FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB

LABORATORY RECORD

Name: AKHILADISON
Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 09**

University Registration Number: FIT21MCA-2009

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done by ***AKHILA DISON (FIT21MCA-2009)*** *in the* ***20MCA131 PROGRAMMING LAB*** *Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge
Name:

Signature of H O D
Name:

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

CONTENT

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

27	**9/12/2021**	Accept a list of words and return length of longest word.	**27**	
28	**9/12/2021**	Construct following pattern using nested loop. *	**28**	
29	**9/12/2021**	Generate all factors of a number.	**29**	
COURSE OUTCOME 3				
30	**29/01/2022**	Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)	**30**	
COURSE OUTCOME 4				
31	**13/01/2022**	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	**33**	
32	**13/01/2022**	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.	**35**	

33	**13/01/2022**	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	**36**	
34	**20/01/2022**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	**37**	
35	**20/01/2022**	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.	**38**	
COURSE OUTCOME 5				
36	**03/02/2022**	Write a Python program to read a file line by line and store it into a list.	**40**	
37	**03/02/2022**	Write a Python program to read each row from a given csv file and print a list of strings.	**41**	

COURSE OUTCOME 1

Sl no.:-1

Aim:-Display future leap years from current leap to a finial year entered by the users.

Program code:-

```
currentYear=int(input("Enter the current  year:"))
futureyear=int(input("Enter the future Year: "))
print("Leap Years in between the",currentYear,"and",futureyear)
for i in range(currentYear,futureyear):
   if i%4==0:
      print(i)
```

Output:-

Sl no:-2

Aim:- List comprehension:-

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

b. square of N numbers.

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

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

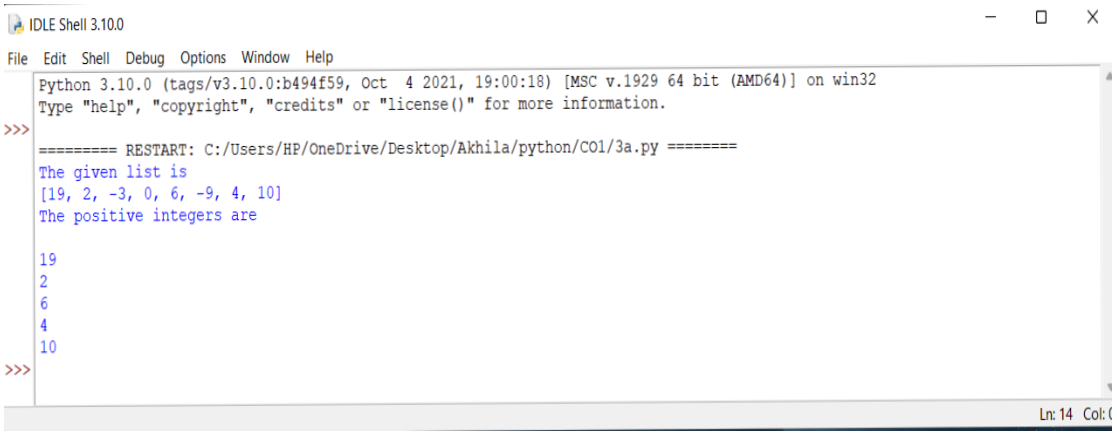
a.

Program code:-

```
print("The given list is ")
list=[19,2,-3,0,6,-9,4,10]
print(list)
print("The positive integers are \n ")
for i in list:
```

```
        if i>0:
            print(i)
```

Output:-



b.

Program code:-

```
        print("The given list is ")
        list=[19,2,-3,0,6,-9,4,10]
        print(list)
        print("The square's of given list is  ")
        for i in list:
            sqr=i*i
            print(sqr)
```

Output:-

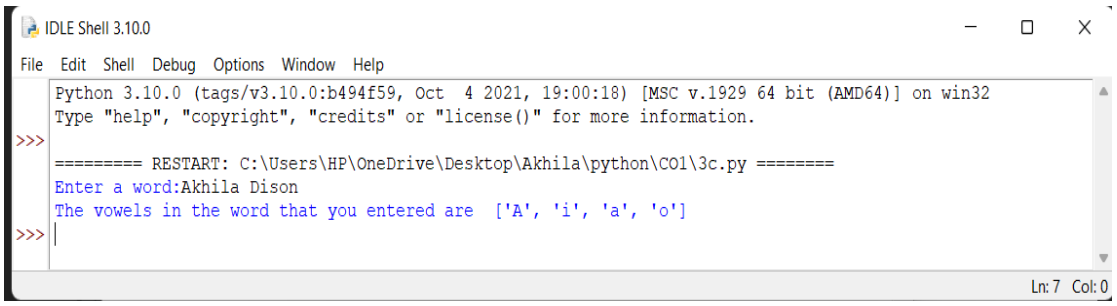
c.

Program code:-

```
        word=input("Enter a word:")
        list=['A','E','I','O','U','a','e','i','o','u']
        temp=[]
        for i in word:
```

```
        if (i in list and i not in temp):
            temp.append(i)
    print("The vowels in the word that you entered are ",temp)
```

Output:-

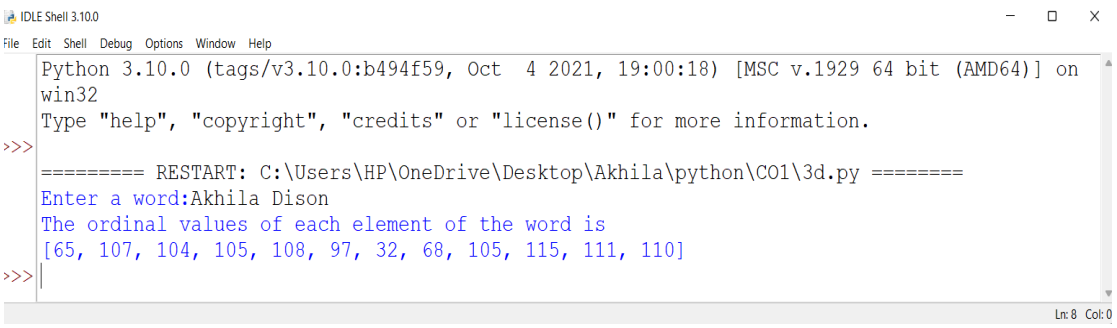


d.

Program code:-

```
    word=input("Enter a word:")
    temp=[]
    print("The ordinal values of each element of the word is ")
    for i in word:
        x=ord(i)
        temp.append(x)
    print(temp)
```

Output:-



Sl no.:-3

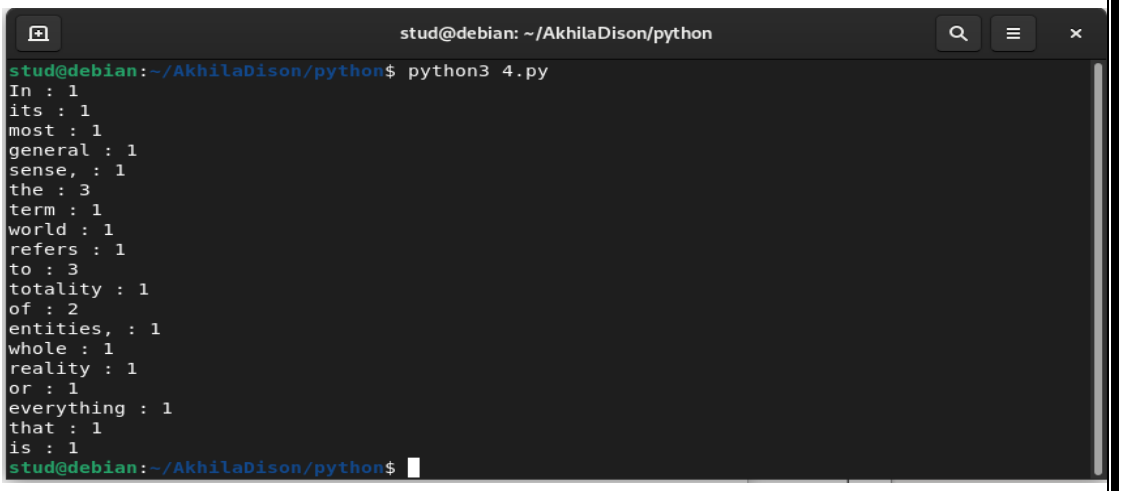
Aim:-Count the occurrence of each word in a line of text.

Program code:-

```
    k="In its most general sense, the term world refers to the totality
    of entities, to the whole of reality or to everything that is"
    h=[]
    for i in k.split(" "):
            if i not in h:
                    h.append(i)
```

```
    for i in h:
        print(i,":",k.split(" ").count(i))
```

Output:-



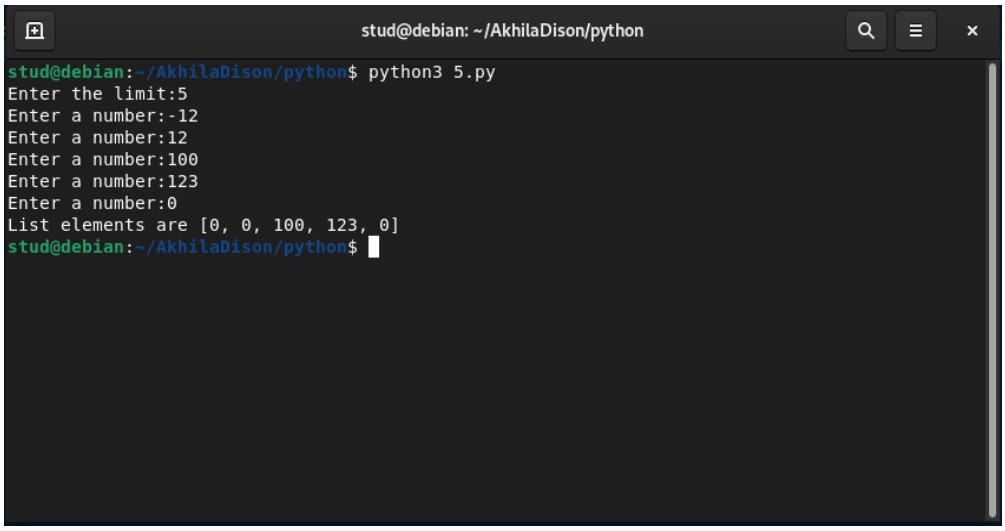
Sl no.:-4

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

Program code:-

```
integers=int(input("Enter the limit:"))
temp=[]
over=0
for i in range(0,integers):
    user=int(input("Enter a number:"))
    if(user>=100):
        temp.append(user)
    else:
        temp.append(over)
print("List elements are",temp)
```

Output:-



Sl no.:-5

Aim:-store a list of first names.count the occurence of “a” within the list.

Program code:-

```
list= ['akhila','ashik','mary']
count=0
for i in list:
  value= i.count('a')
  count=count+value
print(list)
print(count)
```

Output:-

Sl no.:- 6

Aim:- Enter 2list of first integers.Check

a.whether list are of same length.

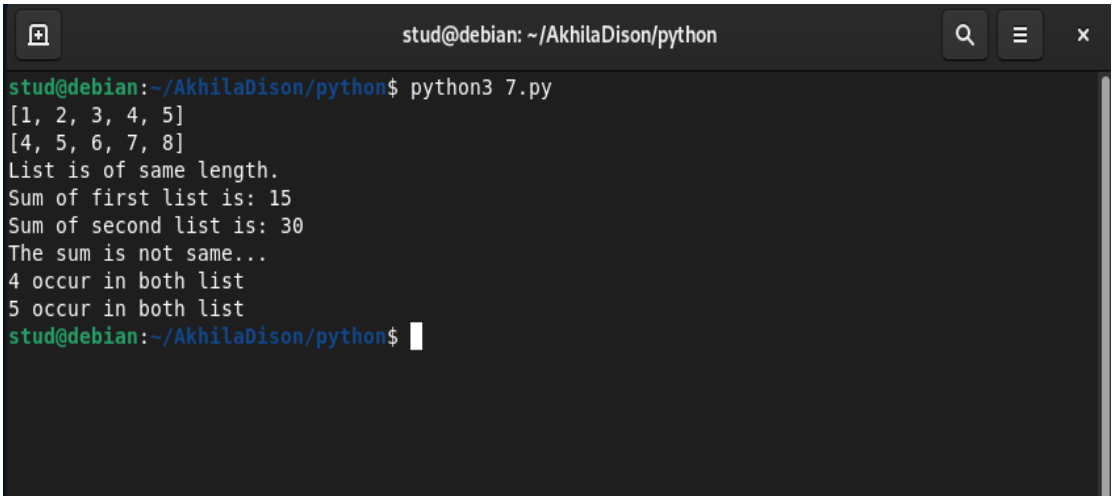
b.Whether list sums to same value.

b.whether any value occur in both.

Program code:-

```
#whether list are of same length.
list1=[1,2,3,4,5]
list2=[4,5,6,7,8]
print(list1)
print(list2)
x=len(list1)
y=len(list2)
if(x==y):
 print("List is of same length.")
else:
 print("List is not same length.")

#Whether list sums to same value.
sum1=0
sum2=0
for i in list1:
        sum1=sum1+i
print("Sum of first list is:",sum1)
for j in list2:
        sum2=sum2+j
print("Sum of second list is:",sum2)
if(sum1==sum2):
        print("The sum is same....")
else:
        print("The sum is not same...")

 #whether any value occur in both.
for i in list1:
        if i in list2:
                print(i,"occur in both list")
```

Output:-



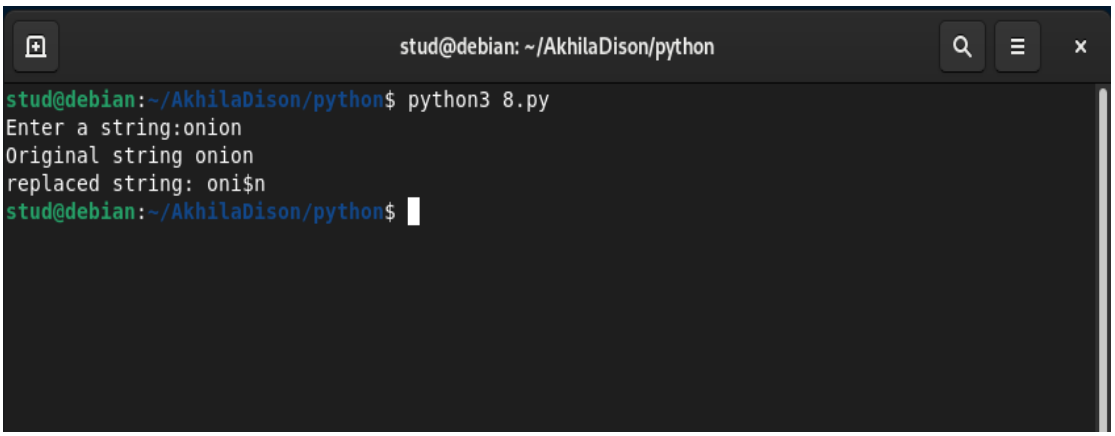
Sl no:-7

Aim:-Get a stringfrom an input string where all occurrence of first charater replace with '$',except first charater.
[eg:-onion->oni$n]

Programe code:-

```
s=input("Enter a string:")
print("Original string",s)
char=s[0]
s=s.replace(char,'$')
s=char+s[1:]
print("replaced string:",s)
```

Output:-



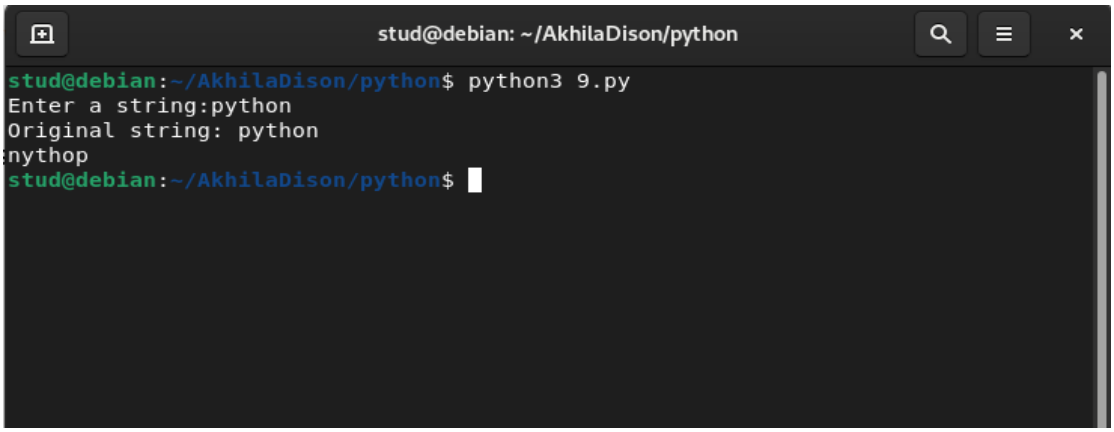
Sl no.:-8

Aim:-Create a string from give string where first and last charater exchanged.[eg:-python->nythop]

Program Code:-

```
s=input("Enter a string:")
print("Original string:",s)
sf=s[0]
sl=s[-1]
n=len(s)
ns=sl+s[1:n-1]+sf
print(ns)
```

Output:-



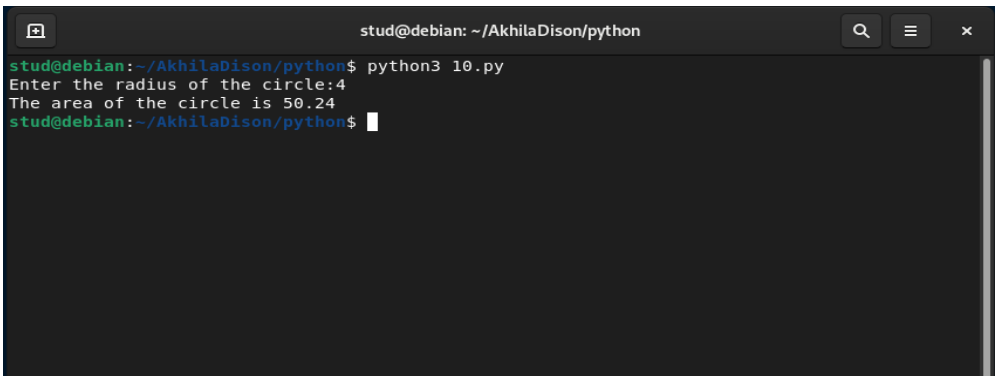
Sl no.:-9

Aim:-Accept the radius from the user and find the area of circle.

Program Code:-

```
y=int(input("Enter the radius of the circle:"))
a=3.14*y*y
print("The area of the circle is",a)
```

Output:-



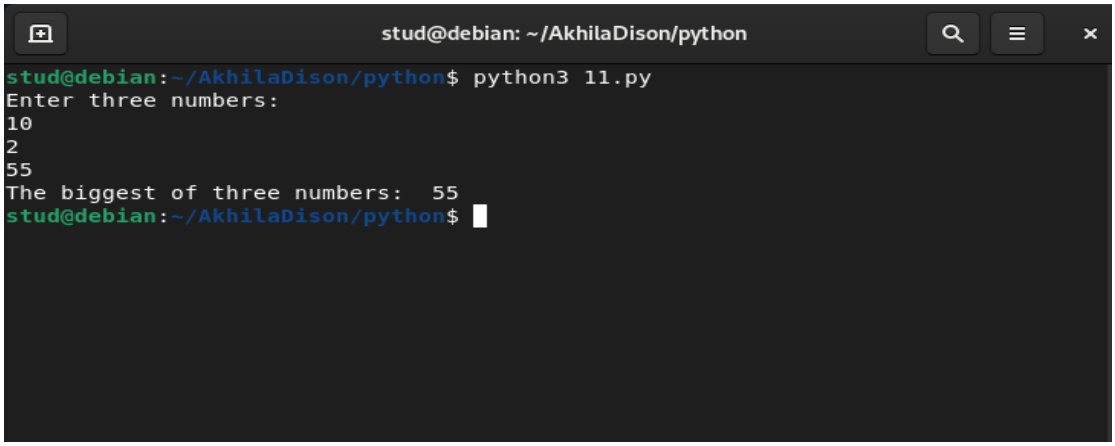
Sl no.:-10

Aim:-find the greatest of 3 entered.

Program code:-

```
print("Enter three numbers: ")
a=int(input())
b=int(input())
c=int(input())
if a>b and a>c:
        print("The biggest of three numbers: ",a)
if b>a and b>c:
        print("The biggest of three numbers: ",b)
if c>a and c>b:
        print("The biggest of three numbers: ",c)
```

Output:-



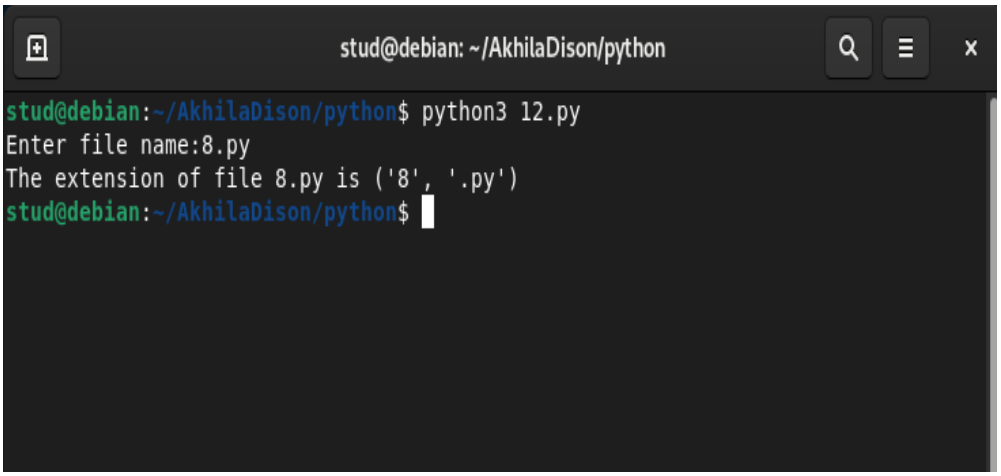
Sl no.:-11

Aim:-Acept file name from user and print extention of that.

Program code:-

```
import os
a=input("Enter file name:")
print("The extension of file",a,"is",os.path.splitext(a))
```

Output:-



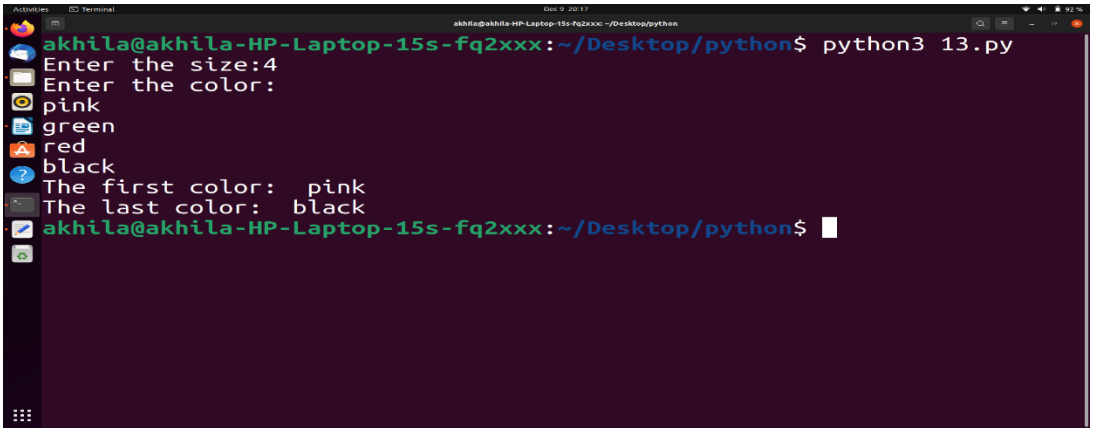
Sl no.:-12

Aim:-Create a list of colours from comma seperated color names entered by users.Display first and last color.

Program code:-

```
n=int(input("Enter the size:"))
lt = []
l=[]
print("Enter the color:")
for i in range(0,n):
        lt = input()
        l.append(lt)
print("The first color: ",l[0])
print("The last color: ",l[n-1])
```

Output:-



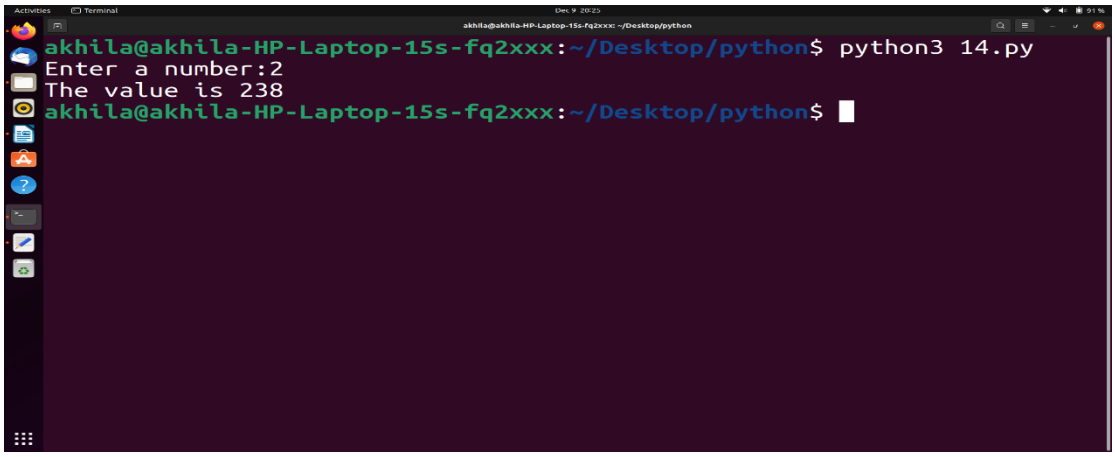
Sl no:-13

Aim:-Accept an integer and compute n+nn+nnn.

Program code:-

```
i=int(input("Enter a number:"))
value=i+((i*10)+i)+((i*100)+(i+10)+i)
print("The value is",value)
```

Output:-



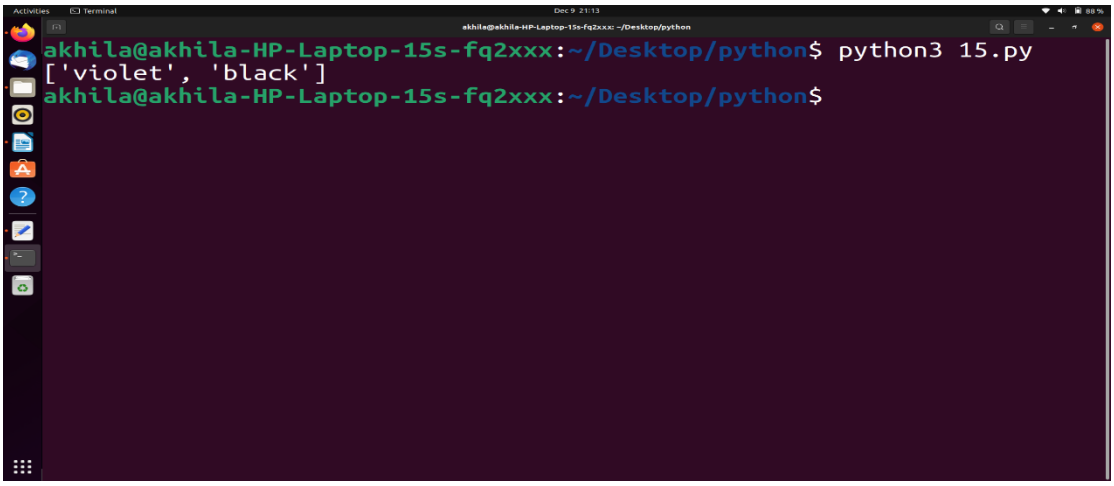
Sl no.:-14

Aim:-Print out all colors from color-list 1 not contained in color-list2

Program code:-

```
list1= ['red', 'violet', 'black']
list2= ['blue', 'pink', 'red']
list3=[]
for i in list1:
        if i not in list2:
                list3.append(i)
print(list3)
```

Output:-



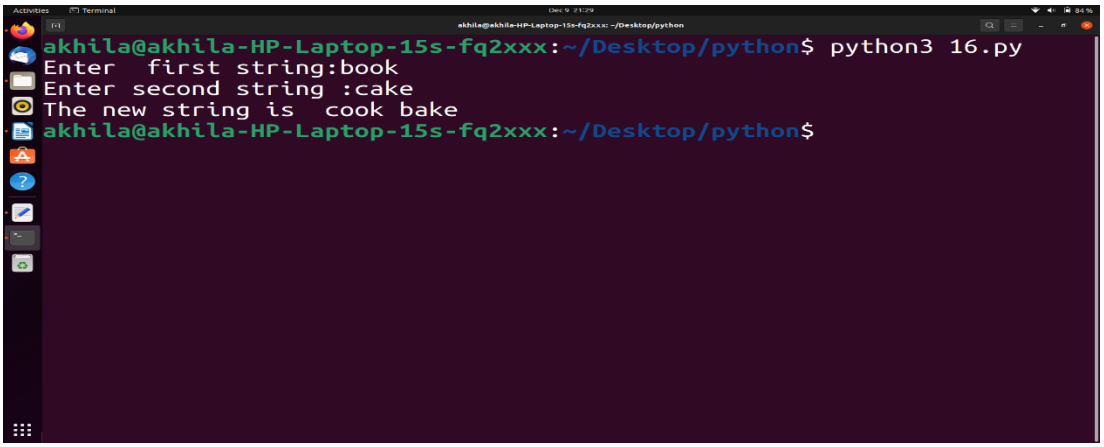
Sl no.:-15

Aim:-Create a single string seperated with space from two strings by swapping the charater at position 1.

Program code:-

```
a=input("Enter string 1:")
b=input("Enter string 2:")
a1=b[0]+a[1:]
b1=a[0]+b[1:]
c=a1+' '+b1
print(c)
```

Output:-



Sl no.:-16

Aim:-Sort dictionary in asending and decending order.

Program code:-

```
d1={"name":"Akhila","age":"21"}
d2={"sex":"F","qualification":"gradution"}
d1.update(d2)
print(d1)
```

Output:-

Sl no.:-17
Aim:-merge two dictionary
Program code:-

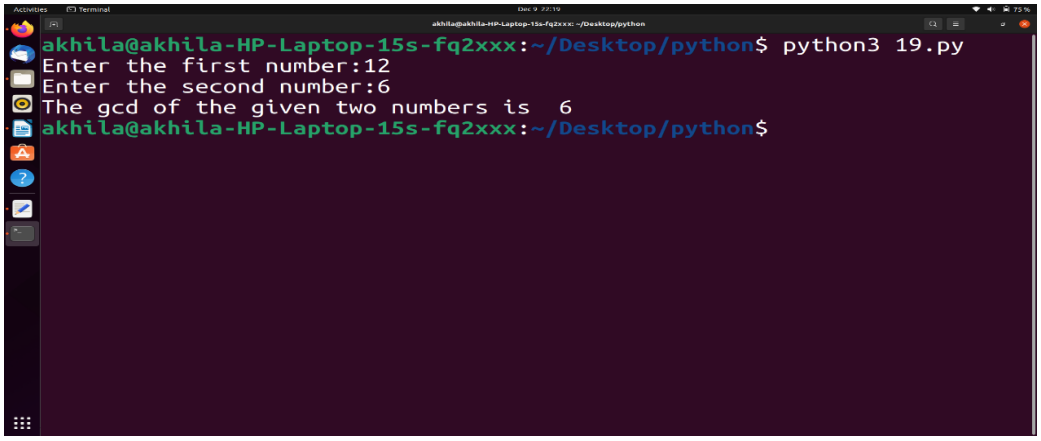
```
thisdict ={
"Name": "Akhila",
"age": 21,
"dob": "02/005/2000"
}
d={"College Name":'FISAT' , "rollno":'09'}
d.update(thisdict)
print(d)
```

Output:-

Sl no.:-18
Aim:-find GCD of 2 numbers
Program Code:-

```
x=int(input("Enter the first number:"))
y=int(input("Enter the second number:"))
z=max(x,y)
for i in range(1,z+1):
        if(x%i==0) and (y%i==0):
                gcd=i
print("The gcd of the given two numbers is ",gcd)
```

Output:-



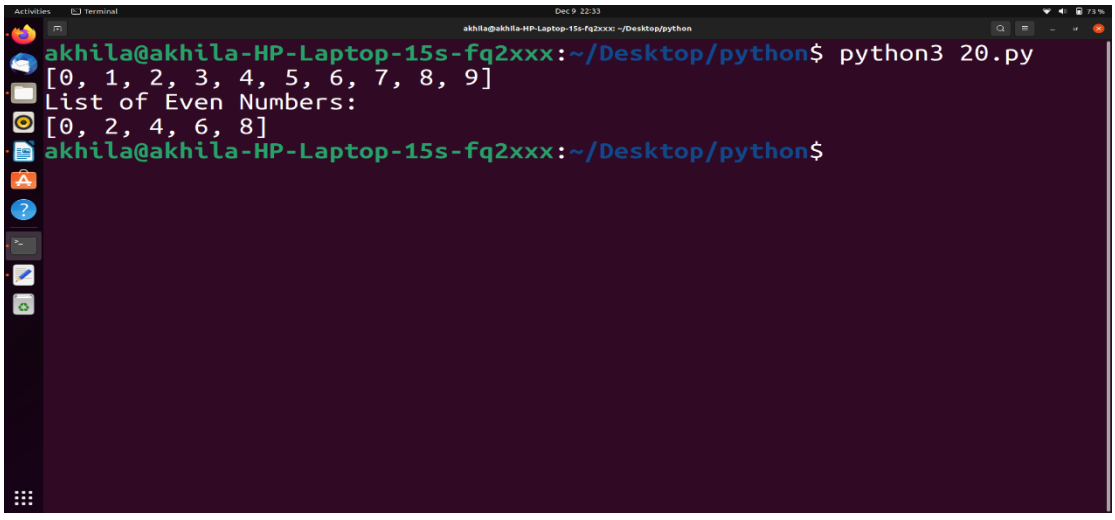
Sl no:-19

Aim:-From a list of integers ,create a list removing even numbers.

Program code:-

```
l=[0,1,2,3,4,5,6,7,8,9]
print(l)
li=[]
print("List of Even Numbers:")
for i in l:
        if(i%2==0):
                li.append(i)
print(li)
```

Output:-



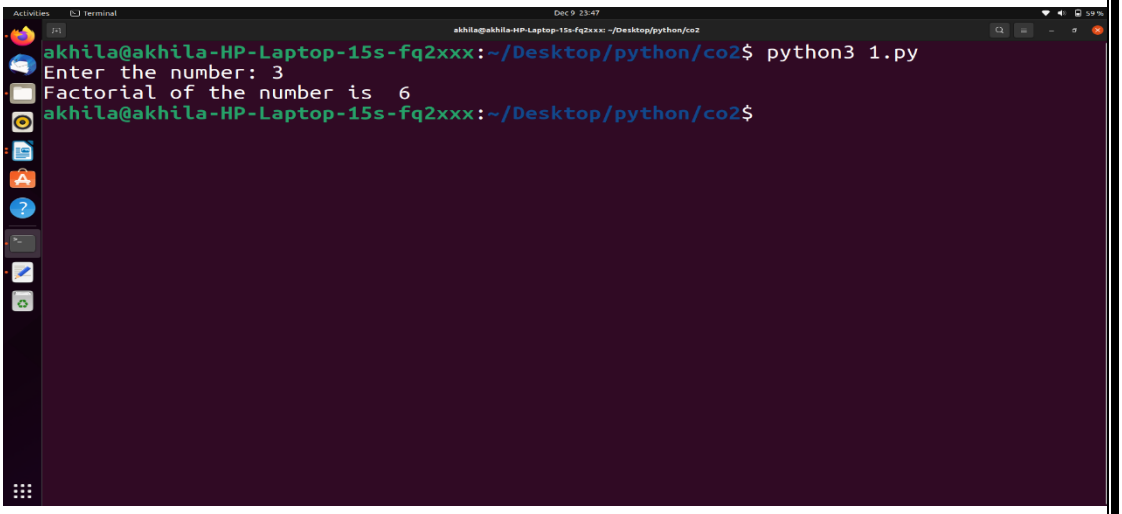
Sl no.:-20

Aim:-Program to find the factorial of a number.

Program code:-

```
num=int(input("Enter the number: "))
fact=1
for i in range(1,num+1):
    fact=fact*i
print("Factorial of the number is ",fact)
```

Output:-



COURSE OUTCOME 2

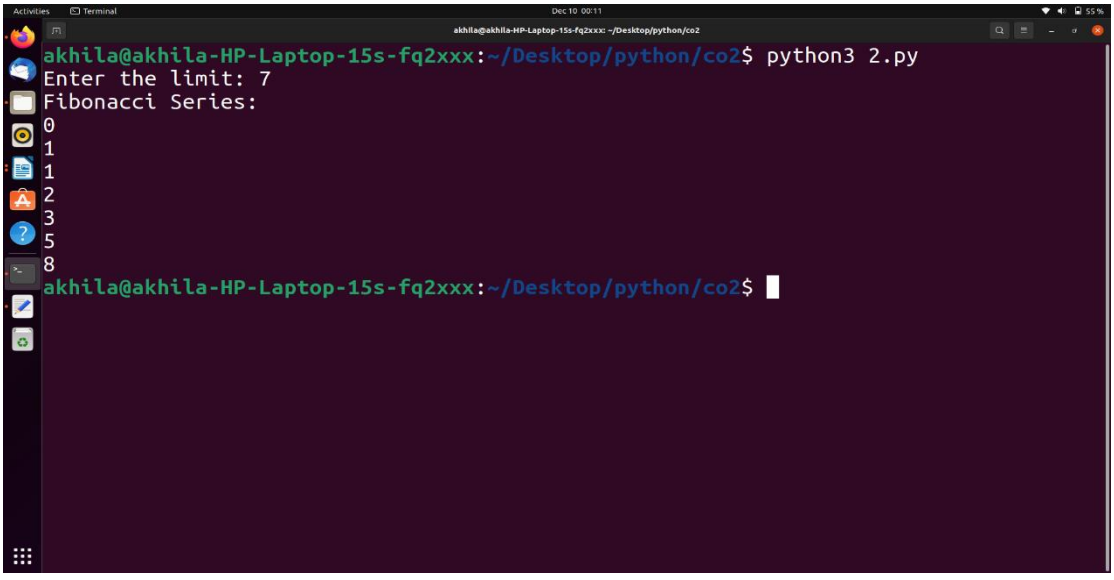
Sl no.:-21

Aim:-Generate Fibonacci series of N terms.

Program Code:-

```
n=int(input("Enter the limit: "))
a=0
b=1
c=0
print("Fibonacci Series:")
print(a)
print(b)
for i in range(3,n+1):
    c=a+b
    print(c)
    a=b
    b=c
```

Output:-



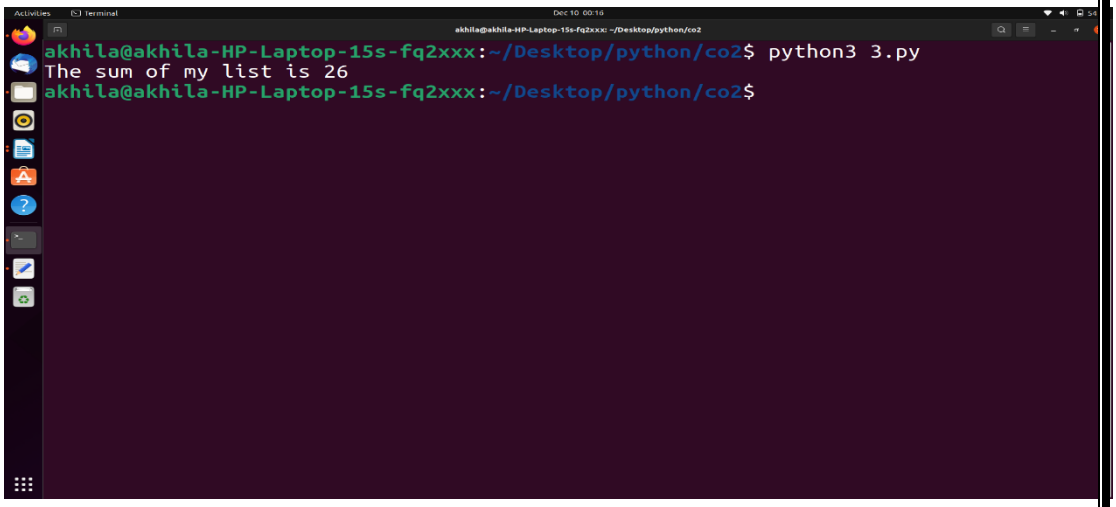
Sl no.:-22

Aim:-Find the sum of all items in a list.

Program code:-

```
def sum_of_list(l):
  total = 0
  for val in l:
    total = total+val
  return total
my_list = [3,5,7,9,2]
print("The sum of my list is", sum_of_list(my_list))
```

Output:-



Sl no.:-23

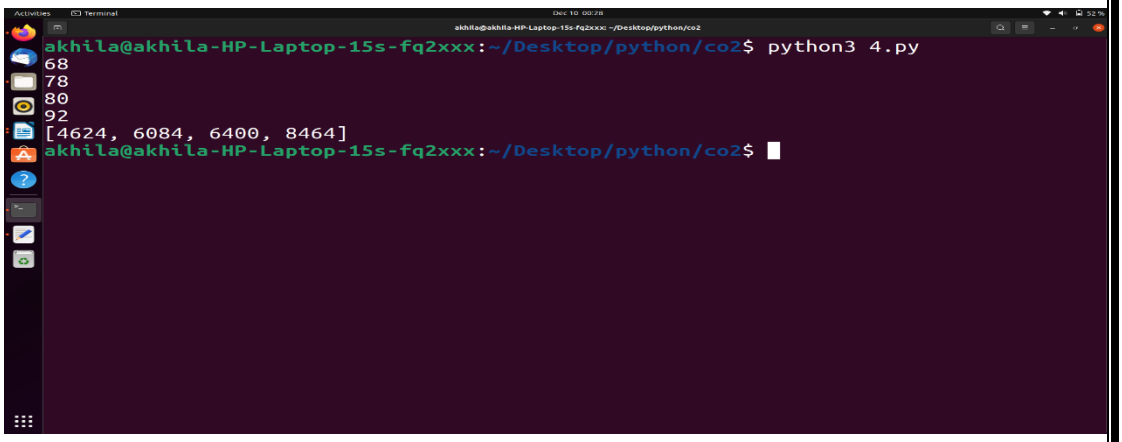
Aim:-Generate a list of 4 digit number in a give range with all their digits even and the number is a perfect square.

Program code:-

```
limit1=4286
limit2=8642
list1=[]
for i in range(limit1,limit2):
        j=i
        digit=[]
        while(i!=0):
                digit.append(i%10)
                i=int(i/10)
```

```
count=0
for n in digit:
        if n%2==0:
                count=count+1
if count==4:
        for k in range(31,100):
                if((k**2)==j):
                        list1.append(j)
                        print(k)
print(list1)
```

Output:-



Sl no:-24

Aim:-Display the given pyramid with step number accepted from user.

Eg=4

1

2 4

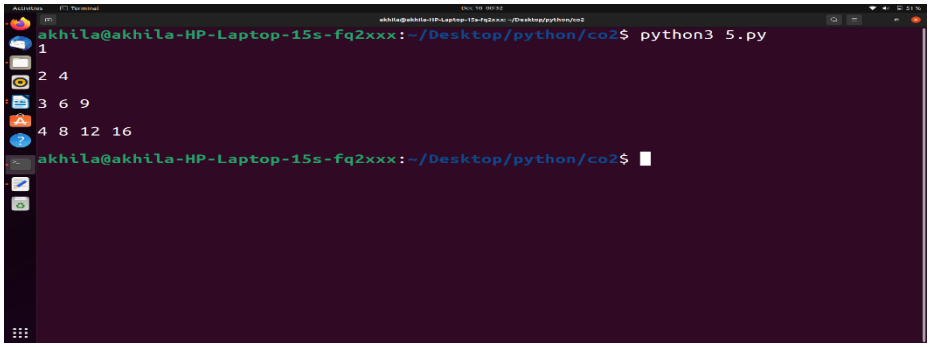
3 6 9

4 8 12 16

Program code:-

```
        for i in range(1,5):
                for j in range(1,i+1):
                        print(i*j,end=" ")
                print("\n")
```

Output:-



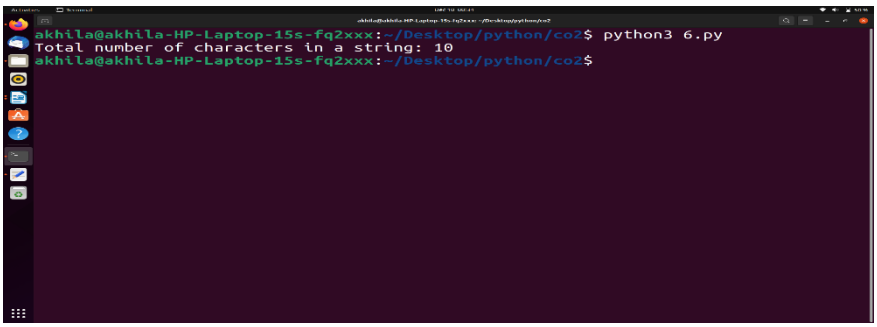
Sl no.:-25

Aim:-Count the number of charater in a string.

Program Code:-

```
string = "hello world";
count = 0;
for i in range(0, len(string)):
   if(string[i] != ' '):
      count = count + 1;
print("Total number of characters in a string: " + str(count));
```

Output:-



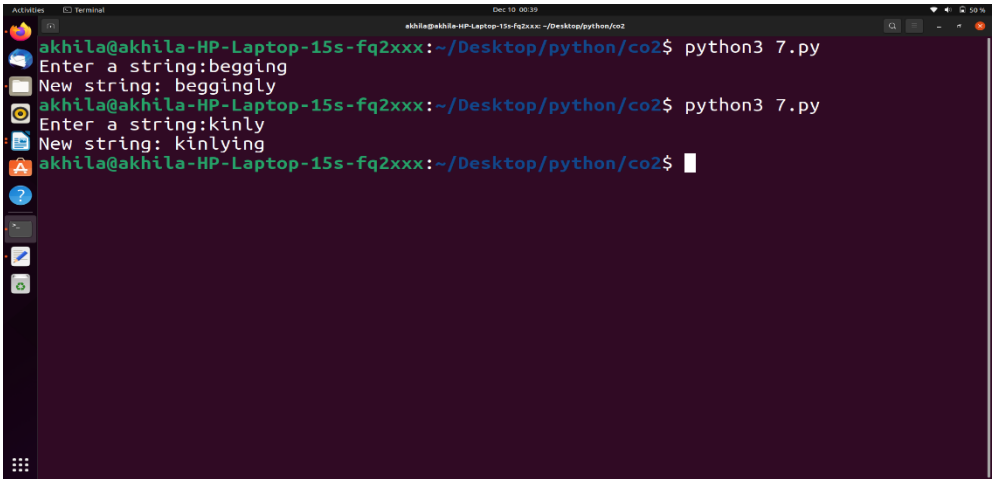
Sl no.:-26

Aim:-Add ‘ing’at the end of a given string.If it already end with ‘ing’,then add ’ly’.

Program code:-

```
s=input("Enter a string:")
if s[-3:]=='ing':
  s=s+'ly'
else:
  s=s+'ing'
print("New string:",s)
```

Output:-



Sl no:-27

Aim:-Accept a list of words and return length of longest word.

Program code:-

```
l1=[]
num=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,num):
 l1.append(input(""))
longest=l1[0]
for i in range(1,num):
 if(len(l1[i])>len(longest)):
  longest=l1[i]
print("Length of longest word is",len(longest))
```

Output:-

Sl no.:-28

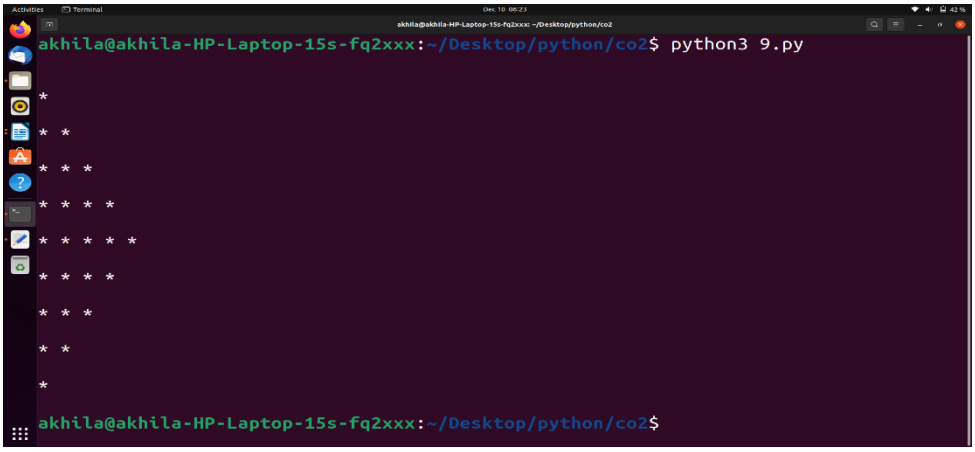
Aim:-Construct the pattern using nested loop.

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

Program code:-

```
for i in range(0,5):
    for j in range(0,i):
        print("*",end=" ")
    print("\n")
for i in range(5,0,-1):
    for j in range(0,i):
        print("*",end=" ")
    print("\n")
```

Output:-



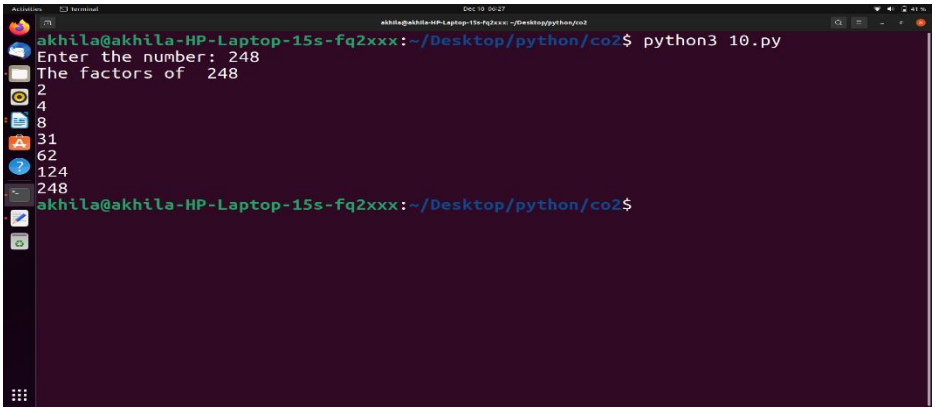
Sl no:-29

Aim:-Generate all factors of a number.

Program code:-

```
n=int(input("Enter the number: "))
i=2
print("The factors of ",n)
while i<=n:
    if (n % i==0):
        print(i)
    i = i + 1
```

Output:-



COURSE OUTCOME 3

Sl no.:-30

Aim :- Create a package graphics with modules rectangle,circle and sub-package 3D graphics with module cuboid and sphere. Include methods to find area and perimeter of respective figures in each modules. Write programs that finds area and perimeter of figures by different importing statements.

Terminal Commands:-

Program Code:-

graphics\rectangle.py

```
class Rectangle:
def __init__(self,length,width):
        self.length=length
        self.width=width
def area(self):
        return (self.length*self.width)
def perimeter(self):
        return (2*(self.length+self.width))
```

graphics\circle.py

```
global pi
pi=3.1416
class Circle:
        global pi
        pi=3.1416
        def __init__(self,radius):
                self.radius=radius
        def area(self):
                return (pi*(self.radius**2))
```

```
        def perimeter(self):
                return (2*pi*self.radius)
```

tdgraphics\cuboid.py

```
class Cuboid:
def __init__(self,length,width,height):
        self.l=length
        self.w=width
        self.h=height
def volume(self):
        return (self.l*self.w*self.h)
def area(self):
        l=self.l
        w=self.w
        h=self.h
        return (2*((l*w)+(w*h)+(l*h)))
```

tdgrapics\sphere.py

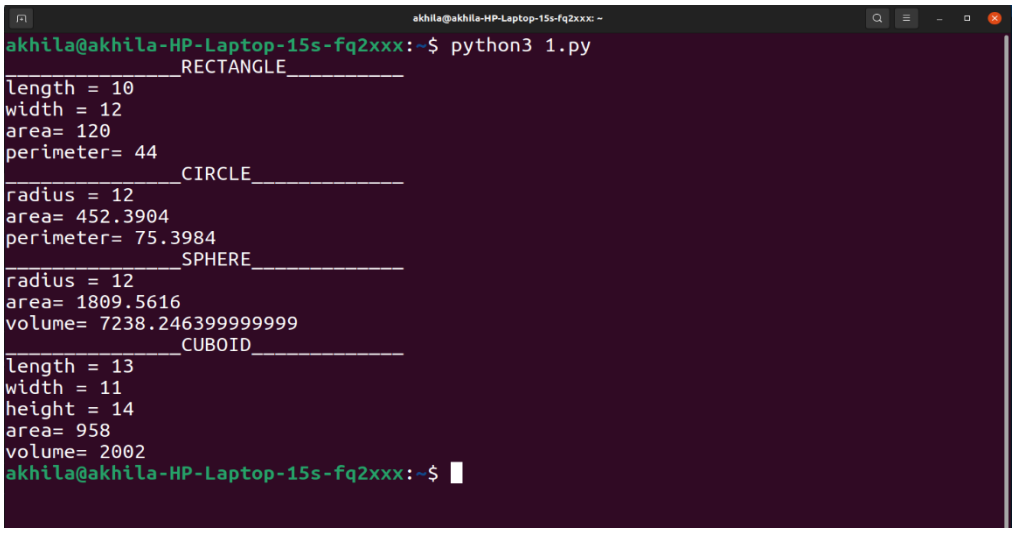
```
global pi
pi=3.1416
class Sphere:
        def __init__(self,radius):
                self.radius=radius
        def volume(self):
                r=self.radius
                return ((4/3)*pi*(r**3))
        def area(self):
                r=self.radius
                return (4*pi*(r**2))
```

1.py

```
from graphics import rectangle as rt
from graphics import circle
from graphics.tdgraphics import sphere
from graphics.tdgraphics import cuboid


#Rectangle
r=rt.Rectangle(10,12)
```

```
print("_________________RECTANGLE_____________")
print("length =",r.length)
print("width =",r.width)
print("area=",r.area())
print("perimeter=",r.perimeter())

#Circle
c=circle.Circle(12)
print("_________________CIRCLE_________________")
print("radius =",c.radius)
print("area=",c.area())
print("perimeter=",c.perimeter())

#Sphere
s=sphere.Sphere(12)
print("_________________SPHERE_________________")
print("radius =",s.radius)
print("area=",s.area())
print("volume=",s.volume())
#Cuboid
cu=cuboid.Cuboid(13,11,14)
print("_________________CUBOID_________________")
print("length =",cu.l)
print("width =",cu.w)
print("height =",cu.h)
print("area=",cu.area())
print("volume=",cu.volume())
```

Output:-



COURSE OUTCOME 4

Sl no.:-31

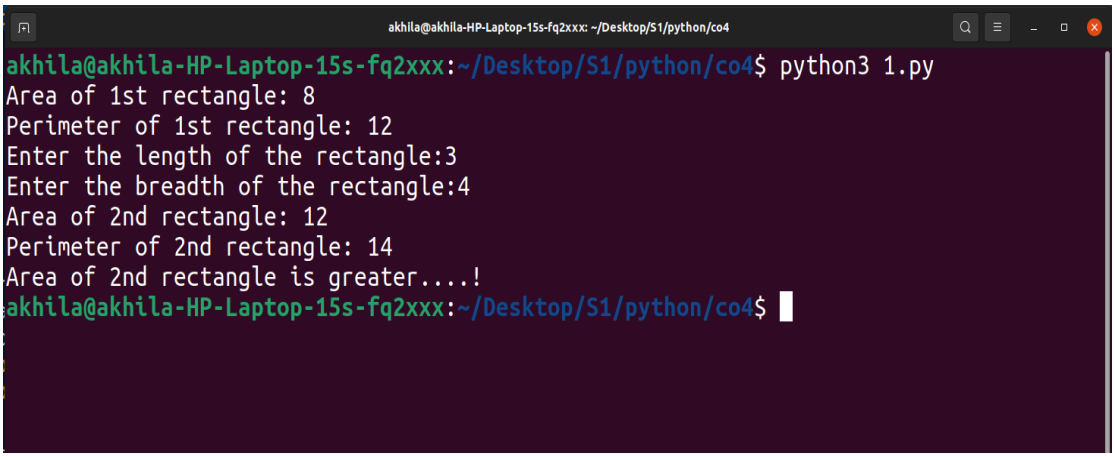
Aim:-Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two rectangle objects by their area.

Program code:-

```
class Rectangle:
def __init__(self,length,breadth):
        self.len=length
        self.brea=breadth
def area(self):
        return self.len*self.brea


def perimeter(self):
        return 2*(self.len+self.brea)


r1=Rectangle(2,4)
a1=r1.area()
p1=r1.perimeter()
print("Area of 1st rectangle:",a1)
print("Perimeter of 1st rectangle:",p1)


l=int(input("Enter the length of the rectangle:"))
b=int(input("Enter the breadth of the rectangle:"))
r2=Rectangle(l,b)
a2=r2.area()
p2=r2.perimeter()
print("Area of 2nd rectangle:",a2)
print("Perimeter of 2nd rectangle:",p2)
```

```
    if (a1>a2):
        print("Area of 1st rectangle is greater.....!")
    elif(a1==a2):
        print("Area of both rectangle is same.....!")
    else:
        print("Area of 2nd rectangle is greater....!")
```

Output:-



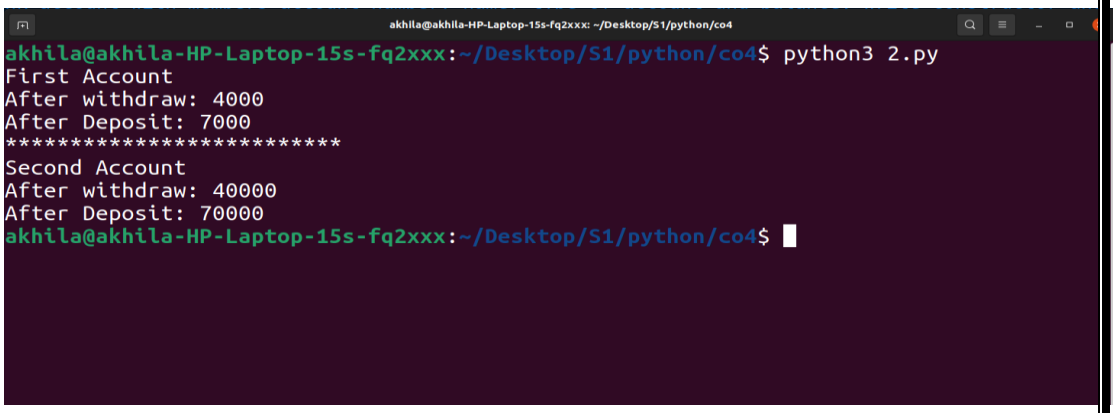
Sl no.:-32

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

Program code:-

```
class Bank:
        def __init__(self,acc_no,name,type_of_acc,balance):
                self.acc_no=acc_no
                self.name=name
                self.type_of_acc=type_of_acc
                self.balance=balance

        def deposit(self,x):
                return self.balance+x

        def withdraw(self,y):
                return self.balance-y

print('First Account')
acc1=Bank(111,"Emma","personal",5000)
w1=acc1.withdraw(1000)
d1=acc1.deposit(2000)
print('After withdraw:',w1)
print('After Deposit:',d1)
print('**************************')
print('Second Account')
acc2=Bank(222,"Emliy","personal",50000)
w2=acc2.withdraw(10000)
d2=acc2.deposit(20000)
print('After withdraw:',w2)
print('After Deposit:',d2)
```

Output:-



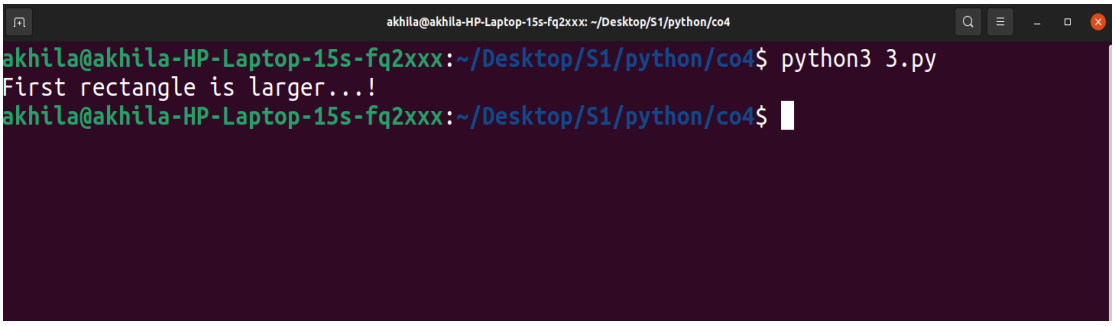
Sl no.:-33

Aim:-Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of two rectangles.

Program code:-

```
class Rectangle:
        def __init__(self,lenght,width):
                self.__length=lenght
                self.__width=width
        def __lt__(self,rec2):
                if self.__length * self.__width < rec2.__length *
rec2.__width :
                        return True
                else:
                        return False
rec1=Rectangle(2,3)
rec2=Rectangle(1,3)

if rec1 < rec2:
        print("Second rectangle is larger...!")
else:
        print("First rectangle is larger...!")
```

Output:-



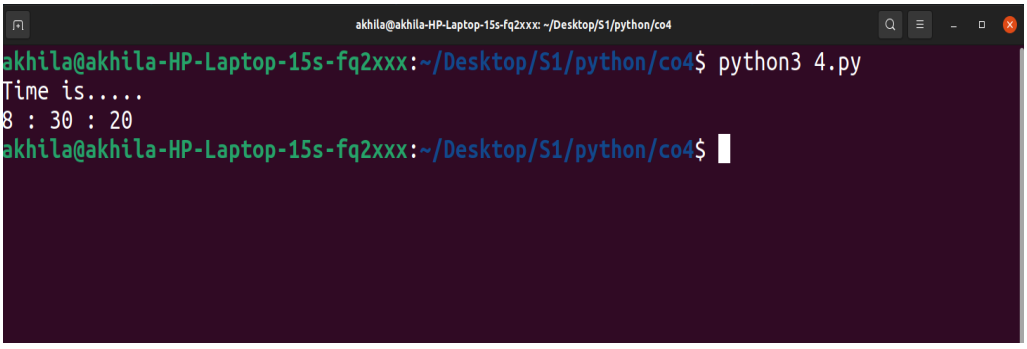
Sl no.:-34

Aim:-Create a class Time with private attributes hour,minute and second. Overload '+' operator to find sum of two time.

Program code:-

```
class Time:
    def __init__(self,hour,minute,second):
        self.__hour=hour
        self.__minute=minute
        self.__second=second

    def __add__(self,t2):
        x=t1.__hour+t2.__hour
        y=t1.__minute+t2.__minute
        z=t1.__second+t2.__second
        print('Time is.....')
        print(x,":",y,":",z)
t1=Time(4,10,5)
t2=Time(4,20,15)
t3=t1+t2
```

Output:-



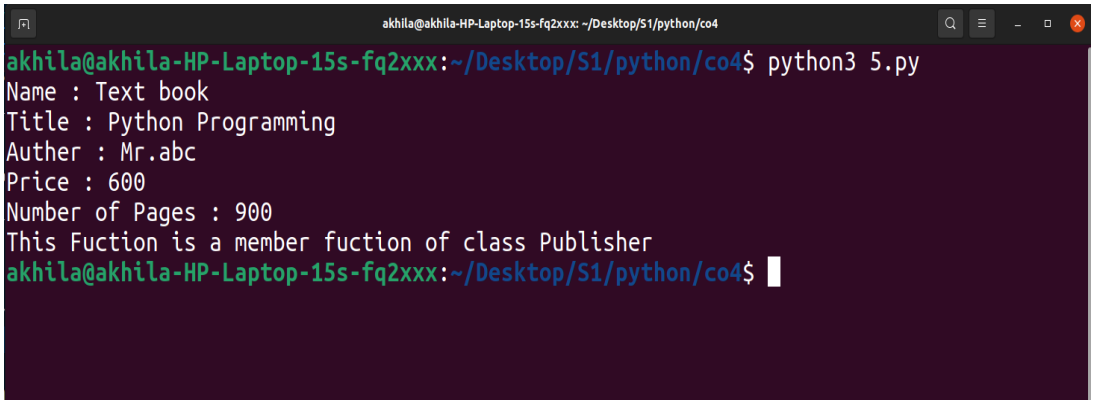
Sl no.:-35

Aim:-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 overreading.

Program code:-

```
class Publisher:
        def __init__(self,name):
                self.name=name

class Book(Publisher):
        def __init__(self,name,title,author):
                super().__init__(name)
                self.title=title
                self.author=author


class Python(Book):
        def __init__(self,name,title,author,price,pages):
                super().__init__(name,title,author)
                self.price=price
                self.pages=pages

        def print_function(self):
                print("Name :",self.name)
                print("Title :",self.title)
                print("Auther :",self.author)
                print("Price :",self.price)
                print("Number of Pages :",self.pages)

p1=Python("Text book","Python Programming","Mr.abc",600,900)
p1.print_function()
p2=Book("a","b","c")
p2.print_function()
```

Output:-


akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$ python3 5.py
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 600
Number of Pages : 900
This Fuction is a member fuction of class Publisher
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$

COURSE OUTCOME 5

Sl no.:-36

Aim:-Write a program to read a file line by line and store it into a list.

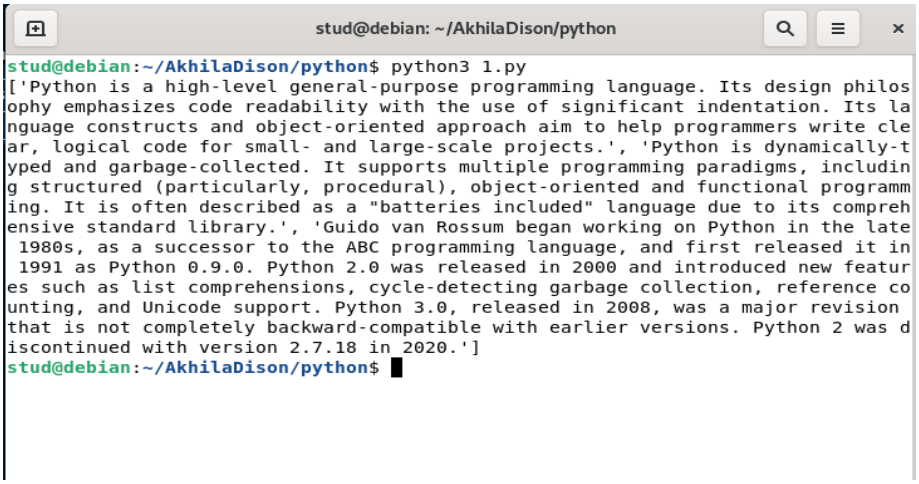
Program code:-

```
file=open("text.txt","r")
lines=[]
for line in file:
        lines.append(line.strip())
print(lines)
```

text.txt

Python is a high-level general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Output:-



Sl no.:-37

Aim:-Write a Python program to read each row from a given csv file and print a list
of strings.

Program code:-

```
import csv
with open("text.csv","r") as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
```

test.csv

```
Id,Name,Desig,Salary
001,Emma,Manager,100000
002,Anu,Secretary,40000
003,Emily,Security,30000
```

Output:-