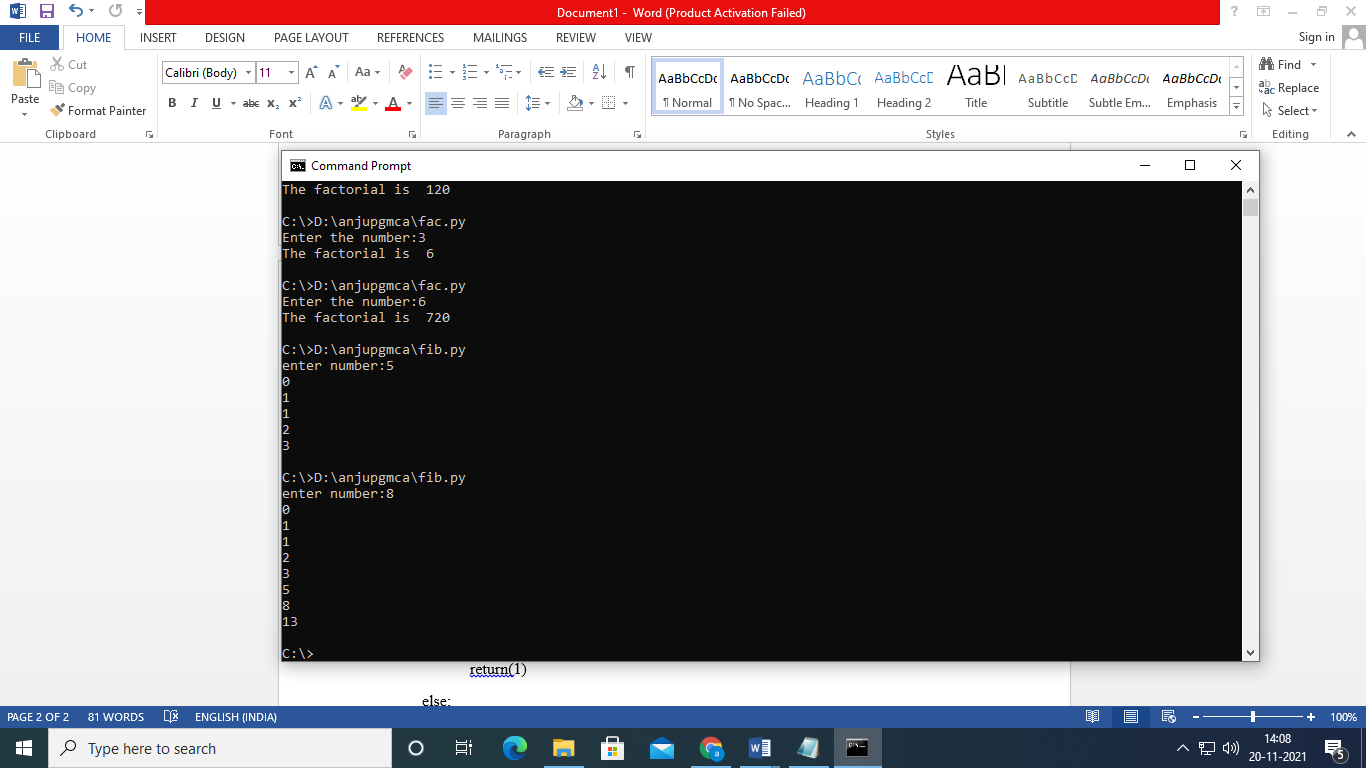
return(fib(n-1)+fib(n-2))

l=int(input("enter number:"))

for i in range(0,l):

r=fib(i)

print(r)



23. PROGRAM TO FIND POWER OF A NUMBER USING RECURSION

def pow(x,n):

if n==0:

return(1)

else:

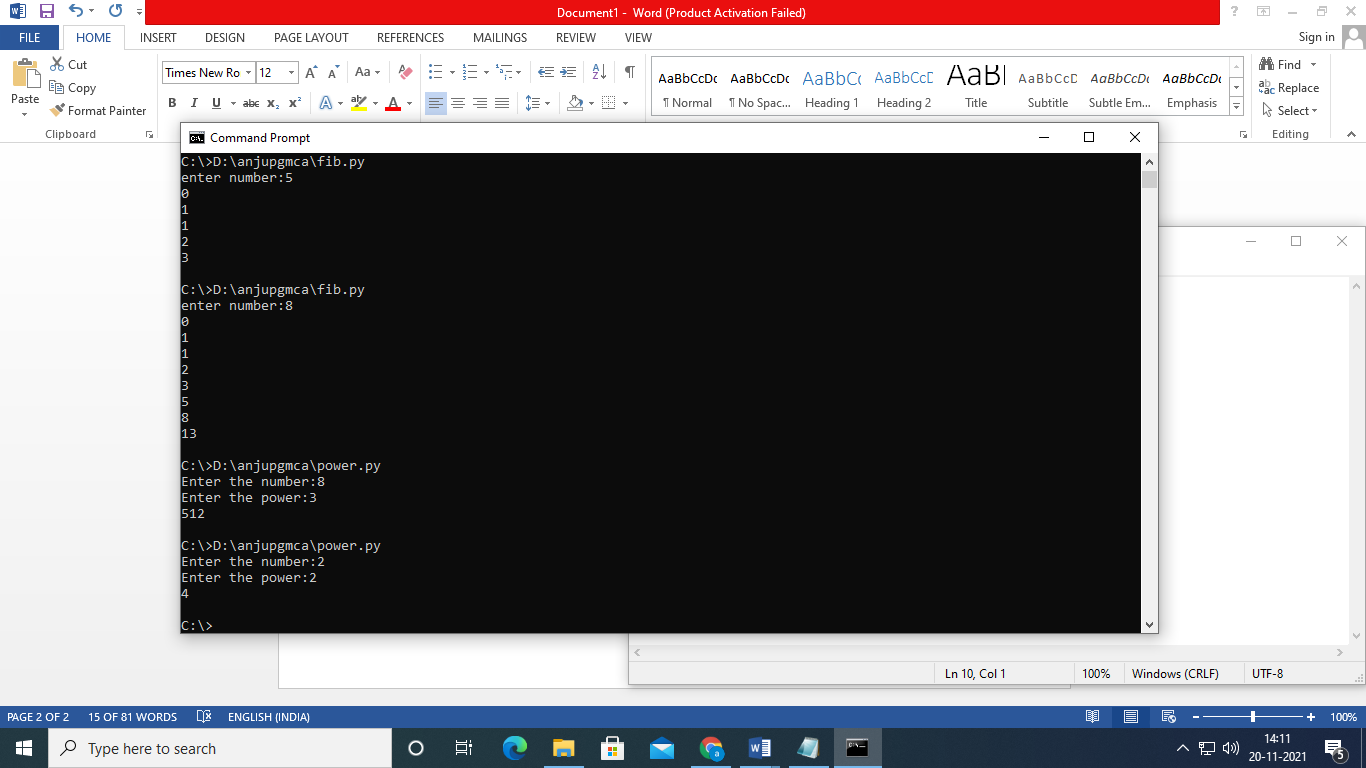
return(x\*pow(x,n-1))

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

b=int(input("Enter the power:"))

r=pow(a,b)

print(r)



24. Store a list of first names. Count the occurrences of ‘a’ within the list.

str=input("enter the string:")

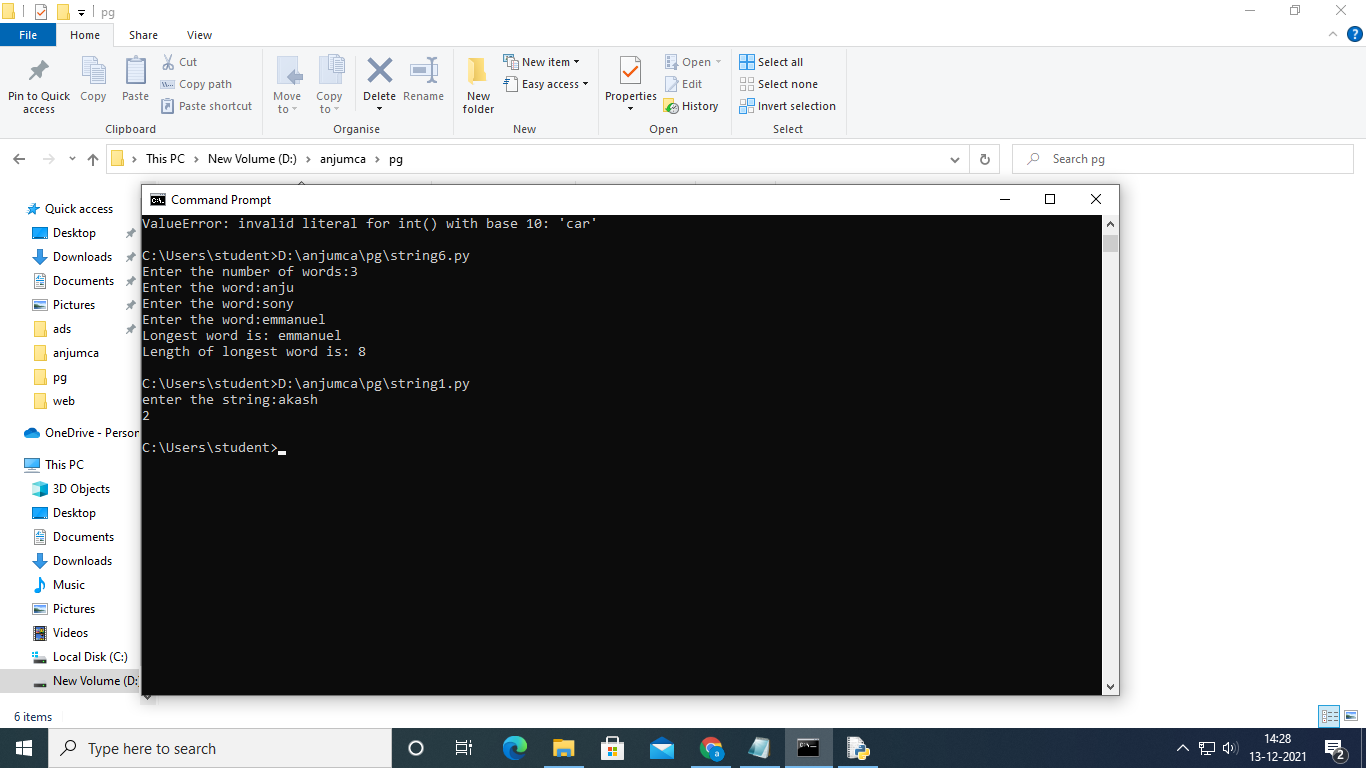
count=0

for i in range(len(str)):

if(str[i]==a):

count=count+1

print(count)



**25. Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.**

**[eg: onion -> oni$n] .**

str=input("enter the string:")

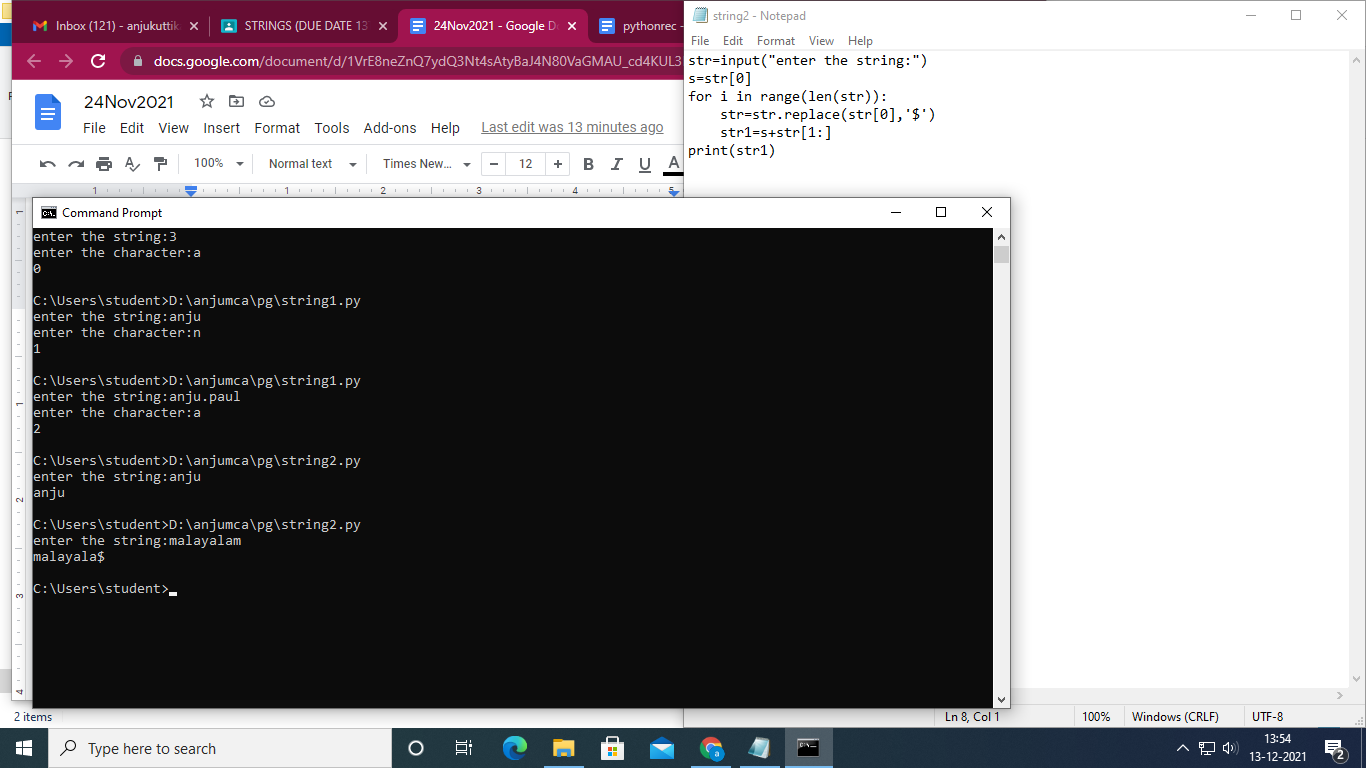
s=str[0]

for i in range(len(str)):

str=str.replace(str[0],'$')

str1=s+str[1:]

print(str1)



**26 . Create a string from given string where first and last characters exchanged. [eg: python - > nythop]**

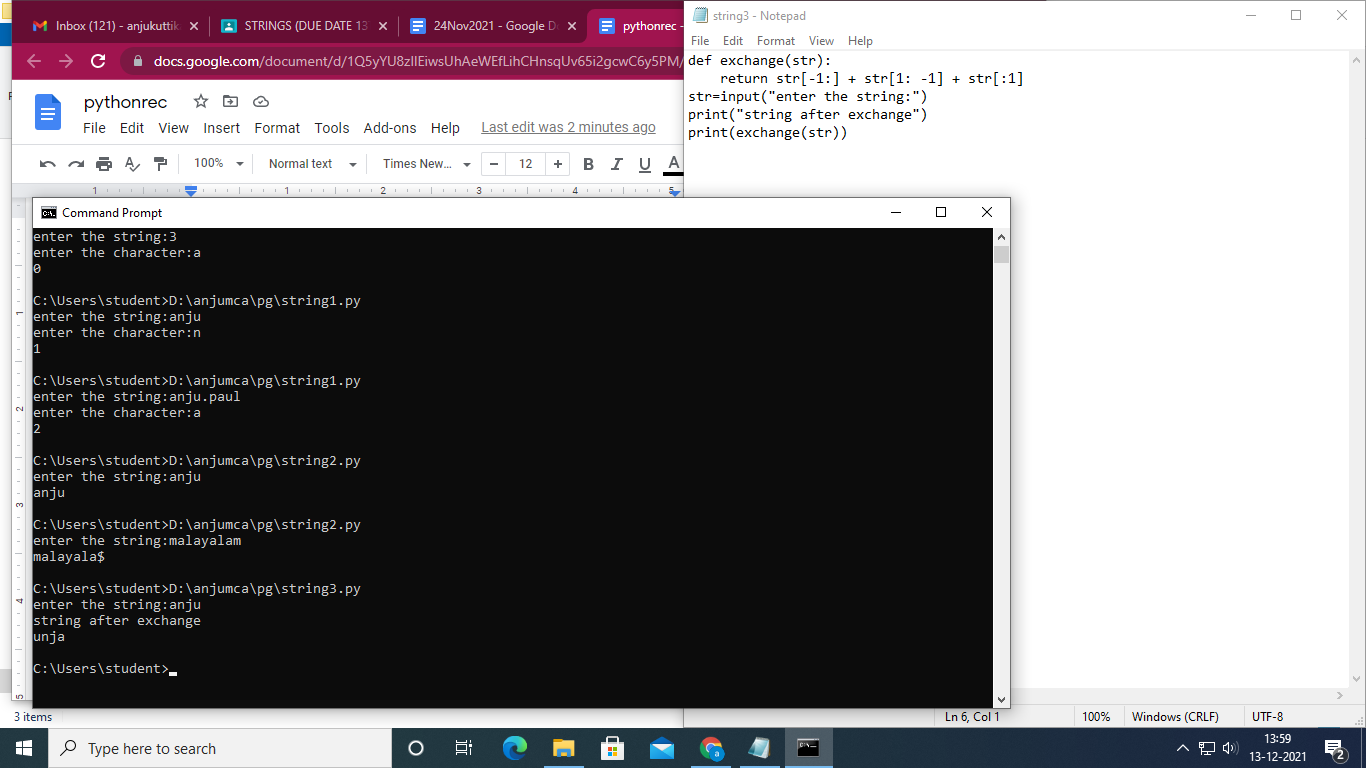
def exchange(str):

return str[-1:] + str[1: -1] + str[:1]

str=input("enter the string:")

print("string after exchange")

print(exchange(str))



**27. Create a single string separated with space from two strings by swapping the character at position 1.**

s1= input("Enter first string:")

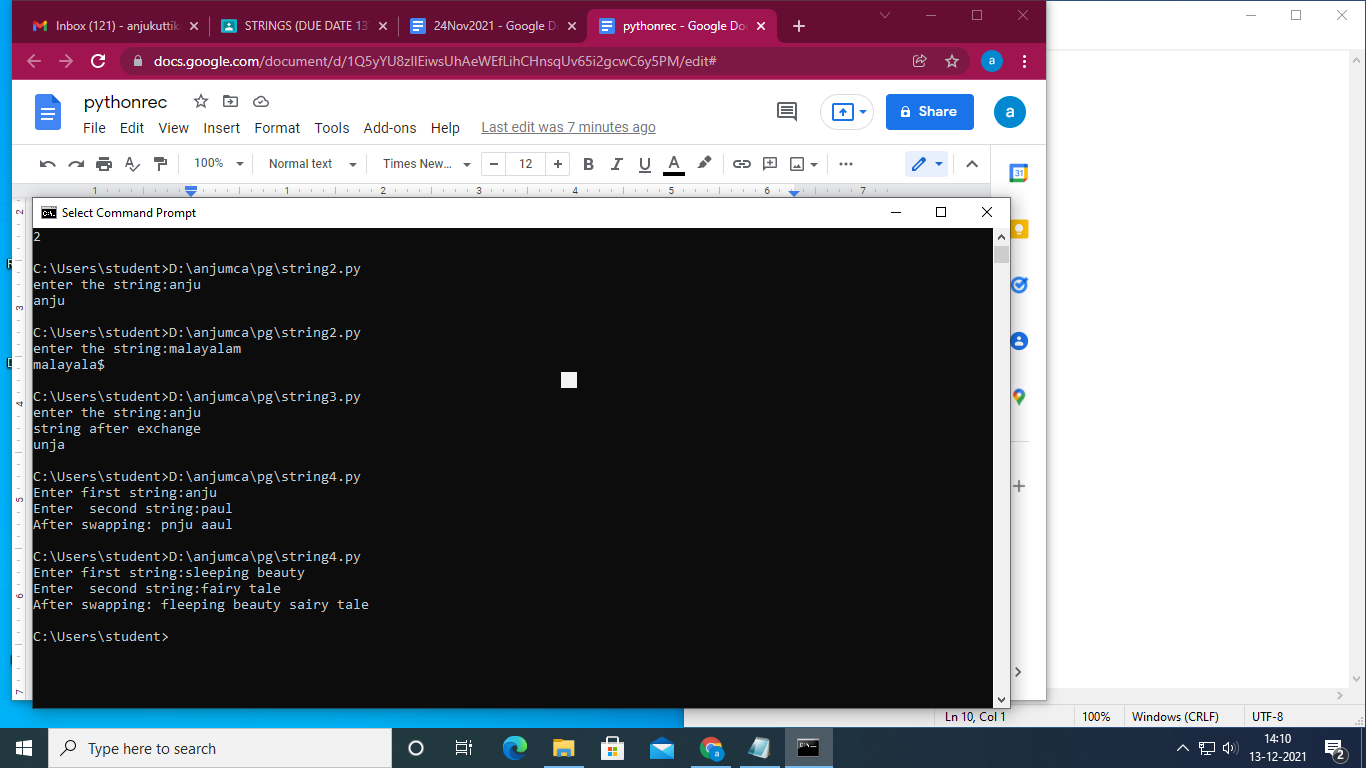
s2= input("Enter second string:")

a=s1.replace(s1[0:1],s2[0:1])

b=s2.replace(s2[0:1],s1[0:1])

c=a+' '+b

print("After swapping:",c)



**28. Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’**

n = input("Enter the string:")

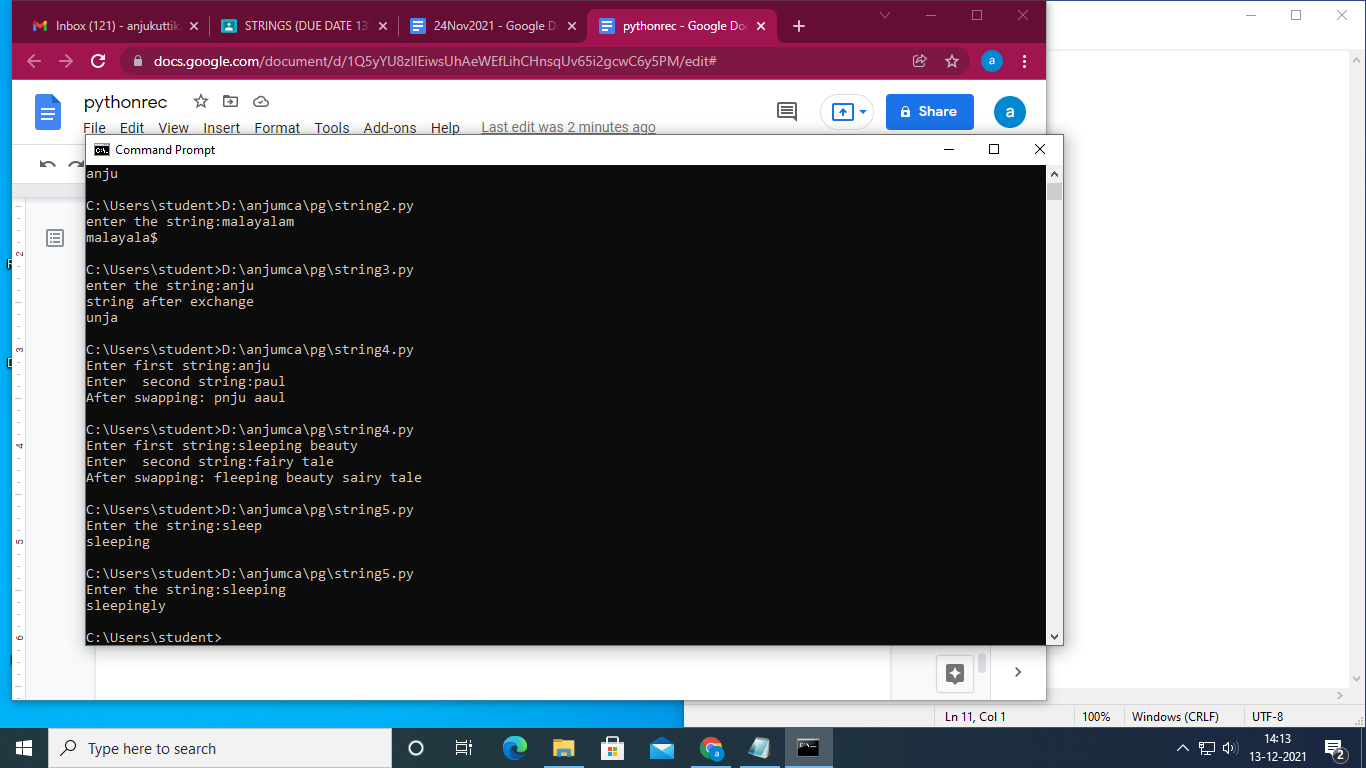
if n[-3:] == 'ing':

n = n+'ly'

else:

n = n+'ing'

print(n)



**29. Accept a list of words and return length of longest word.**

a =[]

n = int(input("Enter the number of words:"))

for i in range(0, n):

m = input("Enter the word:")

a.append(m)

max\_length =len(a[0])

temp=a[0]

for i in a:

if (len(i)>max\_length):

max\_length=len(i)

temp=i

print("Longest word is:",temp)

print("Length of longest word is:",max\_length)



**30. Count the number of characters (character frequency) in a string.**

s1=input("Enter the string : ")

freq = {}

for i in s1:

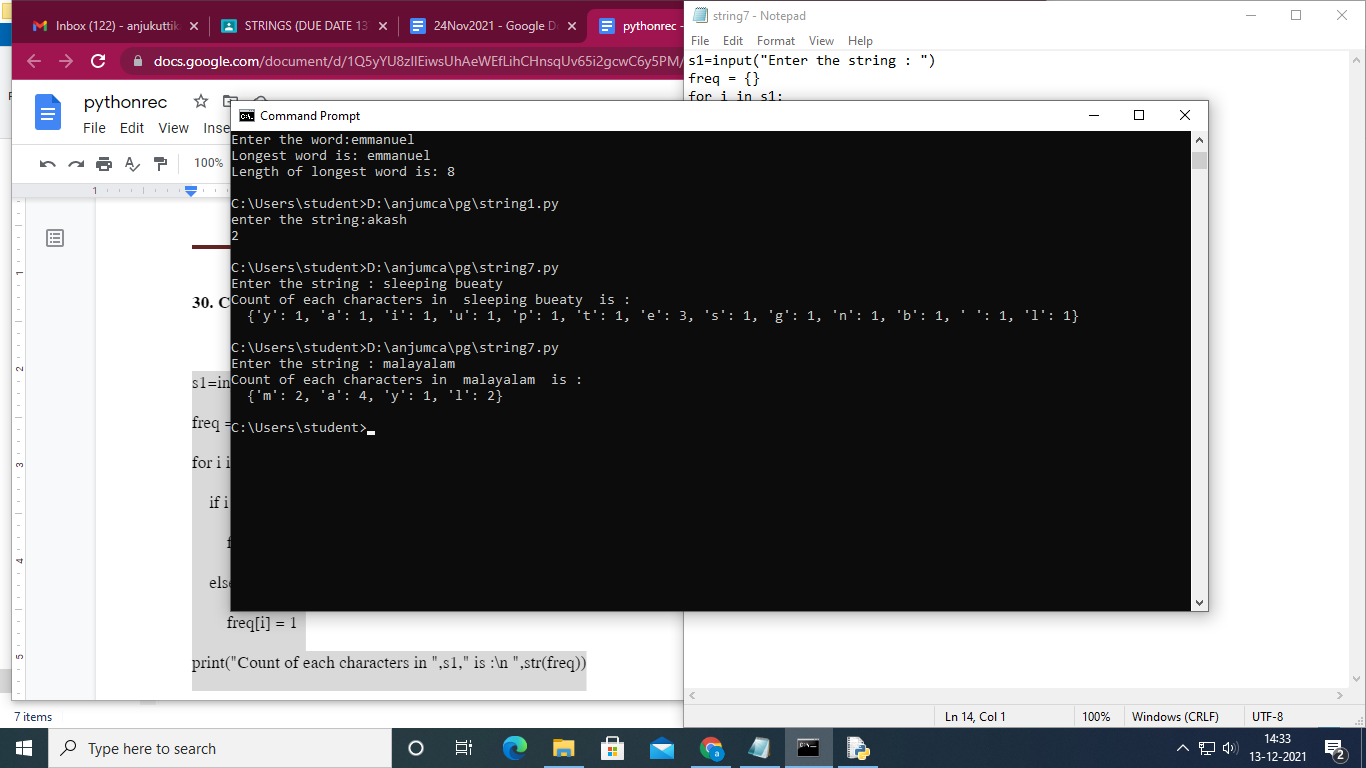
if i in freq:

freq[i]+=1

else:

freq[i] = 1

print("Count of each characters in ",s1," is :\n ",str(freq))



31. To acess a file

PRO

fo = open('abc.txt', "r+")

print("Name of the file: ", fo.name)

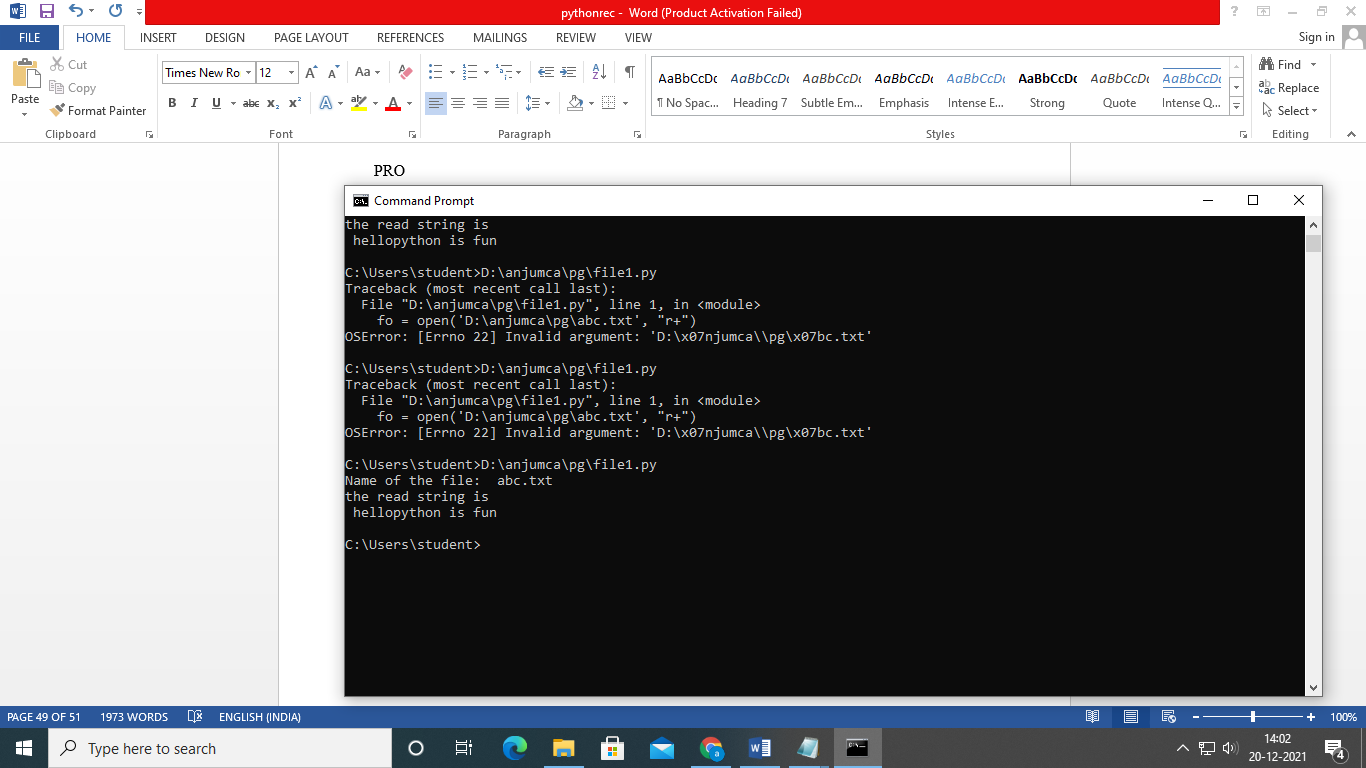
hi = "hellopython is fun for everyone"

fo.write(hi)

str = fo.read(18)

print("the read string is\n", str)

fo.close()



32.To obtain a file extention.

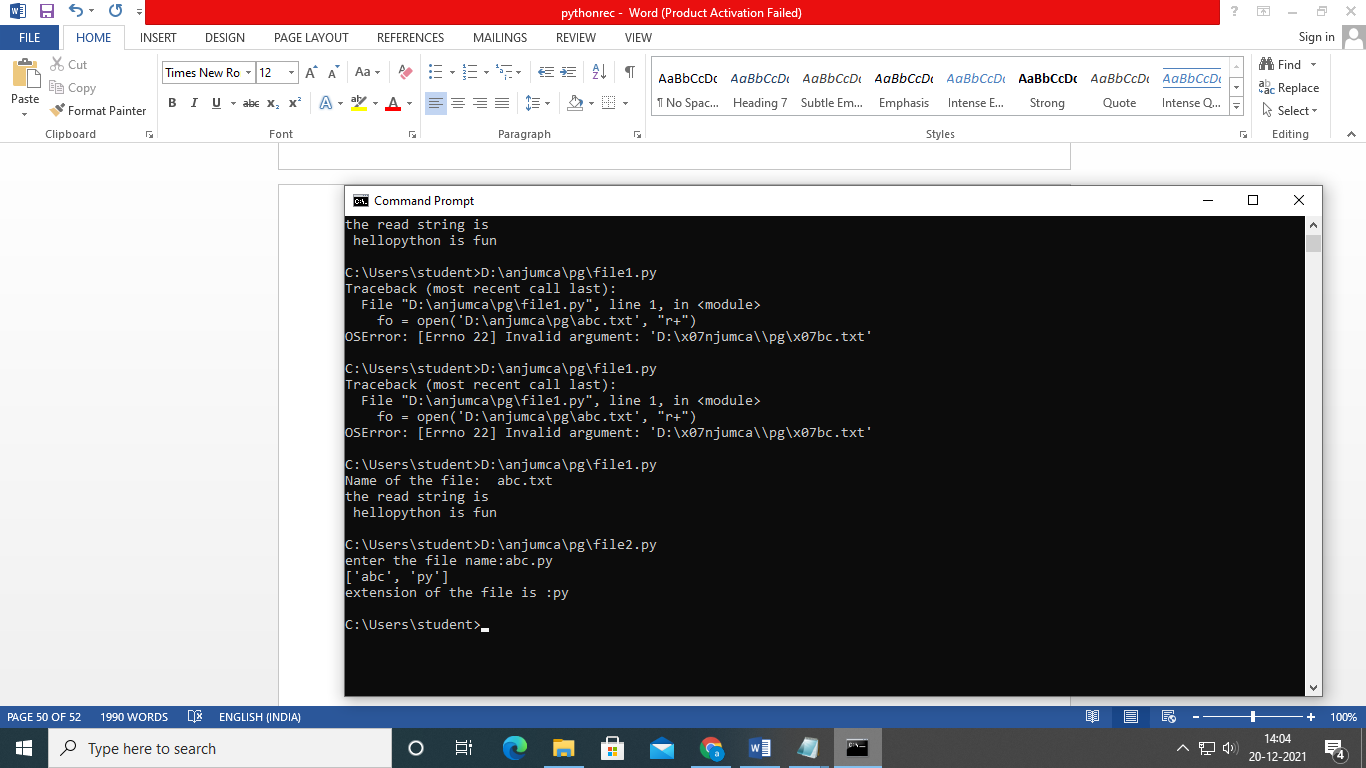
f=[]

file=input("enter the file name:")

f= file.split(".")

print(f)

print("extension of the file is :" + f[-1])



**33. To copy odd lines from a file to another file**

**fn=open('D:/anjumca/pg/abc.txt','r')**

**fn1=open('aa.txt','w+')**

**cont=fn.readlines()**

**print(cont)**

**for i in range(0, len(cont)):**

**if(i % 2 ==0):**

**fn1.write(cont[i])**

**else:**

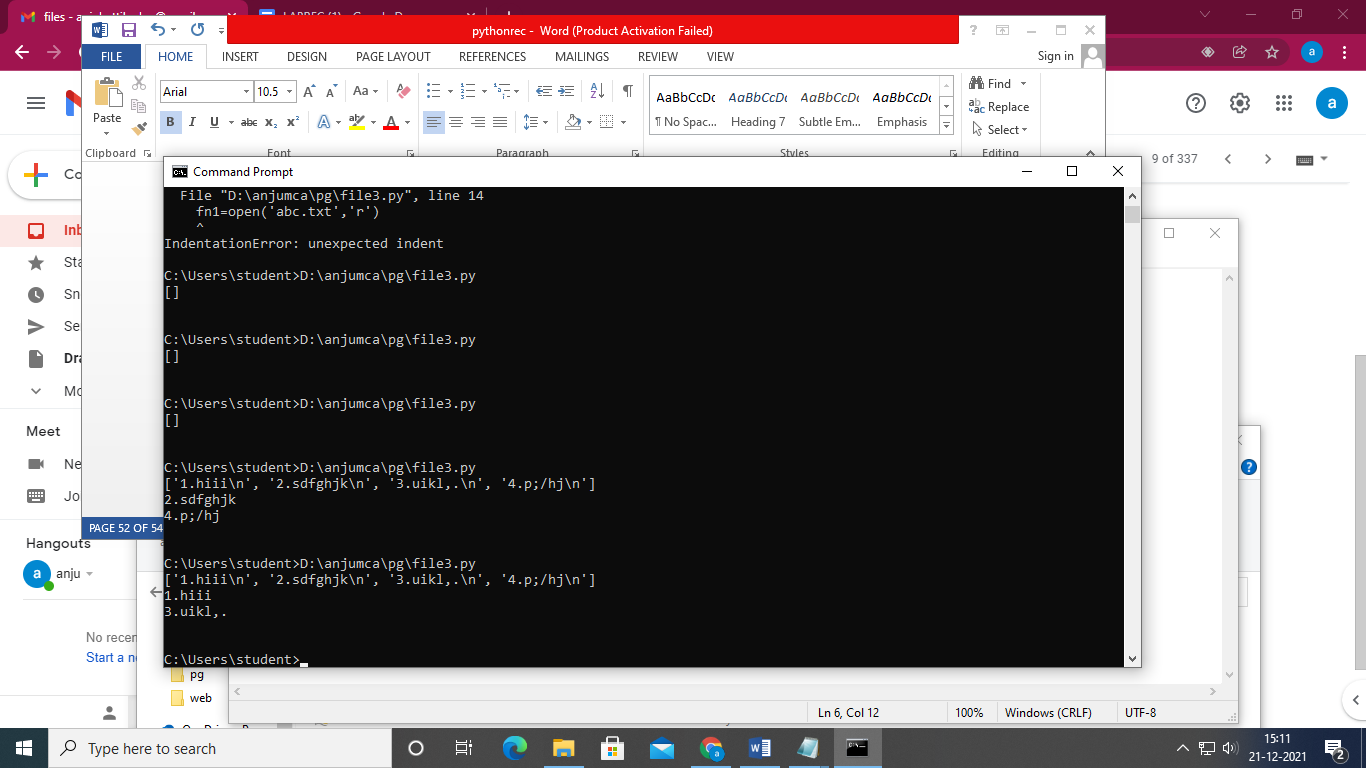
**continue**

**fn1.close()**

**fn1=open('aa.txt','r')**

**print(fn1.read())**

**fn1.close()**



**34.** **PROGRAM TO CREATE A NAMELIST OF STUDENT IN ALPHABETICAL ORDER:**

stud={}

n=int(input("Enter the number of students:"))

for i in range(0,n):

a=input("Enter the name of student:")

stud[a]=int(input("Enter the mark of student:"))

print(stud)

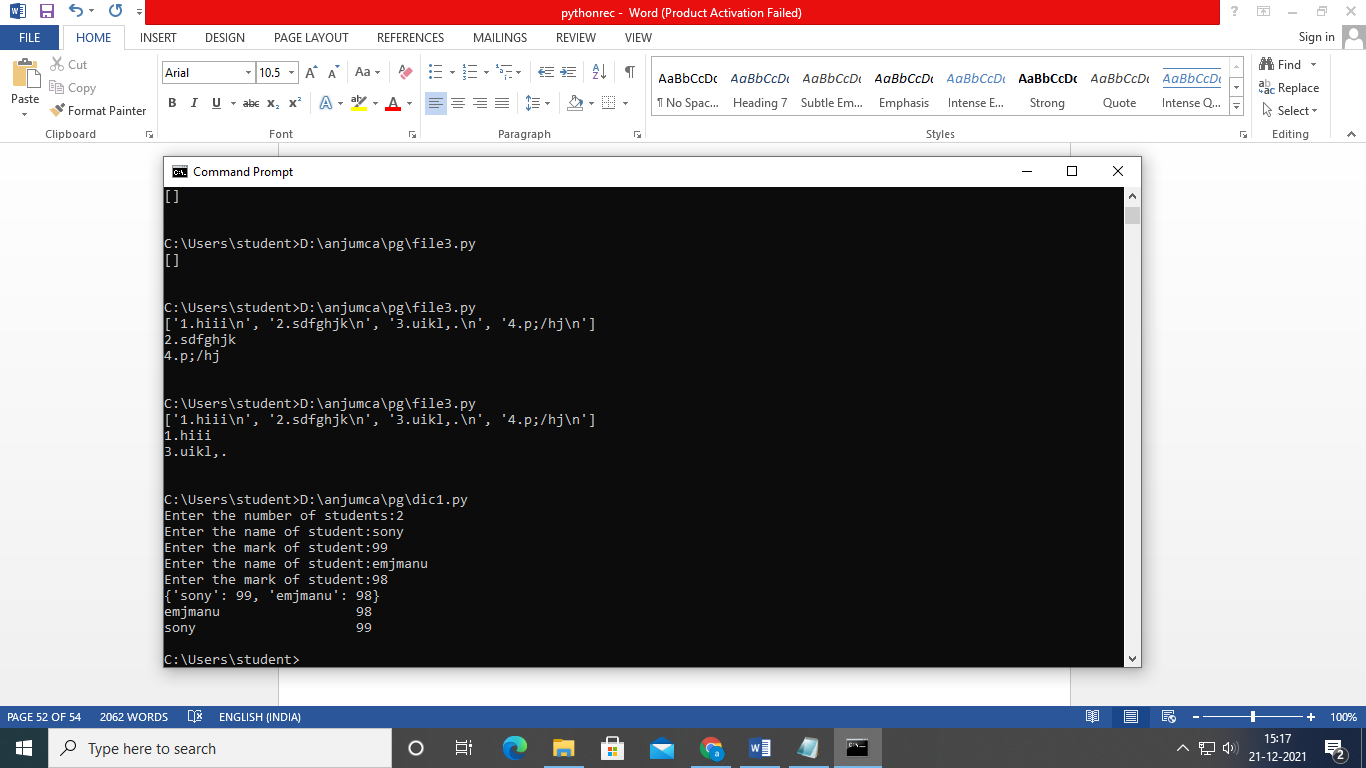
b=[]

b+=stud.keys()# append each key value pair to a list named b

b.sort()

for i in range(n):

print("%s\t\t\t%d" %(b[i],stud[b[i]]))



**35. PROGRAM TO CREATE A LIST OF STUDENT RECORDS USING DICTIONARY:**

**n=int(input("enter number of students:"))**

**studlist=[]**

**for i in range(0,n):**

**stud={}**

**stud['name']=input("Enter name ")**

**stud["branch"]=input("Enter branch: ")**

**stud["roll no"]=int(input("Enter roll no:\n"))**

**studlist.append(stud)**

**print("name branch roll no")**

**print()**

**print()**

**for i in range(0,n):**

**print(studlist[i])**



**36.PROGRAM TO PRINT RANKLIST OF STUDENTS(Descending order):**

stud={}

n=int(input("Enter the number of students:"))

for i in range(n):

        a=input("Enter name:")

        stud[a]=int(input("Enter the mark :"))

mark=[]

print(stud)

mark+=stud.values()

mark.sort()

a=[]

for i in mark:

for k,v in stud.items():

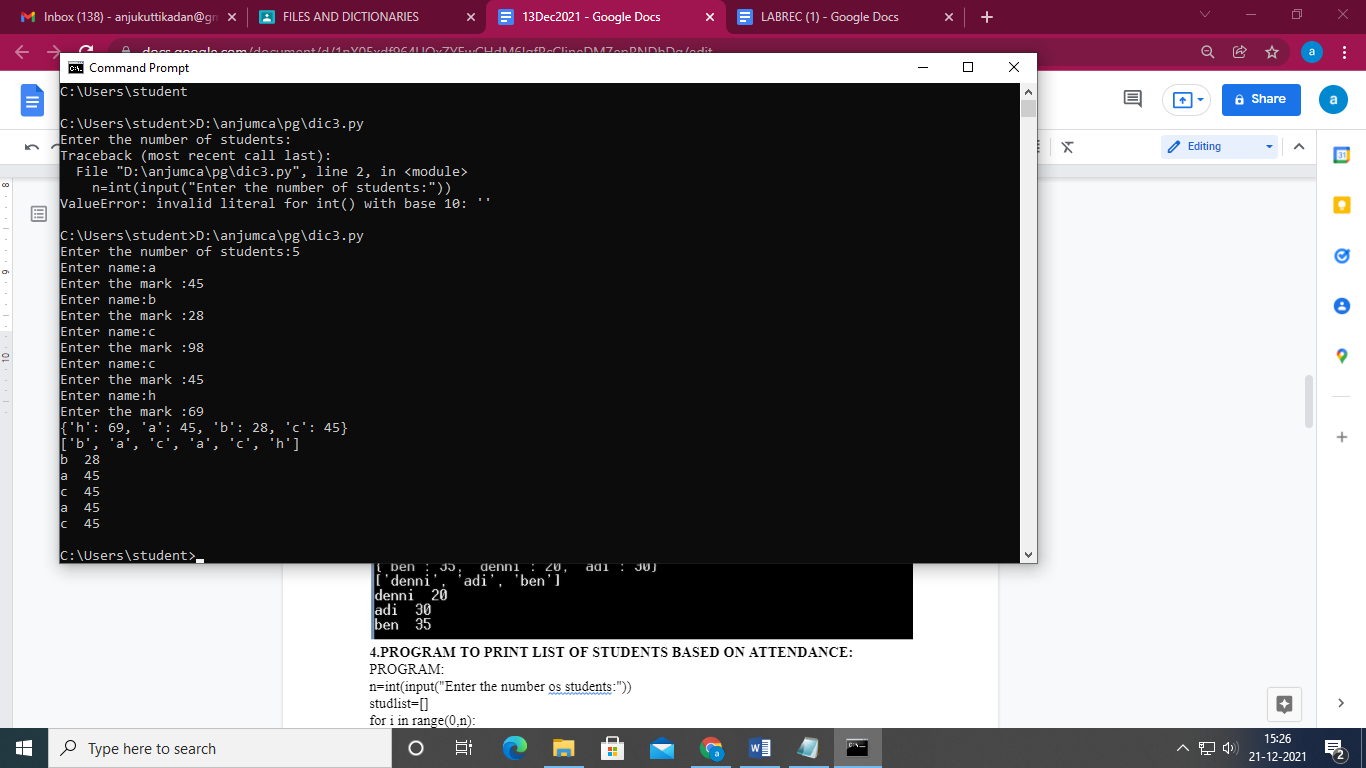
if(i==v):

a.append(str(k))

print(a)

for i in range(n):

        print("%s  %d" %(a[i],stud[a[i]]))



**37. PROGRAM TO PRINT LIST OF STUDENTS BASED ON ATTENDANCE**

n=int(input("Enter the number os students:"))

studlist=[]

for i in range(0,n):

stud={}

stud["name"]=input("Enter the name:")

stud["roll"]=int(input("Enter the roll number:"))

stud["attendance"]=int(input("Enter the attendance percentage:"))

studlist.append(stud)

for i in range(1,n):

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

if(studlist[j]["attendance"]<studlist[j+1]["attendance"]):

temp=studlist[j]

studlist[j]=studlist[j+1]

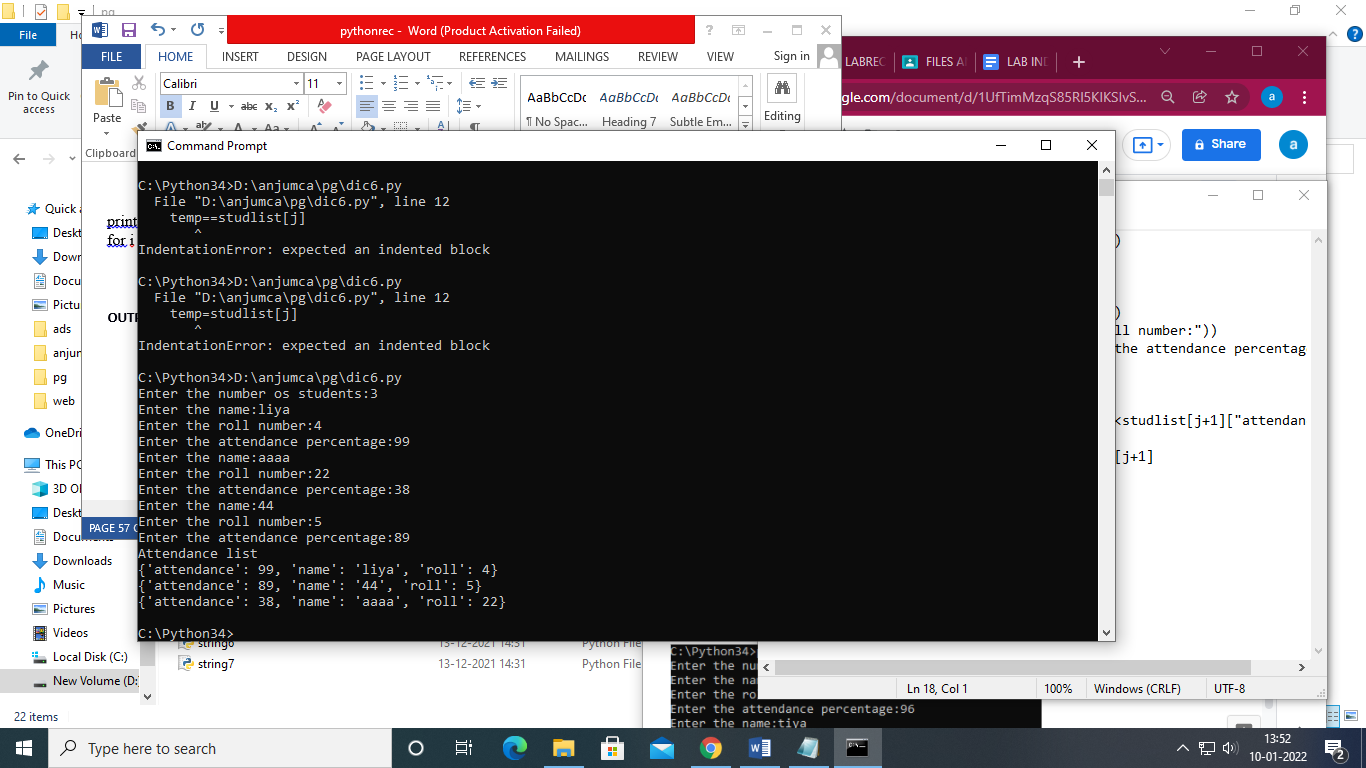
studlist[j+1]=temp

print("Attendance list")

for i in range(0,n):

print(studlist[i])

**OUTPUT:**



**38**Program to represent a sparse matrix using dictionary

a=[]

print("Square matrix")

n=int(input("Enter no. of rows and columns: "))

b={}

a=[[]\*n for i in range (n)]

print("\n Enter the elements of matrix: ")

for i in range (0,n):

for j in range (0,n):

a[i].append(int(input()))

print ("\nGiven matrix: \n",a)

for i in range (0,n):

for j in range (0,n):

if a[i][j]!=0:

b[(i,j)]=a[i][j]

print("Dictionary is :",b)

output



**39.SORT DICTIONARY IN ASCENDING AND DESCENDING ORDER.**

a = {'Mary': 20, 'Veronica': 30, 'Charles': 25, 'Hazel': 23}

print("Dictionary : ",a)

print("ascending order using keys : ",sorted(a.keys()))

print("ascending order using values : ",sorted(a.values()))

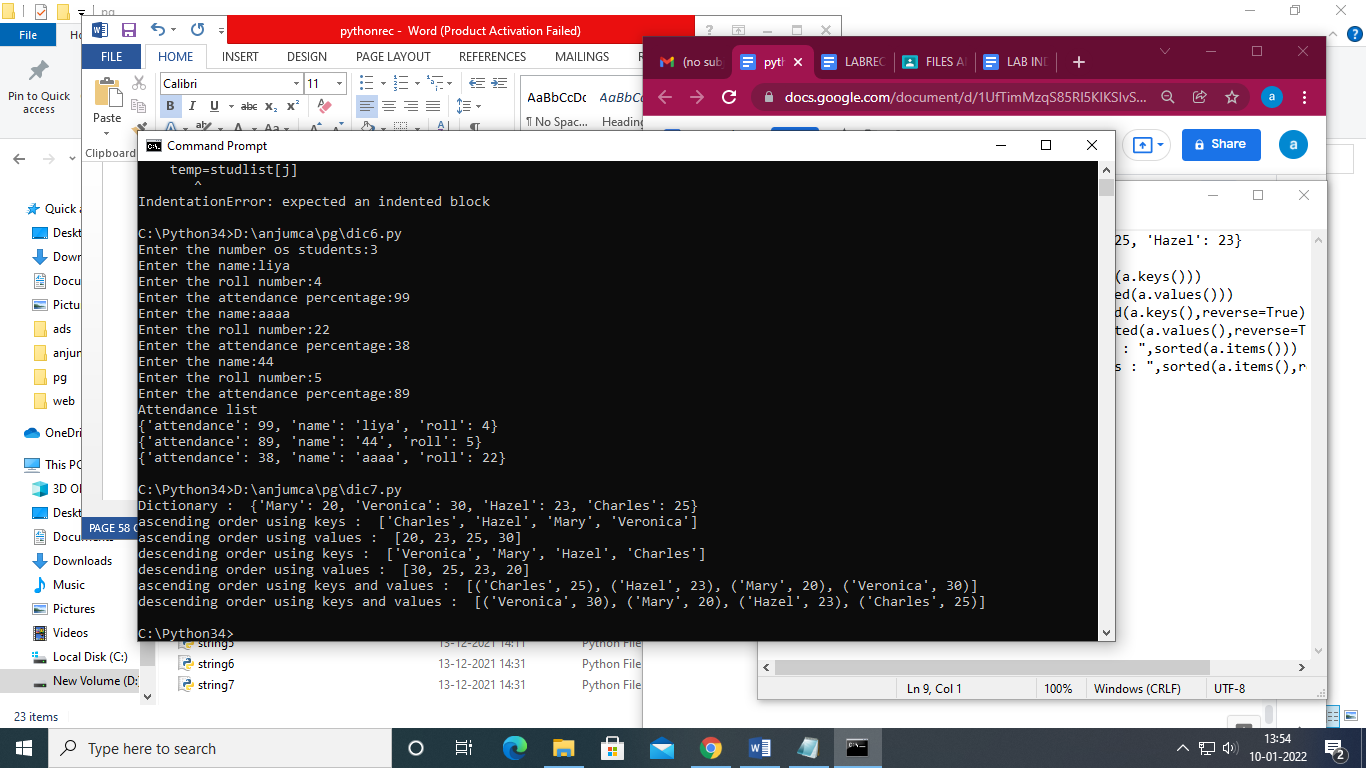
print("descending order using keys : ",sorted(a.keys(),reverse=True))

print("descending order using values : ",sorted(a.values(),reverse=True)) print("ascending order using keys and values : ",sorted(a.items()))

print("descending order using keys and values :

",sorted(a.items(),reverse=True))

**OUTPUT:**



**40.**Merge two dictionaries**.**

x={"name":"diya","age":20,"year":1999}

y={"sub":"python","year":2000,"class":3}

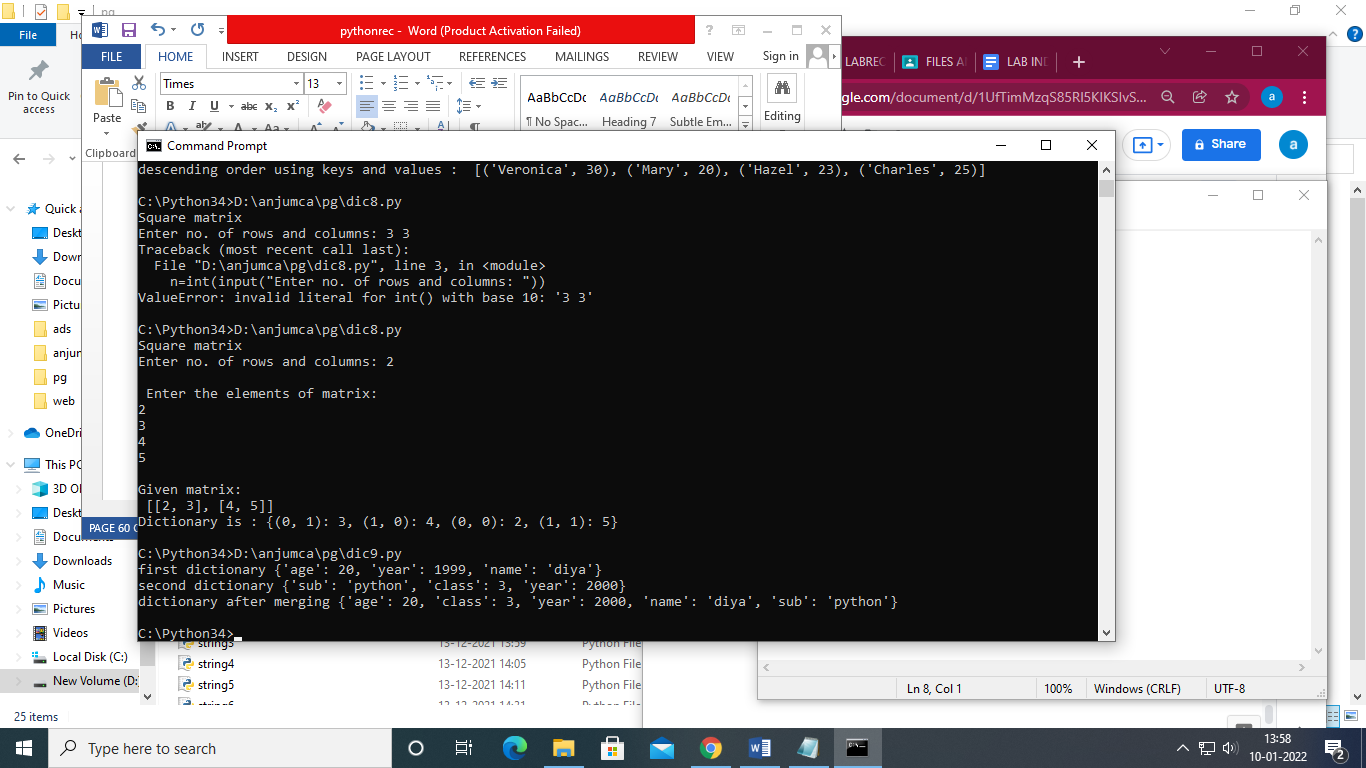
print("first dictionary",x)

print("second dictionary",y)

x.update(y)

print("dictionary after merging",x)

**output**



**EXCEPTION HANDLING**

**1. using try**

**Code**

f=input("Enter the filename:")

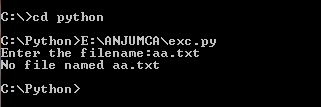
try:

f1=open(f,"r")

except IOError:

print("No file named",f)

**Output**

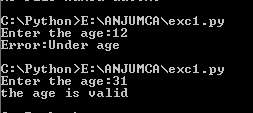


**2. using raise**

**Code**

try:      
    age = int(input("Enter the age:"))      
    if(age<18):      
        raise ValueError    
    else:      
        print("the age is valid")      
except ValueError:      
    print("Error:Under age")

**Output**



**3. using try -finally**

**Code**

try:      
    fileptr = open("newfile.txt","r")        
    try:      
        fileptr.write("Hi I am good")      
    finally:      
        fileptr.close()      
        print("file closed")      
except:      
    print("Error")

**Output**



**4.using try else**

**Code**

try:

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

    assert num % 2 == 0

except:

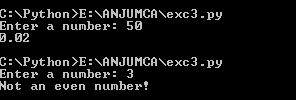
    print("Not an even number!")

else:

    reciprocal = 1/num

print(reciprocal)

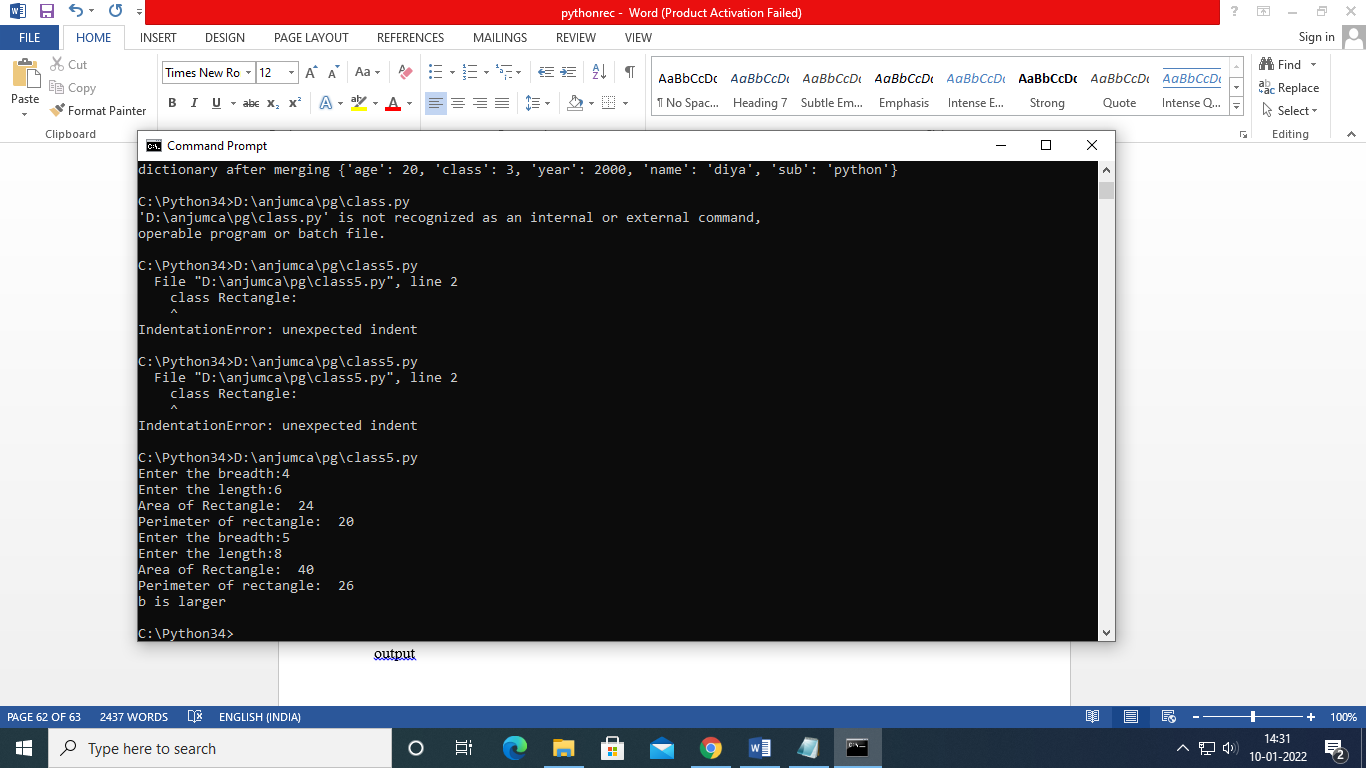
**Output**



42. **Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.**

class rectangle:  
    def \_init\_(self):  
        self.breadth  
        self.length  
  
    def size(self):  
        self.area= self.length \* self.breadth  
        self.p=2\*(self.length+self.breadth)  
        print("Area of Rectangle: ", self.area)  
        print("Perimeter of rectangle: ",self.p)  
        return self.area  
x=rectangle()  
x.breadth=int(input("Enter the breadth:"))  
x.length=int(input("Enter the length:"))  
a=x.size()  
y=rectangle()  
y.breadth=int(input("Enter the breadth:"))  
y.length=int(input("Enter the length:"))  
b=y.size()  
if(a>b):  
    print("a is larger")  
elif(a==b):  
    print("a and b are same")  
else:  
    print("b is larger ")

output



**43.        Create a Bank account with members account number, name, type of account and balance. Write    constructor and methods to deposit at the bank and withdraw an amount from the bank**

class Bank:

def \_init\_(self):

self.balance=0

print("\_\_Create an Account\_\_")

self.name=input("Enter the name :")

self.acctno=int(input("Enter the account number :"))

self.typeofac=input("Enter the type of account :")

def deposit(self):

amount=int(input("Enter the amount to be deposited :"))

self.balance+=amount

print("Amount Deposited :",amount)

def withdraw(self):

amount=int(input("Enter the amount to be withdrawn :"))

if self.balance>=amount:

self.balance-=amount

print("Amount withdrawn: ",amount)

else:

print("Insufficient balance!")

def display(self):

print("Account Balance :",self.balance)

a=Bank()

a.\_init\_()

while(1):

print("\n1.Deposit\n2.Withdrawal\n3.Balance\n4.Exit\n")

ch=int(input("Enter your choice :"))

if ch==1:

a.deposit()

elif ch==2:

a.withdraw()

elif ch==3:

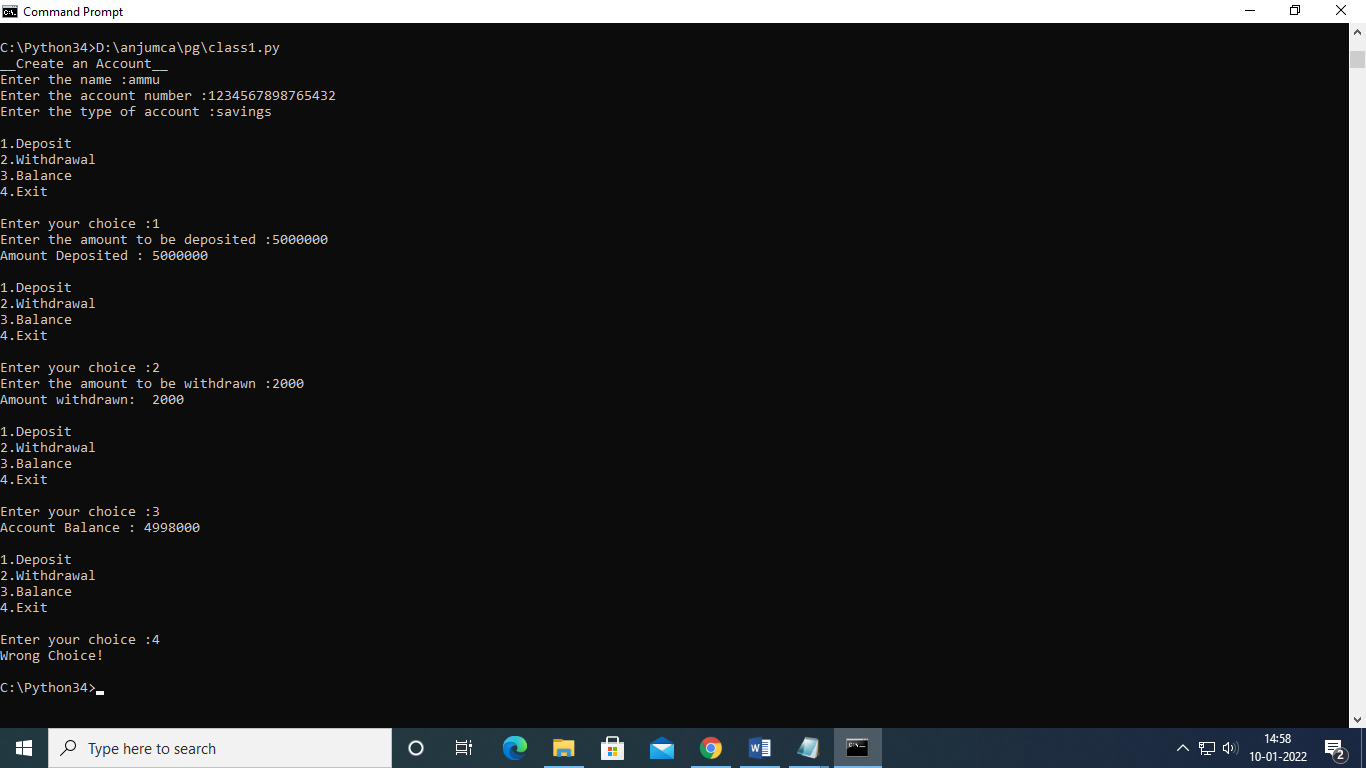
a.display()

else:

print("Wrong Choice!")

exit()

**output**



**44. . Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles. class rect:**

class rect:

def \_\_init\_\_(self, a, b):

self.length = a

self.breadth = b

def area(self):

return self.length \* self.breadth

def \_\_lt\_\_(self, o):

if self.area() < o.area():

return True

else:

return False

h1 = int(input("enter length of first rectangle"))

b1 = int(input("enter breadth of first rectangle"))

h2 = int(input("enter length of second rectangle"))

b2 = int(input("enter breadth of second rectangle"))

rect1 = rect(h1, b1)

rect2 = rect(h2, b2)

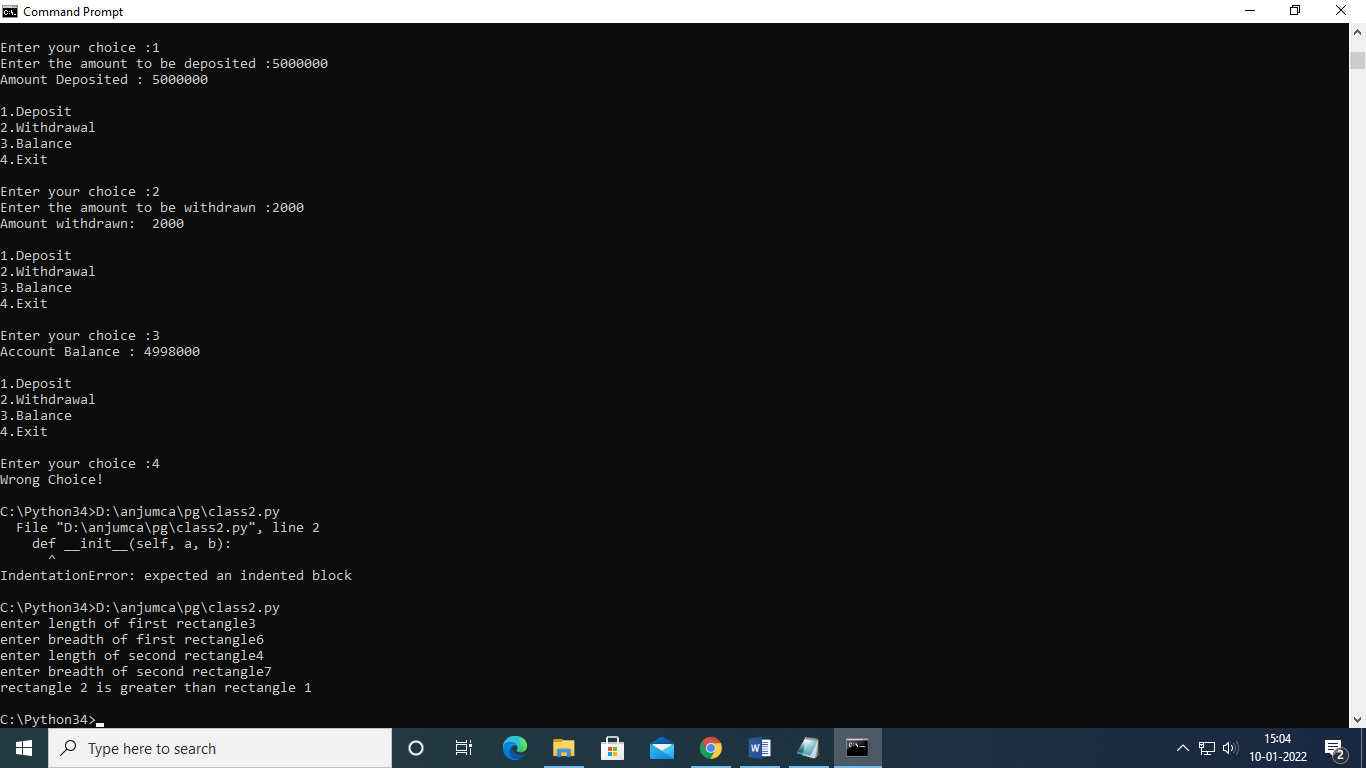
if rect1 < rect2:

print("rectangle 2 is greater than rectangle 1")

else:

print("rectangle 1 is greater than rectangle 2")

**output**



**45. Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to   find sum of 2 time.**

class Time:

def \_\_init\_\_(self, hours, mins, sec):

self.hours = hours

self.mins = mins

self.sec = sec

def \_\_add\_\_(t1, t2):

t3 = Time(0, 0, 0)

if t1.mins + t2.mins > 60:

t3.hours = (t1.mins + t2.mins) // 60

t3.hours = t3.hours + t1.hours + t2.hours

t3.mins = (t1.mins + t2.mins) % 60

t3.sec = (t1.sec + t2.sec)

return t3

def displayTime(self):

print("Time is ", self.hours, "hours ", self.mins, "minutes", "and", self.sec, "seconds")

def displayMinute(self):

print("Minutes:", self.hours \* 60 + self.mins)

def displaysecond(self):

print("Second:", self.hours \* 60 \* 60 + self.mins \* 60 + self.sec)

h1 = int(input("Enter time 1 in hour : "))

m1 = int(input("Enter time 1 in minutes :"))

s1 = int(input("Enter time 1 in second : "))

h2 = int(input("Enter time 2 in hour : "))

m2 = int(input("Enter time 2 in minutes : "))

s2 = int(input("Enter time 2 in second : "))

a = Time(h1, m1, s1)

b = Time(h2, m2, s2)

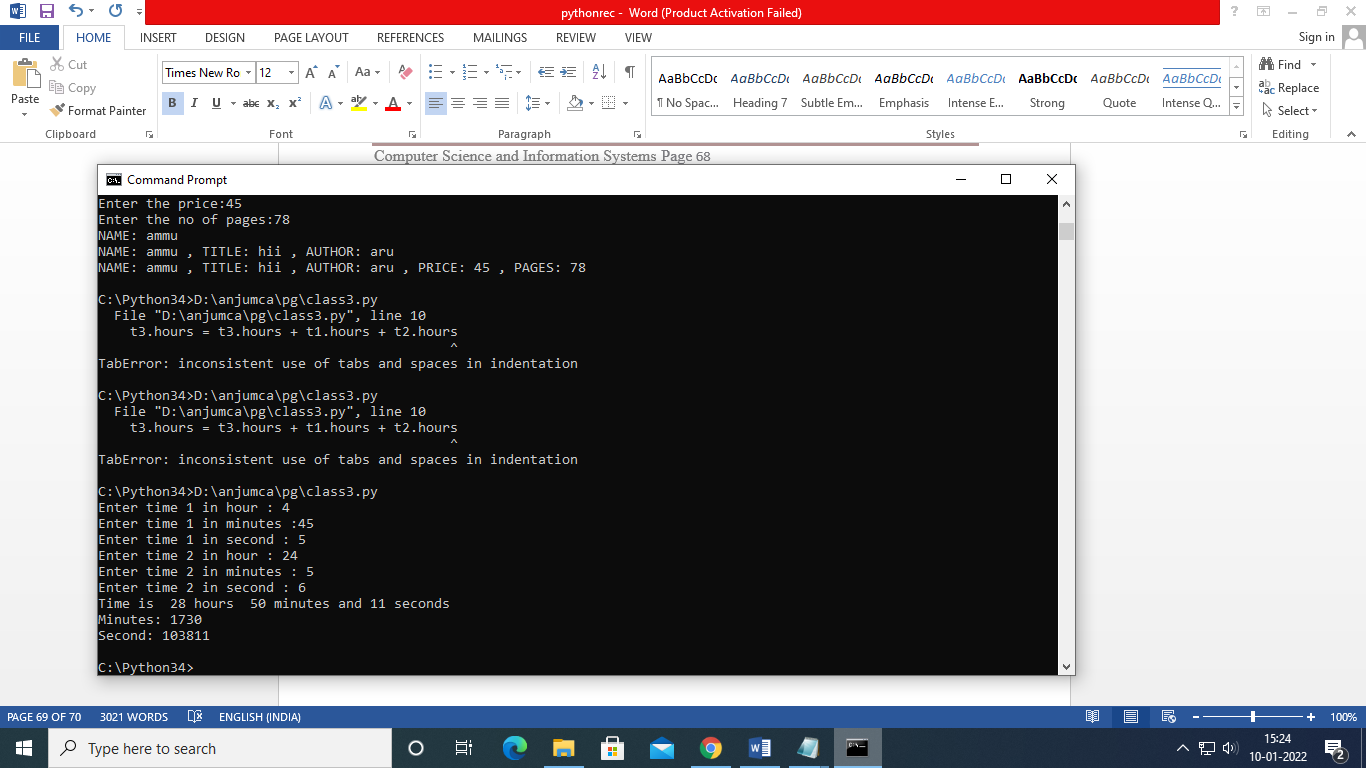
c = a + b

c.displayTime()

c.displayMinute()

c.displaysecond()

**Output**



**46.  Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding**

class Publisher():

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

self.name=name

def display(self):

print("NAME:",self.name)

class Book(Publisher):

def \_\_init\_\_(self,name,title,author):

self.title=title

self.author=author

super().\_\_init\_\_(name)

def display(self):

print("NAME:",self.name,",", "TITLE:",self.title,",","AUTHOR:", self.author)

class python(Book):

def \_\_init\_\_(self,name,title,author,price,no\_of\_pages):

self.price= price

self.noofpages= no\_of\_pages

super().\_\_init\_\_(name,title,author)

def display(self):

print("NAME:",self.name,",","TITLE:",self.title,",", "AUTHOR:", self.author,",","PRICE:",self.price,",","PAGES:",self.noofpages)

a=input("Enter the name:")

x=input("Enter the title:")

c=input("Enter the name of author:")

d=int(input("Enter the price:"))

e=int(input("Enter the no of pages:"))

t = Publisher(a)

b = Book(a,x,c)

p = python( a,x,c,d,e)

t.display()

b.display()

p.display()

**Output**

