FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

PYTHON PROGRAMMING

……..

LABORATORY RECORD

Name : **AKSHAY B**

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 Batch : SEMESTER -1 A

Roll No : 10

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name : AKSHAY B

Branch : **MASTER OF COMPUTER APPLICATION**

Semester : **1** **Roll No: 10**

University Exam.Reg. No:

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to Kerala Technological University in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by AKSHAY B *in the* ***PYTHON PROGRAMMING LAB*** *Laboratory of the Federal Institute of Science and Technology during the academic year 2020-2021.*

Signature of Staff in Charge
Name:
Date:

Signature of H.O.D
Name:

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

CONTENTS

12	**17/11/2021**	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	16	
13	**17/11/2021**	Accept an integer n and compute n+nn+nnn.	17	
14	**17/11/2021**	Print out all colors from color-list1 not contained in color-list2.	18	
15	**25/11/2021**	Create a single string separated with space from two strings by swapping the character at position 1.	19	
16	**2/12/21**	Sort dictionary in ascending and descending order.	20	
17	**2/12/21**	Merge two dictionaries.	21	
18	**2/12/21**	Find gcd of 2 numbers.	22	
19	**25/11/21**	From a list of integers, create a list removing even numbers.	23	
20	**25/11/21**	Program to find the factorial of a number.	24	
21	**25/11/21**	Generate Fibonacci series of N terms.	25	
22		Find the sum of all items in a list.	26	
23		Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	27	
24		Display the given pyramid with step number accepted from user.	28	
25		Count the number of characters (character frequency) in a string.	29	
26		Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’	30	
27		Accept a list of words and return length of longest word.	31	
28		Construct pattern using nested loop	32	
29		Generate all factors of a number.	33	
30		Write lambda functions to find area of square, rectangle and triangle.	34	
31		Work with built-in packages.	35	

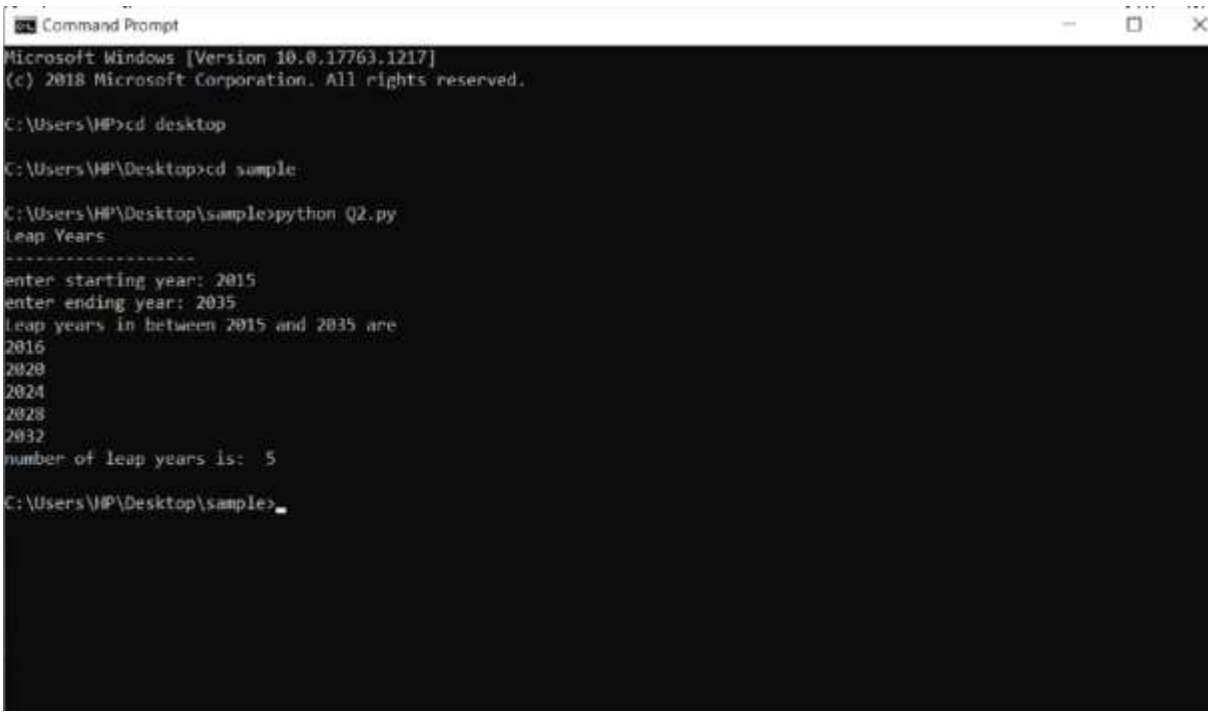
32		Create a package graphics with modules rectangle,circle.include method to find area and perimeter of respective figures in each.Write a program to find area and perimeter of figure by different importing statements.	36	
33		Create Rectangle class with attribute length and breadth and methods to find area and perimeter. Compare 2 Rectangle objects with their area.	37	
34		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	38	
35		Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	39	
36		Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of time	40	
37		Create a class Publisher(name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes and no of pages. Write a program that displays information about a Python book. Use base class constructor and method overriding.	41	
38		Write a Python program to read a file line by line and store it into a list.	42	
39		Python program to copy odd lines of one file to other.	43	

1. Display future leap years from current year to a final year entered by user.

Program:

```
print("Leap Years")
print("-------------------------")
start=int(input("enter starting year: "))
end=int(input("enter ending year: "))
c=0
print("Leap years in between" ,start ,"and",end,"are")
while start <= end :
if start % 4 == 0 and start % 100 !=0 :
print(start)
c+=1
if start % 100 == 0 and start % 400 == 0 :
 print(start)
start = start+1
print("number of leap years is: ",c)
```

Output:



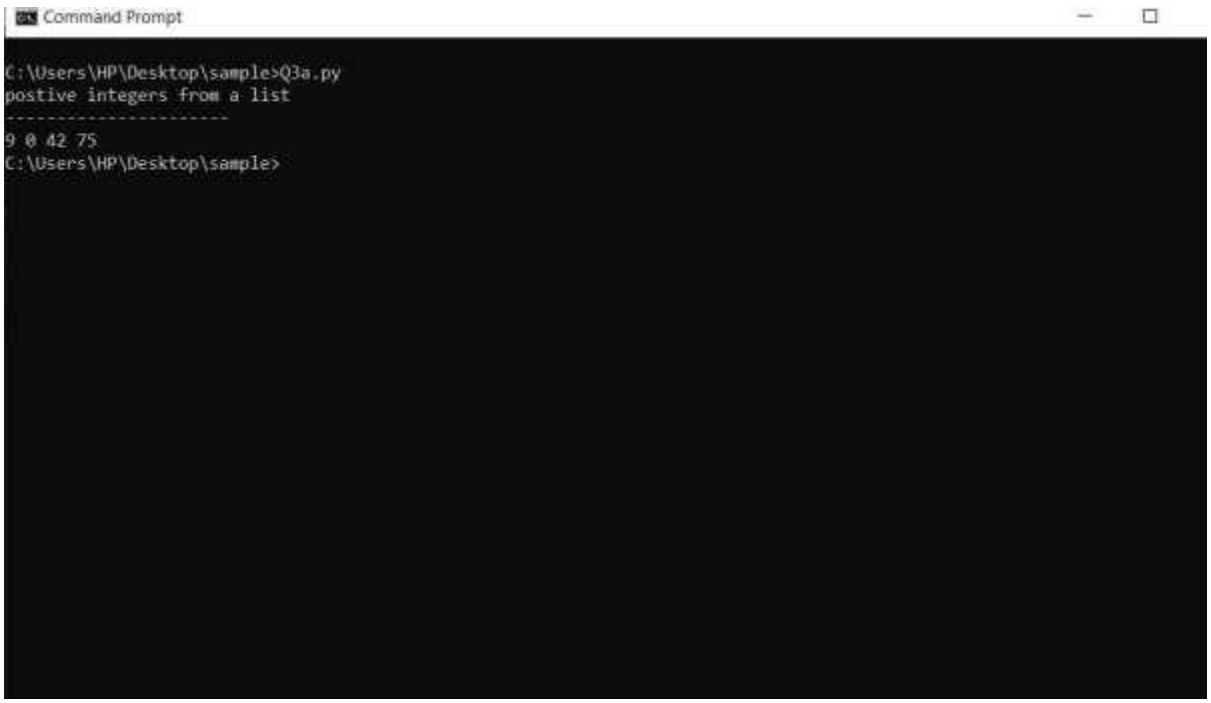
2. List comprehensions

a. Generate positive list of numbers from a given list of integers.

Program:

```
print("postive integers from a list")
print("-----------------------")
 list1=[9,-2,0,42,75,-33]
 for num in list1:
if num >=0 :
print(num,end=" ")
```

Output:

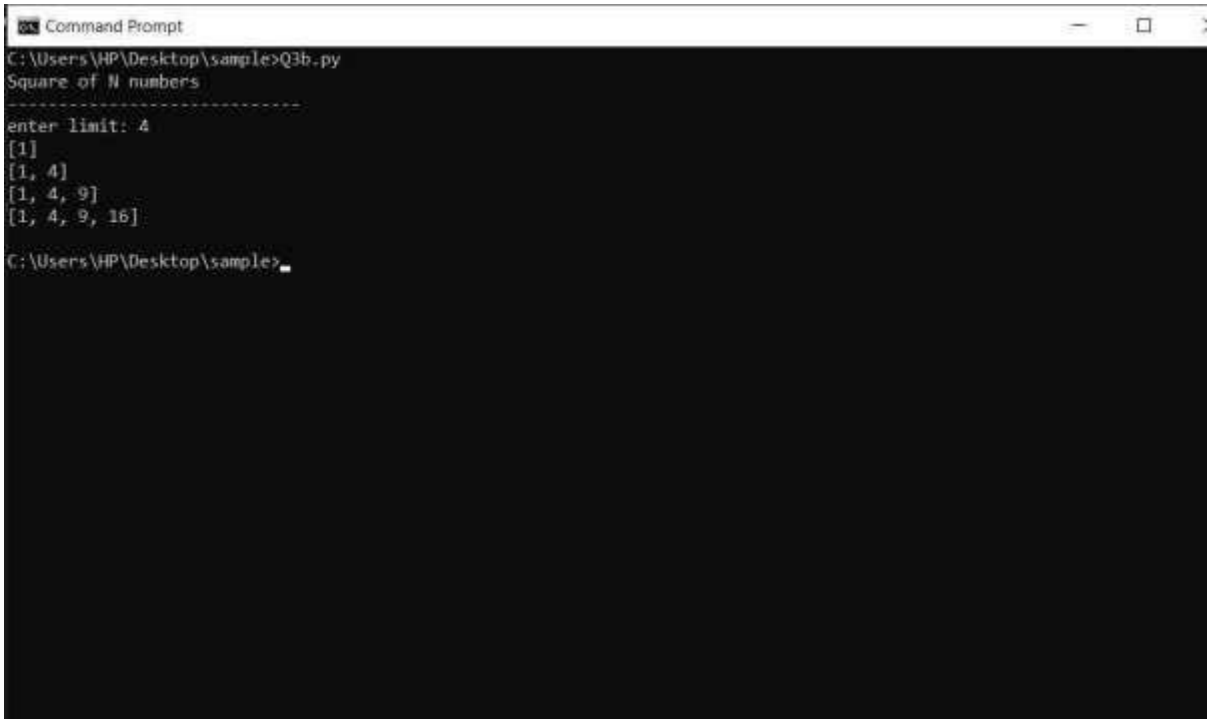


b. Square of N numbers.

Program:

```
print("Square of N numbers")
print("----------")
limit=int(input("enter limit: "))
list1=[]
for i in range(1,limit+1):
list1.append(i*i)
print(list1)
```

Output:

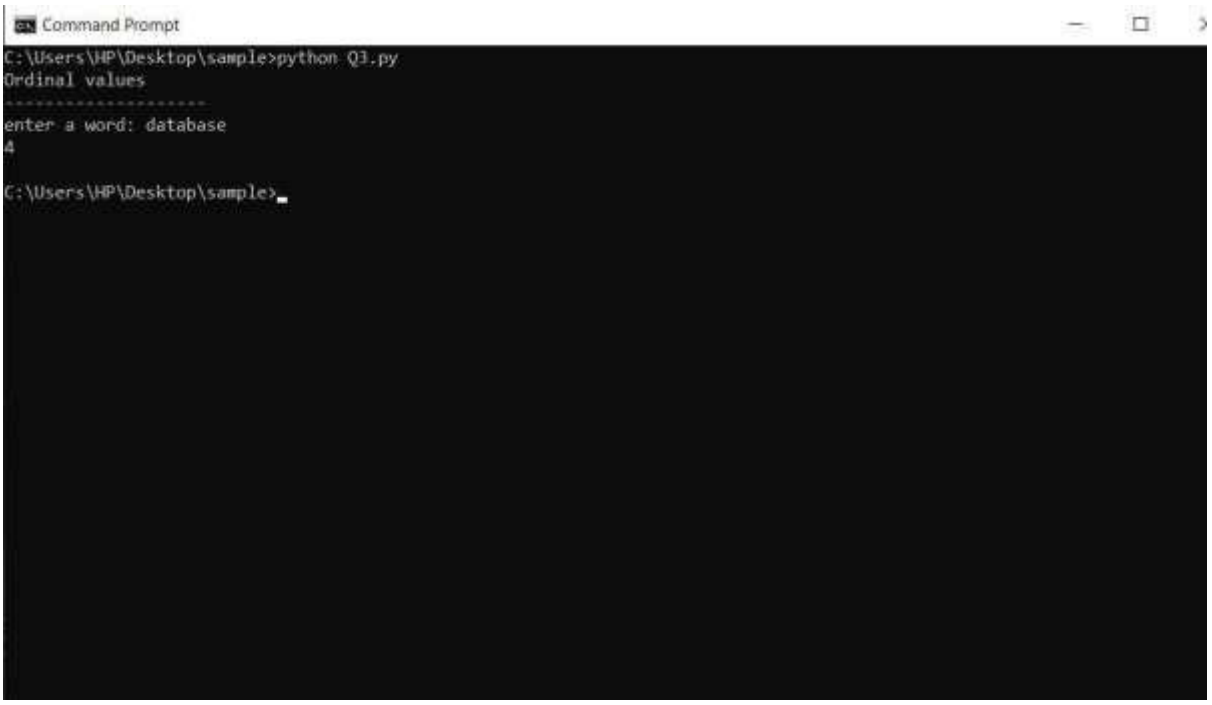


c. Form a list of vowels selected from a given word.

Program:

```
print("Ordinal values")

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

 str=input("enter a word:")

 vowels=0

for char in str :

if char in 'aeiouAEIOU' :

vowels=vowels+1

else :

continue

print(vowels)
```

Output:

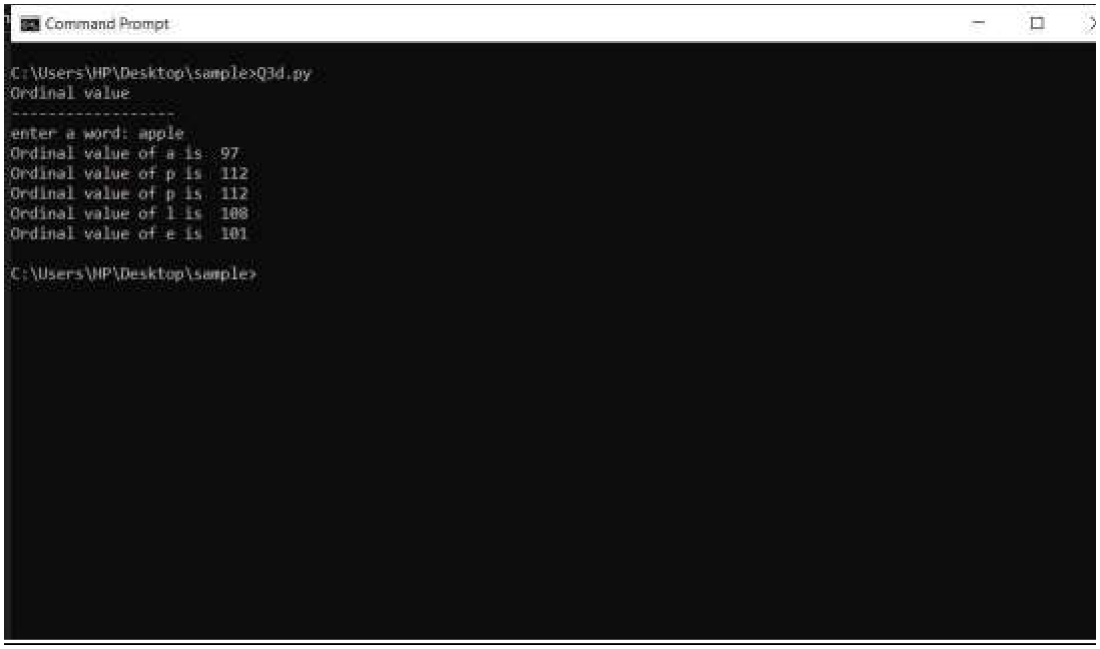


d. List of ordinal values of each element of a word

Program:

```
print("Ordinal  value")
print("..........")
word=input("enter a word: ")
for ch in word:
print("Ordinal value of "+ch+" is ",ord(ch))
```

Output :

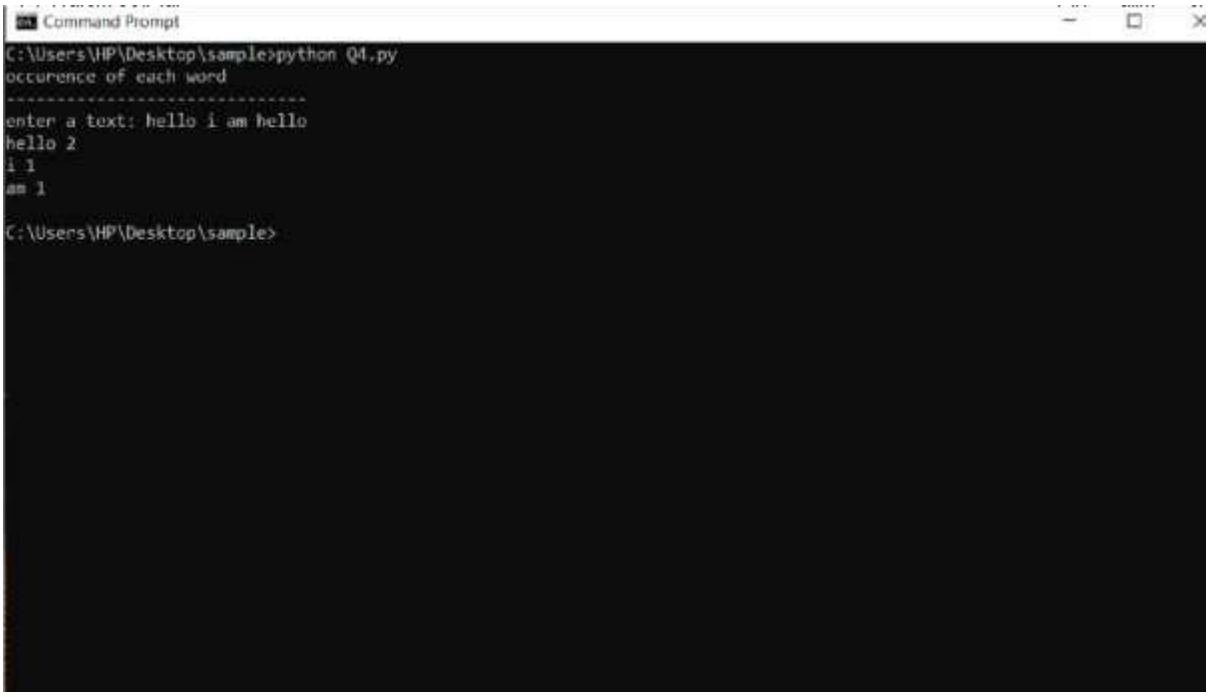


3.Count the occurrences of each word in a line of text.

Program:

```
print("occurence of eachword")
print("-----------------------------------")

str=input("enter a text: ")

counts={}

words=str.split()

for word in words :

if word in counts :

counts[word]+=1

 else :
counts[word]=1

for k,v in counts.items():

print(k,v)
```

Output:

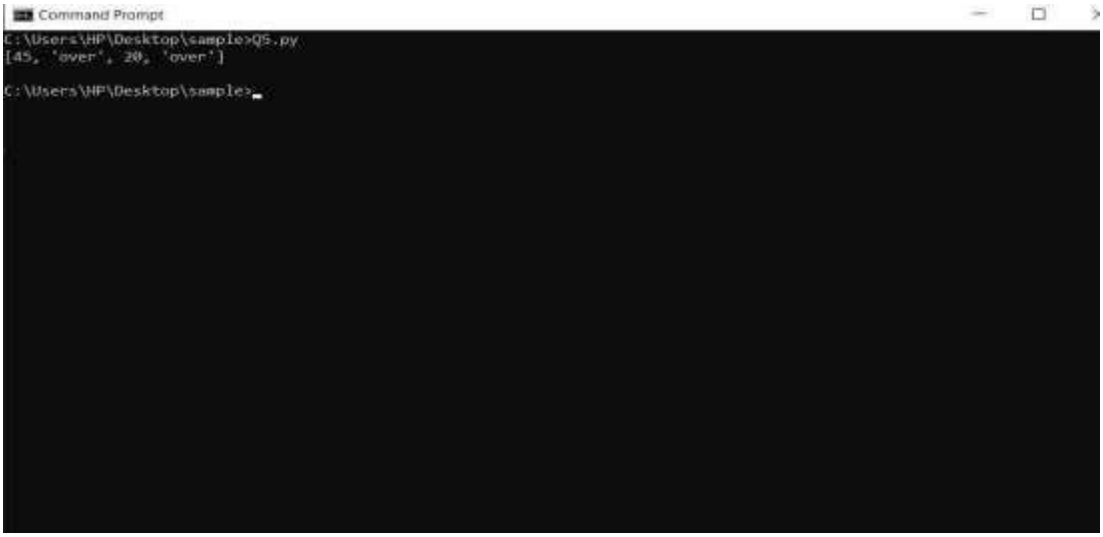


4.Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

Program:

```
list=[45,102,20,120]
new_list=[]
for i in list:
if i>100:
new_list.append("over")
else:
new_list.append(i)
 print(new_list)
```

Output:

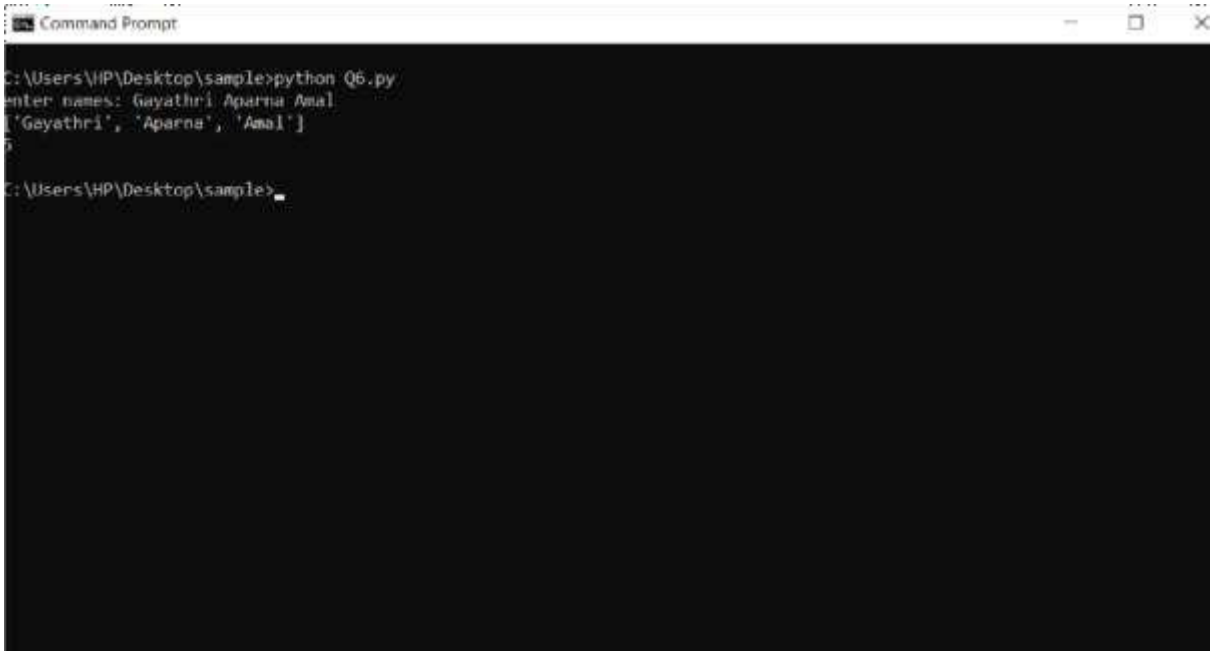


5. Store a list of first names. Count the occurrences of 'a' within the list.

Program:

```
list=input("enter names: ")

words=list.split()

print(words)

 c=0

for word in words :

for char in word :

if char in 'a' :

c=c+1

 else :

continue

print(c)
```

Output:

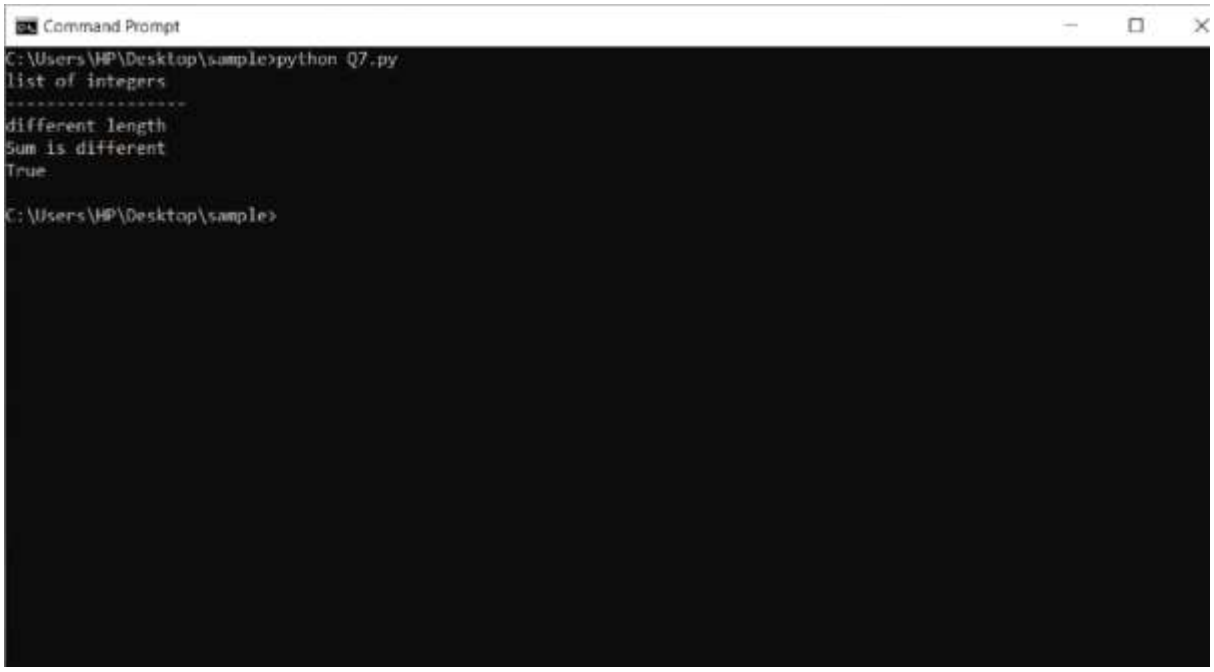


6. Enter 2 list of integers. Check (a) Whether lists are of same length (b) Whether lists sums to same value (c) Whether any value occur in both

Program:

```
print("list of integers")
print("----------")
list1=[1,23,34,26]
list2=[1,56,39,2,67]
if len(list1)==len(list2):
 print("lists are of samelength")
 else :
print("different length")
 if sum(list1)==sum(list2) :
 print("Sum is same")
else :
print("Sum isdifferent")
f=0
for elem in list2 :
if elem in list1 :
f=1
if f==1 :
print('True')
else :
print(False)
```

Output:


Command Prompt
C:\Users\HP\Desktop\sample>python Q7.py
list of integers
different length
Sum is different
True
C:\Users\HP\Desktop\sample>

7. Get a string from an input string where all occurrences of first character replaced with '$', except first character.

Program:

```
str=input("enter a string: ")
first_letter=str[0]
replace_str="$"
 new_str=str.replace(first_letter,replace_str)
print(new_str.replace(replace_str,first_letter,1))
```

Output:

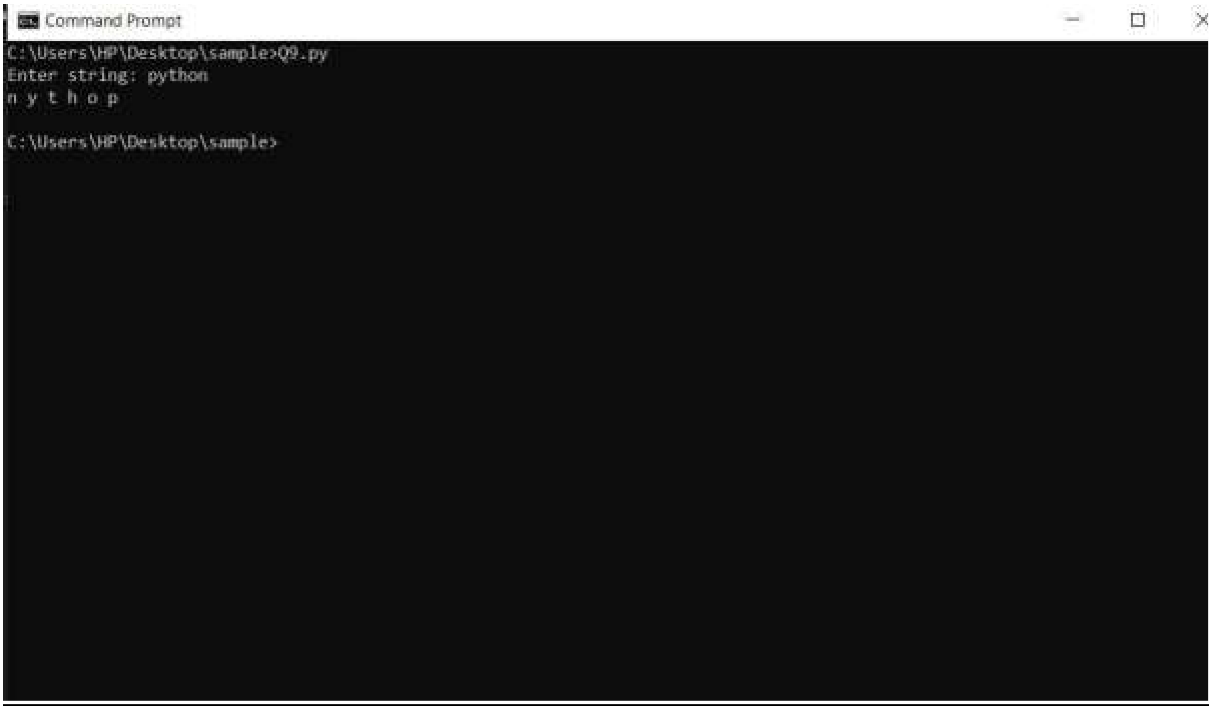
\

8.Create a string from given string where first and last characters exchanged.

Program:

```
str=input("Enter string: ")
letters=list()
for i in str:
letters.append(i)
first_letter=letters[0]
letters[0]=letters[-1]
letters[-1]=first_letter
rev_str=" "
print(rev_str.join(letters))
```

Output:

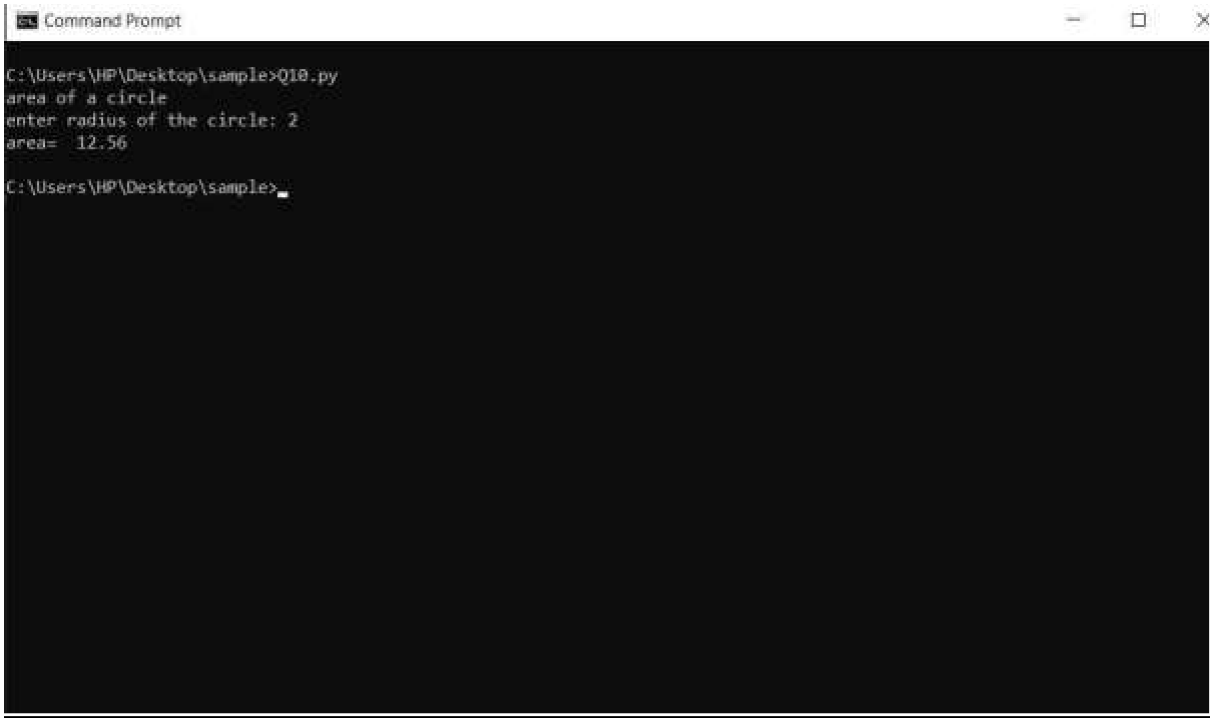


9.Accept the radius from user and find area of circle.

Program:

```
print("area of a circle")
r=float(input("enter radius of thecircle:"))
area=3.14*r*r
print("area= ",area)
```

Output:



.

10.Find the biggest of 3 numbers entered.

Program:

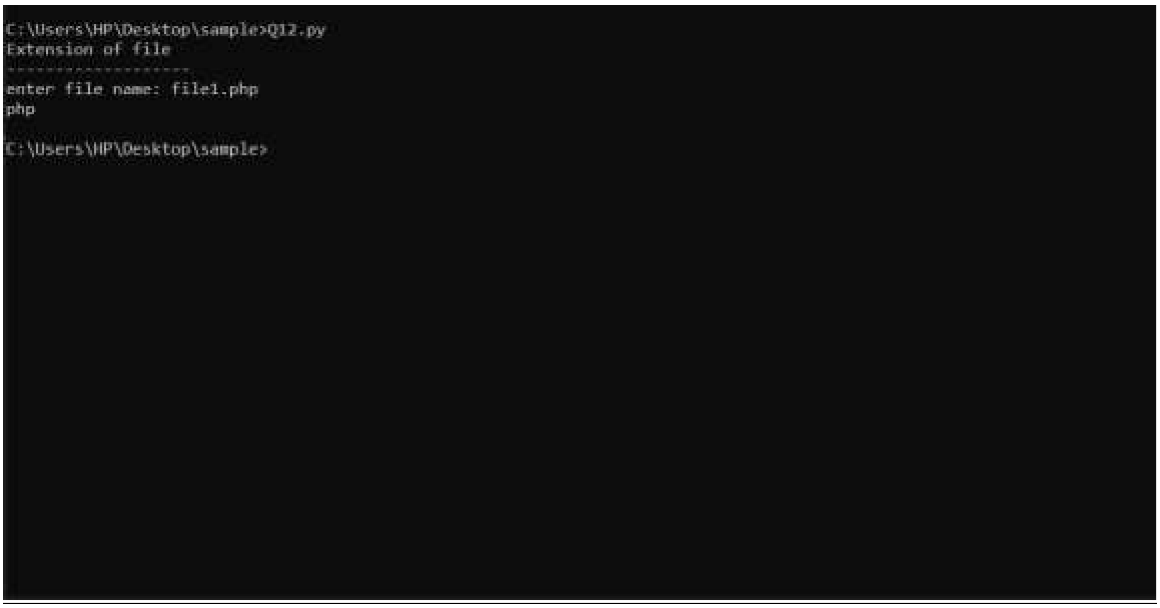
```
print("Largest of 3 numbers")
print("-----------------------------")
 n1=int(input("Enter first number: "))
n2=int(input("Enter second number: "))
n3=int(input("Enter third number: "))
if (n1>=n2) and (n1>=n3):
largest=n1
elif (n2>=n1)and(n2>=n3):
largest=n2
else :
largest=n3
print("Largest number is: ",largest)
```

Output:

11. Accept a file name from user and print extension of that. Program:

```
print("Extension of file")
print("..........")
file=input("enter file name: ")
l=list()
l=file.split(".")
print(l[-1])
```

Output:

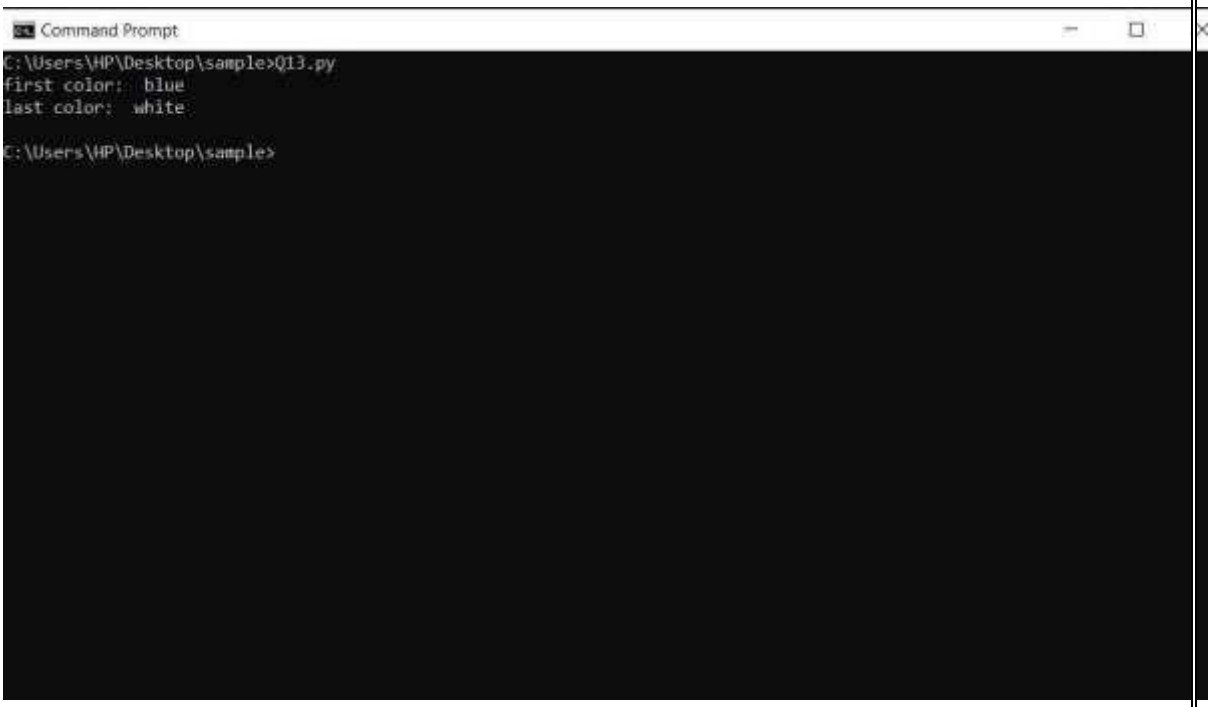


12.Create a list of colors from comma-separated color names entered by user. Display first and last colors.

Program:

```
List1=['blue','black','yellow','red','white']
print("first color",:List1[0])
print("last color: ",List1[4])
```

Output:

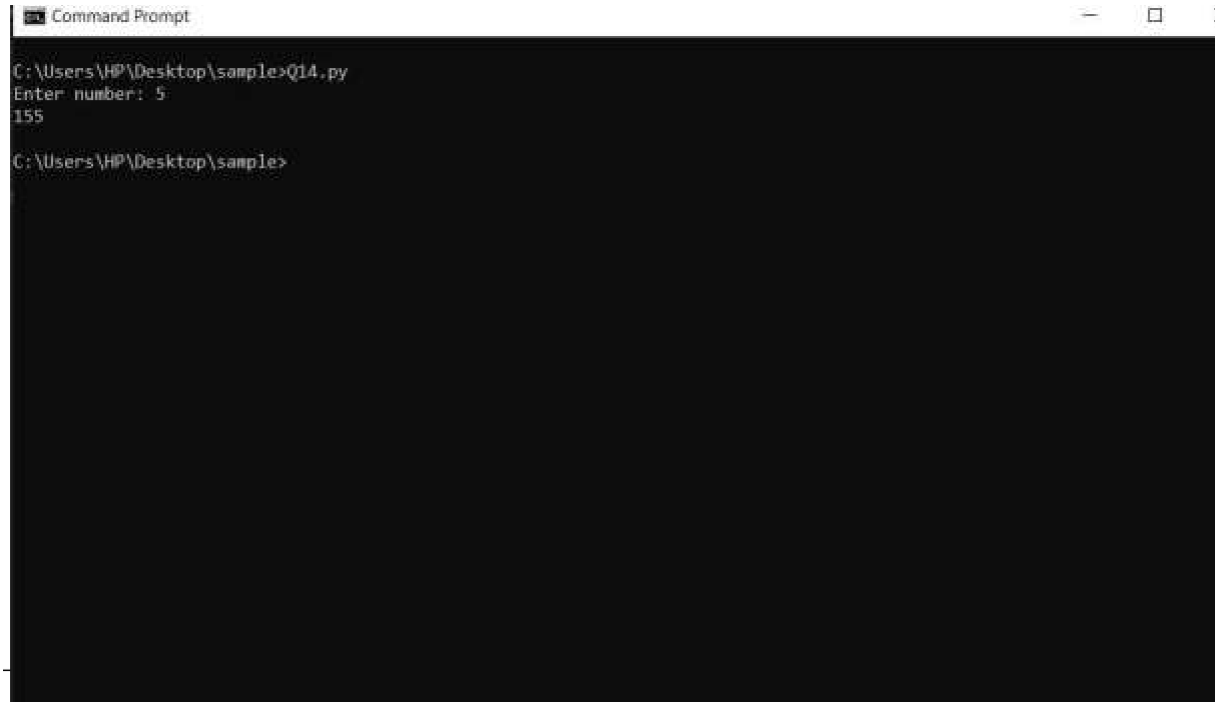


13.Accept an integer n and compute n+nn+nnn.

Program: :

```
n=int(input("Enter number: "))
num=n + n * n + n * n * n
print(num)
```

Output:



14 .Print out all colors from color-list1 not contained in color-list2.

Program:

```
color_list1=['blue','white','black','green','indigo']
color_list2=['green','red','blue','white','yellow'] sorted_list=list(set(color_list1) -
set(color_list2))
print("colors are: ",sorted_list)
```

Output:

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

Program:

```
str1=input("enter first string: ")
str2=input("enter second string: ")
new_str1=str2[:1] + str1[1:]
new_str2=str1[:1] + str2[1:]
print("After swapping: ",new_str1 + ' ' + new_str2)
```

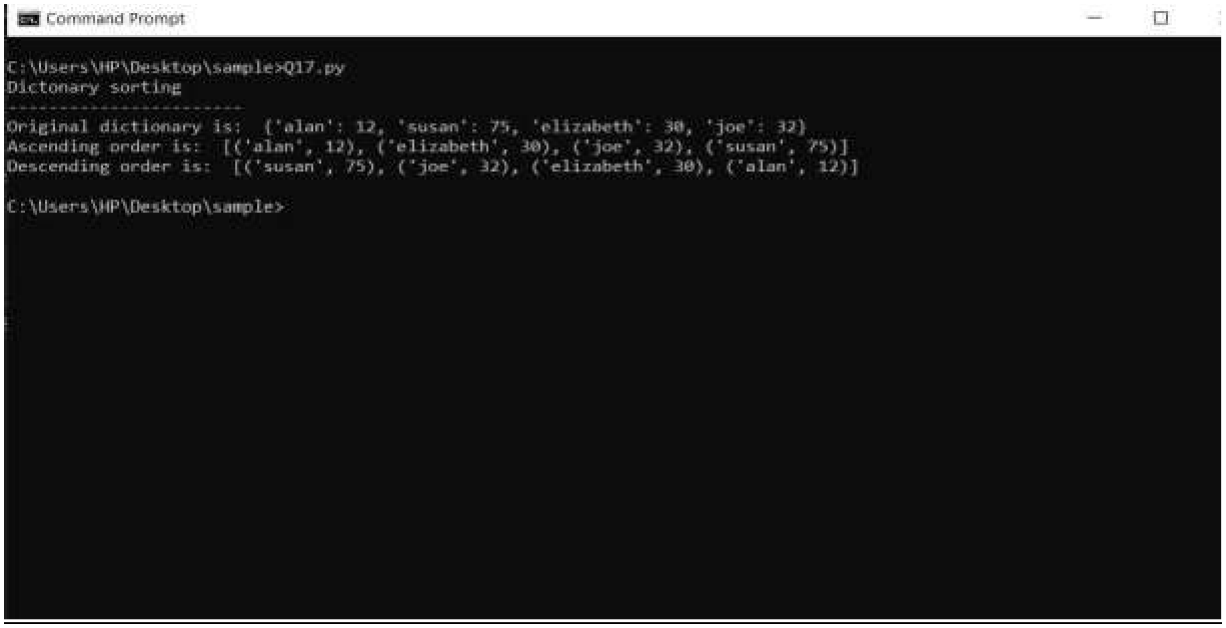
Output:

16.Sort dictionary in ascending and descending order.

Program:

```
print("Dictonary sorting")
print("-------------------------")
D={'alan':12,'susan':75,'elizabeth':30,'joe':32}
print("Original dictionary is: ",D)
l=list(D.items())
l.sort()
print("Ascending order is: ",l)
l=list(D.items())
l.sort(reverse=True)
print("Descending order is: ",l)
```

Output:

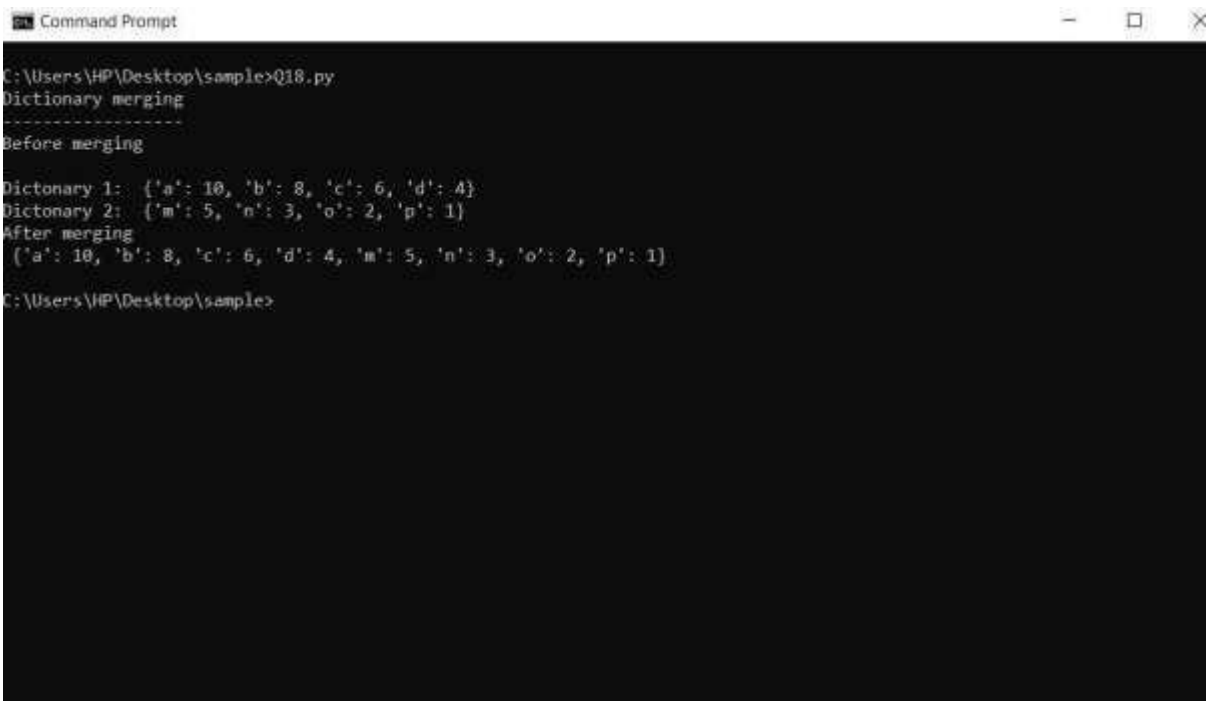


17. Merge two dictionaries

Program:

```
print("Dictionarymerging")
print("..........")
d1={'a':10,'b':8,'c':6,'d':4}
d2={'m':5,'n':3,'o':2,'p':1}
print('Before merging\n')
print("Dictonary 1: ",d1)
print("Dictonary 2: ",d2)
d1.update(d2)
print('After merging\n',d1)
```

Output:

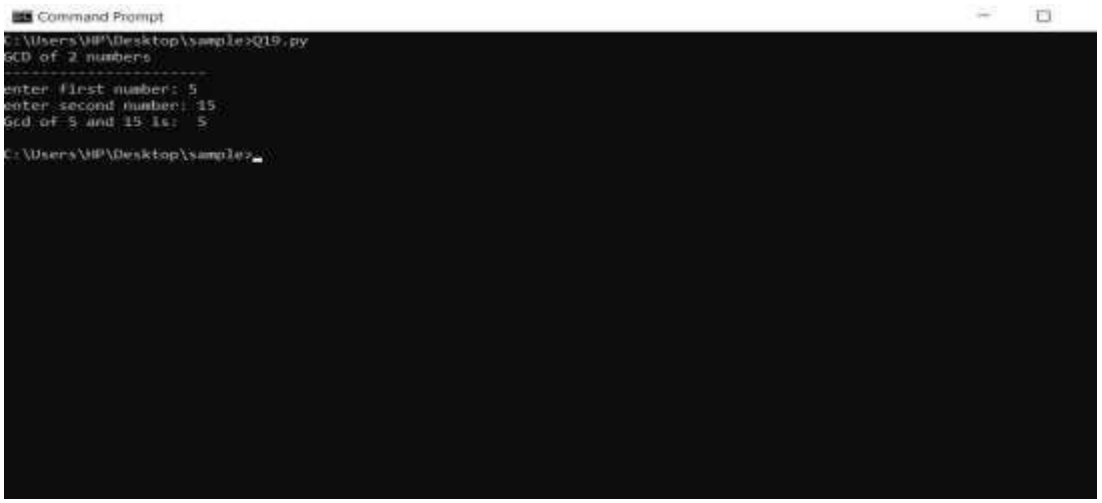


18.Find gcd of 2 numbers.

Program:

```
print("GCD of 2 numbers")
print("-----------------------------")
n1=int(input("enter first number: "))
n2=int(input("enter second number: "))
def gcd(a,b):
if(b==0):
return a else:
return gcd(b,a%b)
result=gcd(n1,n2)
print("Gcd of" ,n1, "and" ,n2, "is:",result)
```

Output:

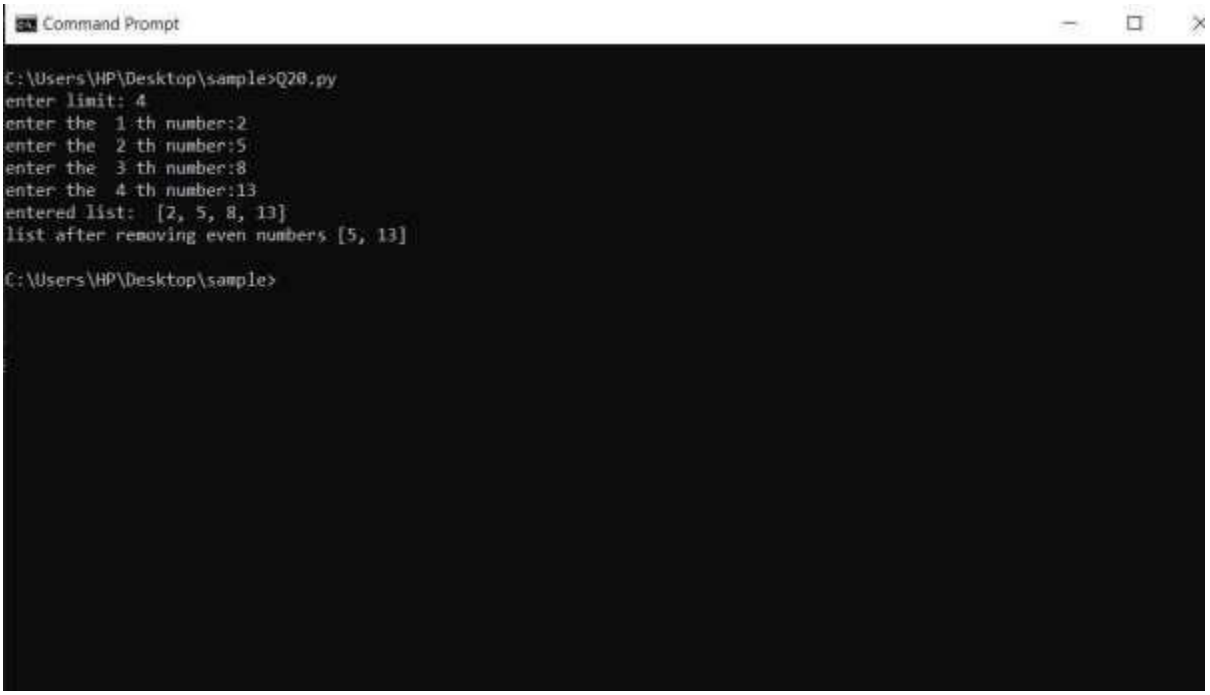


19. From a list of integers, create a list removing even numbers.

Program:

```
limit=int(input("enter limit: "))
n=[]
for i in range(1,limit+1):
num=int(input(f"enter the {i} th number:"))
n.append(num)
print("entered list: ",n)
odd_list=[]
for i in n:
if i%2!=0:
odd_list.append(i)
print("list after removing even numbers",odd_list)
```

Output:



20. Program to find the factorial of a number

Program:

```
print("factorial of a number")
print("-----------------")
num=int(input("enter a number: "))
fact=1
if num < 0:
print("enter a positive number")
else:
for i in range(1,num+1):
fact=fact*i
print("factorial of",num,"is",fact)
```

Output:

21. Generate Fibonacci series of N terms

Program:

```
print("fibonacci series")
print("..........")
limit=int(input("enter limit: "))
n1=0
n2=1
count=1
while count < limit:
print(n1)
n=n1+n2
n1=n2
n2=n
count=count+1
```

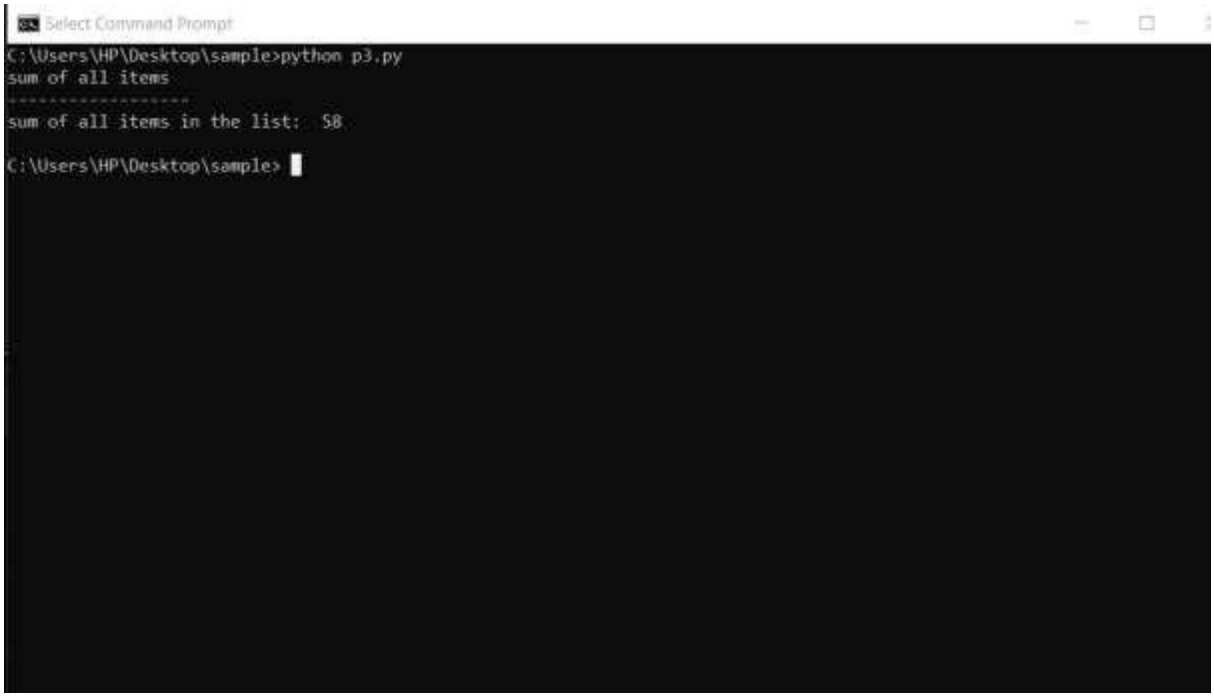
Output:

22. Find the sum of all items in a list

Program:

```
print("sum of all items")
print("..........")
total=0 list1=[11,10,12,20,5]
for ele in range(0,len(list1)):
 total = total + list1[ele]
print("sum of all items in the list: ",total)
```

Output:



23. Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

Program:

```
sq_list=[]
limit=int(input("enter the range: "))
if(limit<1000 or limit>9999):
print("enter a range between 1000 to 9999")
else:
for i in range(32,99):
 s=0
if(i*i>limit):
break
else:
for k in str(i*i):
if(int(k)%2==0):
 s=s+1
if(s==4):
sq_list.append(i*i)
if(len(sq_list)==0):
print("No numbers satisfying both conditions found in the range")

else:

print(f"Numbers satisfying both conditons are->{sq_list}")
```

Output:

24. Display the given pyramid with step number accepted from user.
Eg: N=4
1
2 4
3 6 9
4 8 12 16

Program:

```
sum=0

limit=int(input("enter limit:"))
for i in range(1,limit+1):
print("\n")
for j in range(1,i+1):
sum=i*j
print(sum,end=' ')
print("\n")
```

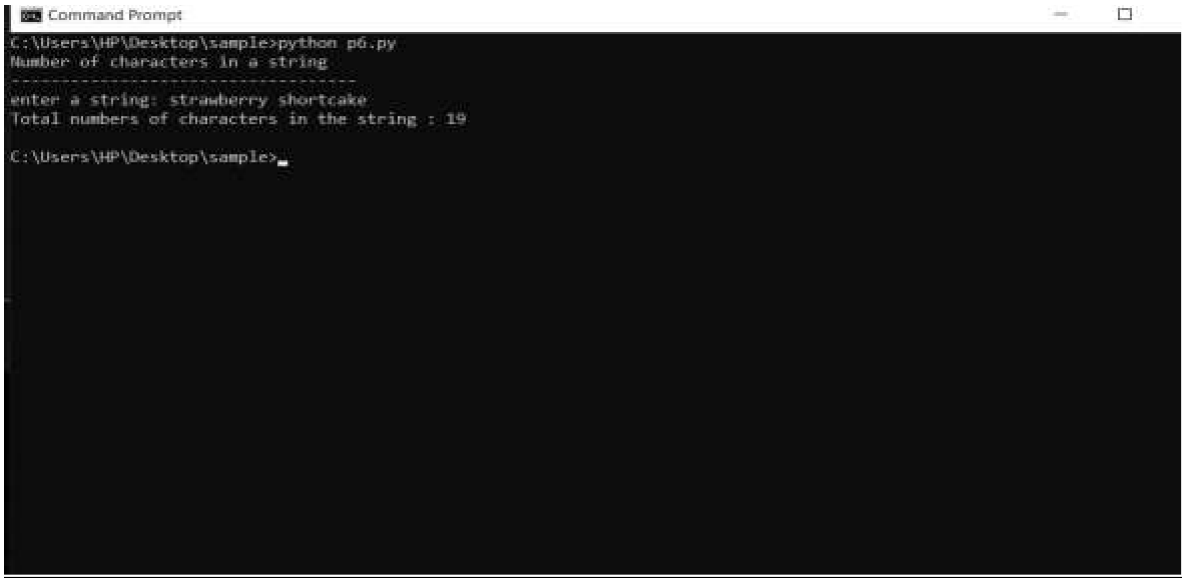
Output :

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

Program:

```
print("Number of characters in a string")
print("-----------------------------------------")
string=input("enter a string: ")
count=0
for i in range(0,len(string)):
if(string[i]!=' '):
count = count + 1
print("Total numbers of characters in the string : "+str(count))
```

Output:



26. Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.

Program:

```
string1=input("enter a string: ")
 str1="ly"
str2="ing"
last=string1[-3:]
if last in 'ing':
string1=string1+str1
print("string is: "+string1)
else:
string1=string1+str2
print("string is: "+string1)
```

Output:

27. Accept a list of words and return length of longest word.

Program:

```
str_list=list() long=0
string=' '
lim=int(input("enter the limit: "))
for i in range(1,lim+1):
item=str(input(f"enter the string{i}:"))
str_list.append(item)
for i in str_list:
if(long<=len(i)):
long=len(i)
string=i
print(f"Longest word in the list is {string} and its length is {long}")
```

Output:

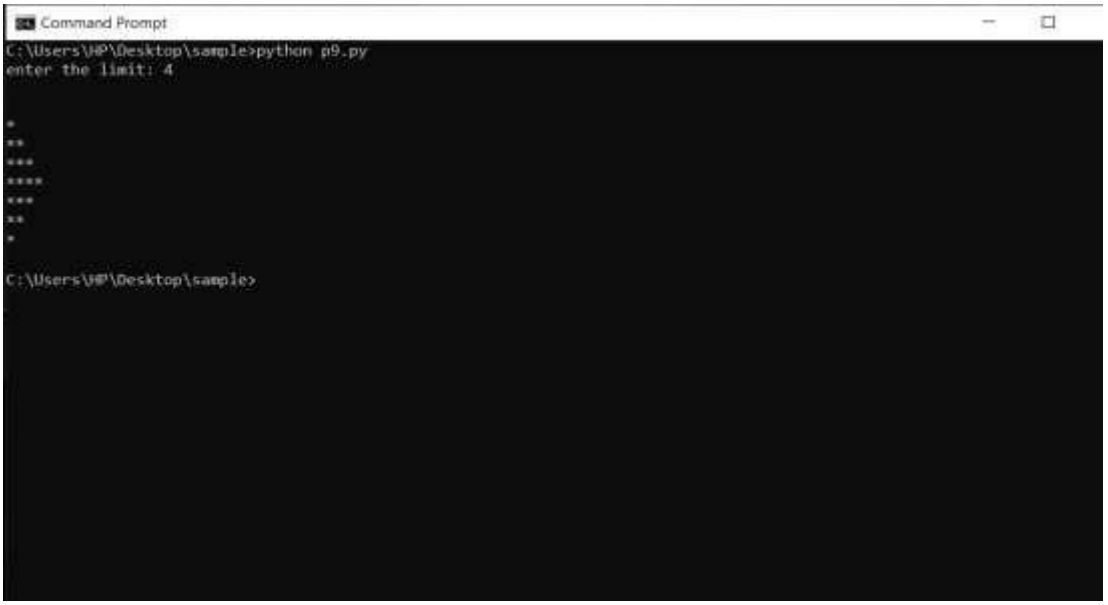
28. Construct following pattern using nested loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Program:

```
lim=int(input("enter the limit: "))
print("\n")
for i in range(1,lim+1):
print('*'*i)
 j=lim-1
while(j!=0):
print('*'*j)
j=j-1
```

Output:

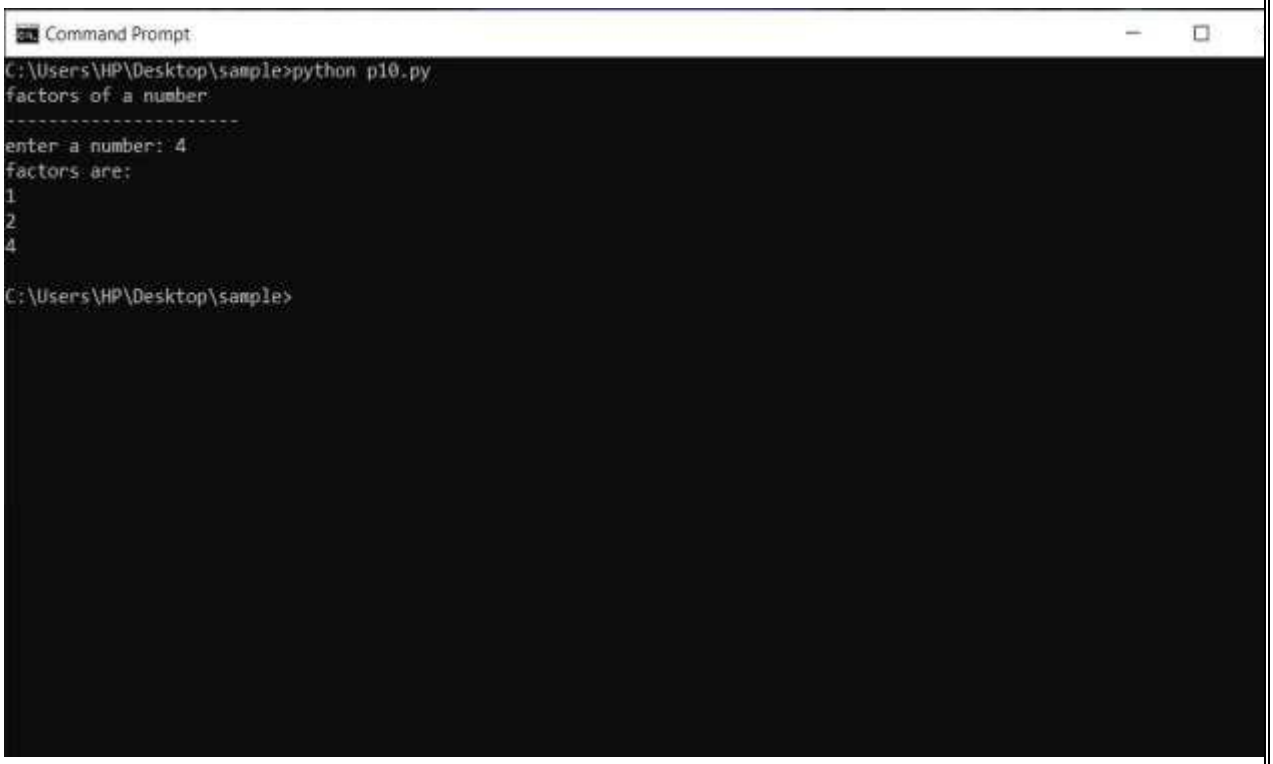


29.Generate all factors of a number.

Program:

```
print("factors of a number")
print("----------------------")
num=int(input("enter a number: "))
print("factors are: ")
 for i in range(1,num+1):
if num % i ==0:
print(i)
```

Output:



30. Write lambda functions to find area of square, rectangle and triangle.

Program:

```
square=lambda x: x ** 2

rectangle=lambda x,y: x*y

triangle=lambda x,y: 0.5*(x*y)

print("1.Area of square")

print("2.Area of rectangle")

print("3.Area of triangle")

print("\n")

ch=int(input("enter a choice: "))
if(ch==1):
side = int(input("enter one side: "))
 print("\n")
print(f"Area of the square is {square(side)}")
elif(ch==2):
length=int(input("enter the length: "))
breadth=int(input("enter the breadth: "))
print("\n")
print(f"Area of the rectangle is{rectangle(length,breadth)}")
elif(ch==3):
height=int(input("enter the height: "))
breadth=int(input("enter the breadth: "))
print("\n")
print(f"Area of triangle is{int(triangle(height,breadth))}")

else:

print("Invalid input")
```

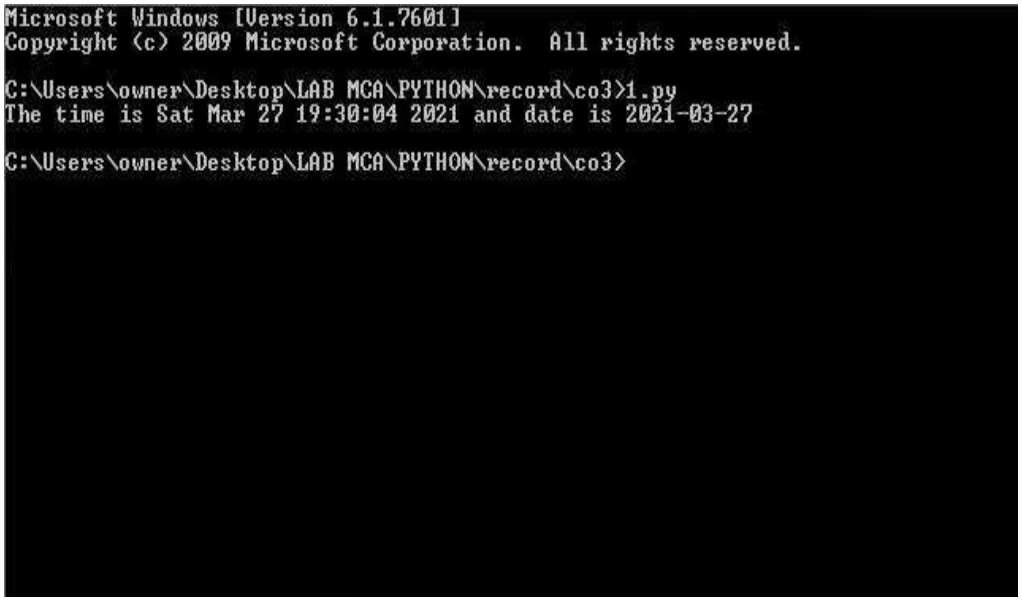
Output:

31. Work with built-in packages

Program:

```
import time

import datetime

today=datetime.date.today()

print(f"The time is {time.ctime()} and date is {today}")
```

Output:



32. Create a package graphics with modules rectangle,circle.include method to find area and perimeter of respective figures in each.Write a program to find area and perimeter of figure by different importing statements

Program:

```
from graphics import rectangle as r, circle as c

from graphics.three_d_graphics import sphere as s, cuboid as cu

print(f"Area of rectangle :{r.area(12,12)}")

print(f"Area of circle :{c.area(6)}")

print(f"Area of sphere :{s.area(12)}")

print(f"Area of cuboid :{cu.area(12,16,18)}")

print(f"Perimeter of rectangle :{r.perimeter(12,12)}")

print(f"Perimeter of circle :{c.perimeter(6)}")

print(f"Diameter of sphere :{s.diameter(12)}")

print(f"Periameter of cuboid :{cu.perimeter(12,16,18)}")
```

Output:

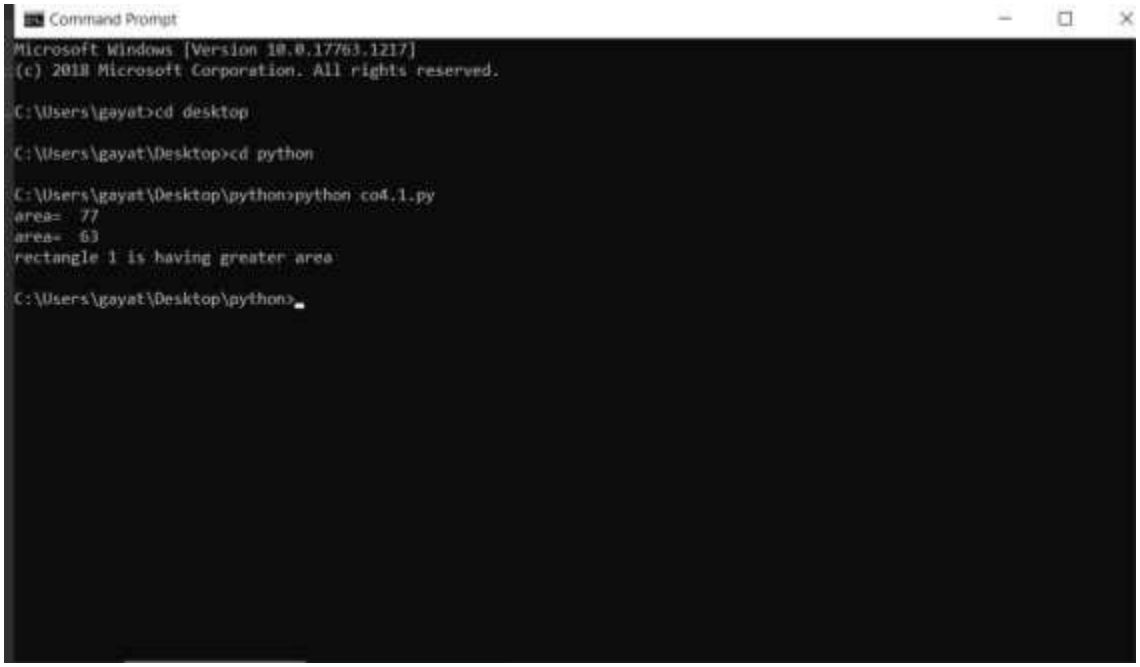
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\owner\Desktop\LAB MCA\PYTHON\record\co3>2.py
Area of rectangle :144
Area of circle :36
Area of sphere :1808.6399999999999
Area of cuboid :95551488
Perimeter of rectangle :48
Perimeter of circle :37.68
Diameter of sphere :24
Periameter of cuboid :184

C:\Users\owner\Desktop\LAB MCA\PYTHON\record\co3>
```

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

Program:

```
class Rectangle:
 def arearect(self,l,w):
      self.l=l
      self.w=w
      self.area=self.l*self.w
      print("area= ",self.area)
rect1=Rectangle()
rect2=Rectangle()
rect1.arearect(11,7)
rect2.arearect(9,7)
if(rect1.area<rect2.area):
       print("rectangle 2 is having greater area")
elif(rect1.area==rect2.area):
       print("both rectangles have some area")
else:
       print("rectangle 1 is having greater area")
```

Output:

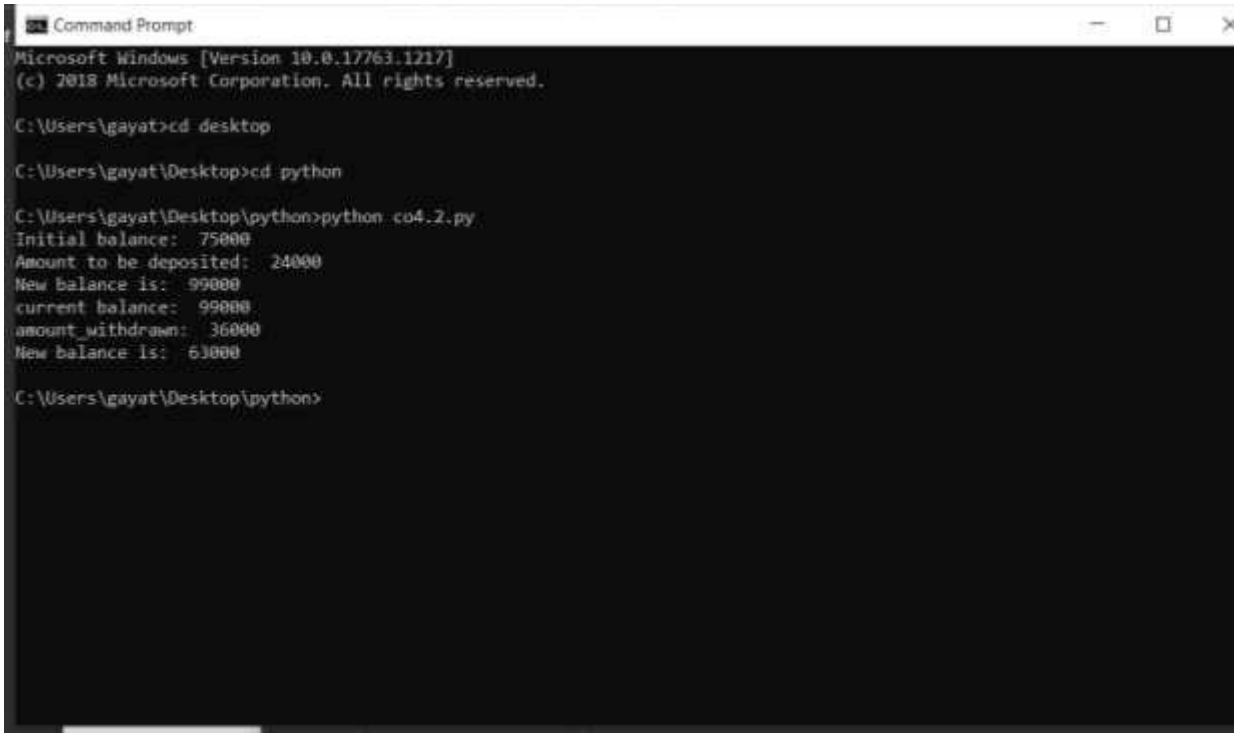


34. Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

Program:

```
class Bank_account:
        def init (self,acc_no,name,acc_type,balance):
                self.acc_no=acc_no
                self.name=name
                self.acc_type=acc_type
                self.balance=balance
        def deposit(self,deposit_am):
                print("Initial balance: ",self.balance)
                print("Amount to be deposited: ",deposit_am)
                self.balance=self.balance+deposit_am
                print("New balance is: ",self.balance)
        def withdraw(self,withdrawn_am):
                print("current balance: ",self.balance)
                print("amount_withdrawn: ",withdrawn_am)
                self.balance=self.balance-withdrawn_am
                print("New balance is: ",self.balance)
P=Bank_account(1234,'Rose','savings',75000)
P.deposit(24000)
P.withdraw(36000)
```

Output:



35. Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

Program:

```
class Rectangle:
def init (self,l,w):
                self.l=l
                self.w=w
def lt (self,a):
                if((self.l*self.w)>(a.l*a.w)):
                        print("rect1 is having greater area")
                        return(self.l*self.w)

else:

                        print("rect2 is having greater area")
                        return(a.l*a.w)
rect1=Rectangle(9,4)
rect2=Rectangle(8,5)
print(rect1<rect2)
```

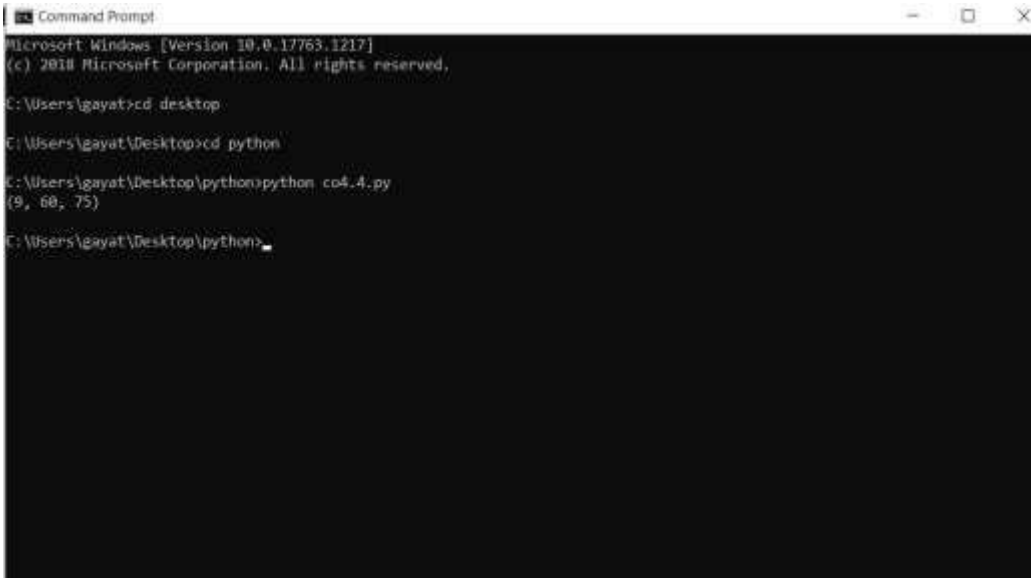
Output:

36 .Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

Program:

```
class Time:
    def init (self,hr,min,sec):
            self.hr=hr
            self.min=min
            self.sec=sec
    def__add__(self,t):
            return(self.hr+t.hr,self.min+t.min,self.sec+t.sec)
    t1=Time(4,15,45)
    t2=Time(5,45,30)
    print(t1+t2)
```

Output:

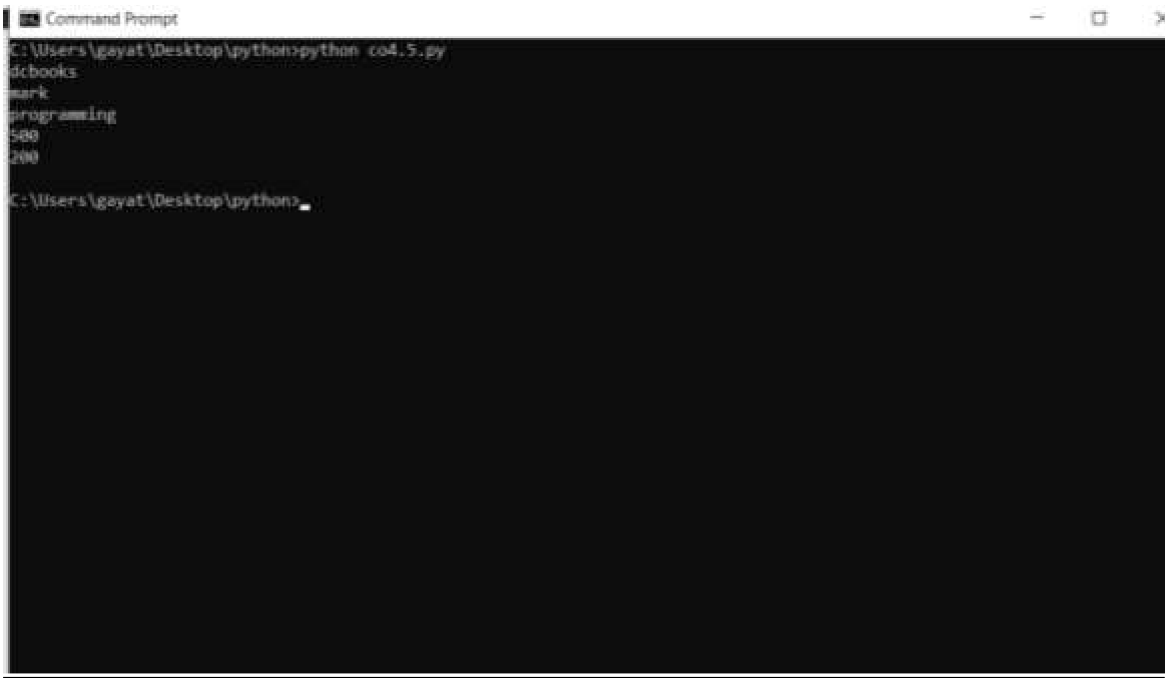


37. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Program:

```
class Publisher:
        def init (self,n):
                self.name=n
class Book(Publisher):
        def init (self,n,a,t):
                super(). init (n)
                self.title=t
                self.author=a
class Python(Book):
        def init (self,n,a,t,p,pg):
                super(). init (n,a,t)
                self.price=p
                self.pages=pg
        def Print(self):
                print(P.name)
                print(P.title)
                print(P.author)
                print(P.price)
                print(P.pages)
P=Python('dcbooks','programming','mark',500,200)
P.Print()
```

Output:

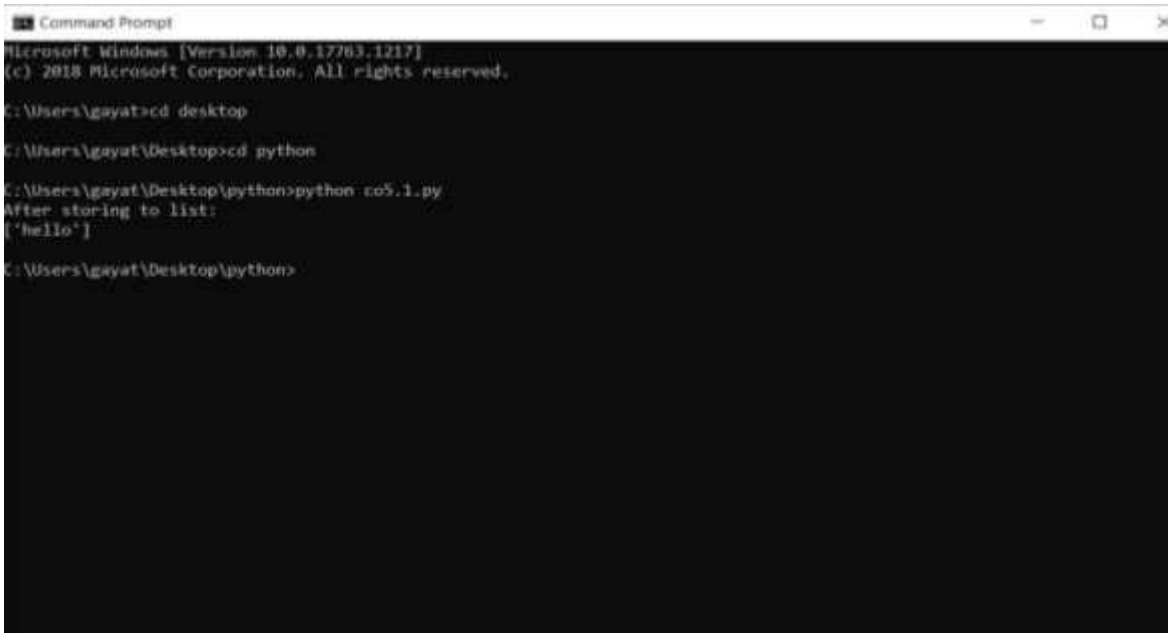


38. Write a Python program to read a file line by line and store it into a list

Program:

```
file1=open("demofile1.txt",'r')
list1=file1.readlines()
print(f"After storing to list:\n{list1}")
```

Output:



39. Python program to copy odd lines of one file to other

Program:

```
file1=open("demofile1.txt",'r')
file2=open("demofile2.txt",'w+')
list1=file1.readlines()
for i in range(0,len(list1),2):
file2.write(list1[i])
print("odd lines copied to another file")
```

Output: