FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)^{TM}

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ANJU K S

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 26**

Register No:FIT21MCA-2026

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 **ANJU K S (FIT21MCA-2026)** 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 Signature of H O D

Name: Name:

Date of University practical examination ………………………

Signature of Signature of
Internal Examiner External Examiner

CONTENT

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In –Charge
27	9/12/2021	Accept a list of words and return length of longest word.	21	
28	9/12/2021	Construct following pattern using nested loop *	22	
29	9/12/2021	Generate all factors of a number.	23	
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)	24	
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.	26	
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.	27	

COURSE OUTCOME 1

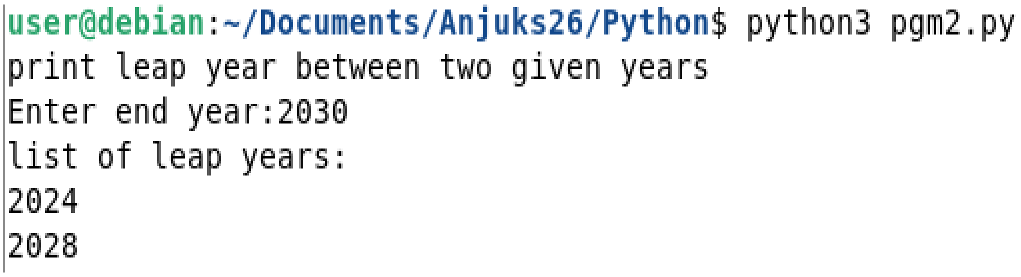
AIM

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

PROGRAM

```
print("print leap year between two given years");
startyear=2021
endyear=int(input("Enter end year:"))
print("list of leap years:")
for year in range(startyear,endyear):
        if(0==year%4):
                print(year)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm2.py
print leap year between two given years
Enter end year:2030
list of leap years:
2024
2028
```

AIM

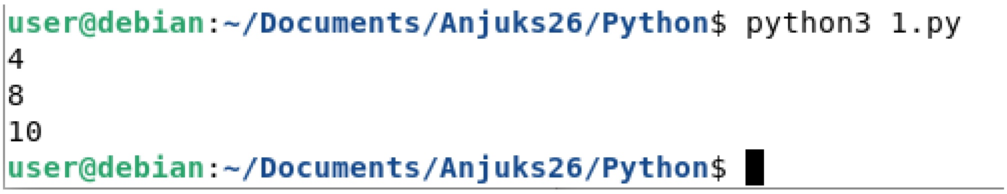
2. List comprehensions:

(a) Generate positive list of numbers from a given list of integers

PROGRAM

```
list=[-11,4,8,-34,10,-14]
  for num in list:
        if num>=0:
                print(num)
```

OUTPUT



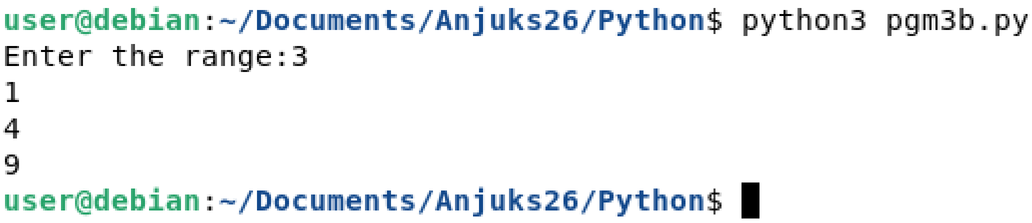
```
user@debian:~/Documents/Anjuks26/Python$ python3 1.py
4
8
10
user@debian:~/Documents/Anjuks26/Python$ 
```

(b) Square of N numbers

PROGRAM

```
n=int(input("Enter the range:"))
   for num in range(1,n+1):
         num=num*num
         print(num)
```

OUTPUT



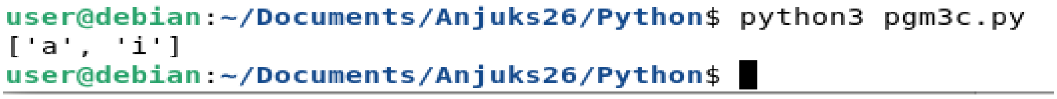
```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm3b.py
Enter the range:3
1
4
9
user@debian:~/Documents/Anjuks26/Python$ 
```

(c) Form a list of vowels selected from a given word

PROGRAM

```
list=[]
s="playing"
for i in s:
   if i in "aeiou AEIOU":
          list.append(i)
print(list)
```

OUTPUT



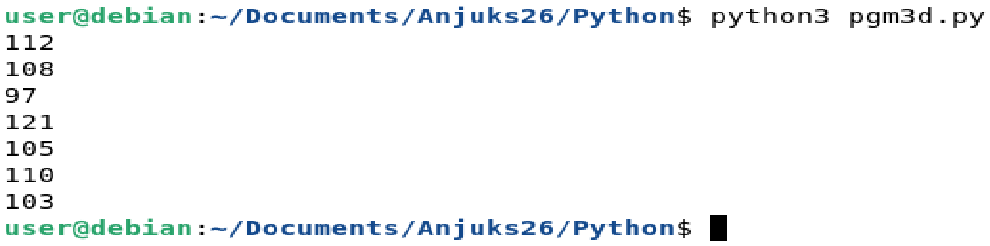
```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm3c.py
['a', 'i']
user@debian:~/Documents/Anjuks26/Python$ █
```

(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

PROGRAM

```
for i in 'p''l''a''y''i''n''g':
    x=ord(i)
    print(x)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm3d.py
112
108
97
121
105
110
103
user@debian:~/Documents/Anjuks26/Python$ █
```

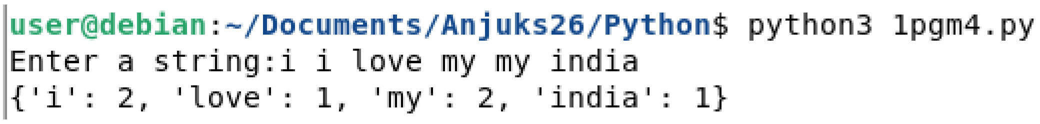
AIM

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

PROGRAM

```
s=input("Enter a string:")
count=dict()
word=s.split()
for i in word:
   if i in count:
     count[i]+=1
   else:
     count[i]=1
print(count)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 1pgm4.py
Enter a string:i i love my my india
{'i': 2, 'love': 1, 'my': 2, 'india': 1}
```

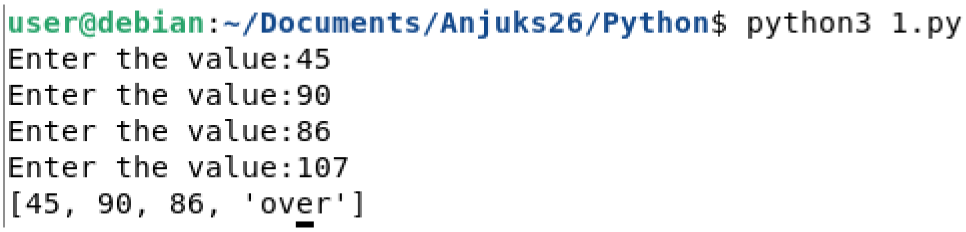
AIM

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

PROGRAM

```
list=[]
while True:
   n=int(input("Enter the num:"))
   if(n<=100):
     list.append(n)
   else:
      list.append(‘over’)
      print(list)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 1.py
Enter the value:45
Enter the value:90
Enter the value:86
Enter the value:107
[45, 90, 86, 'over']
```

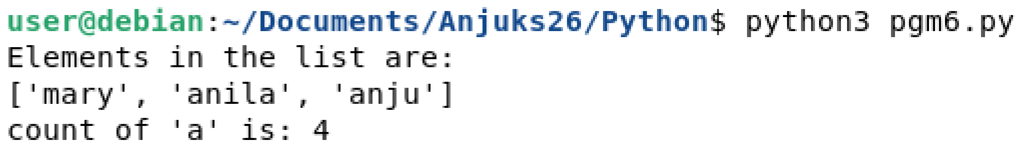
AIM

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

PROGRAM

```
list=['mary','anila','anju']
print("Elements in the list are:")
print(list)
count=0
for word in list:
   for i in word:
      if i=='a':
       count+=1
print("count of 'a' is:", count)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm6.py
Elements in the list are:
['mary', 'anila', 'anju']
count of 'a' is: 4
```

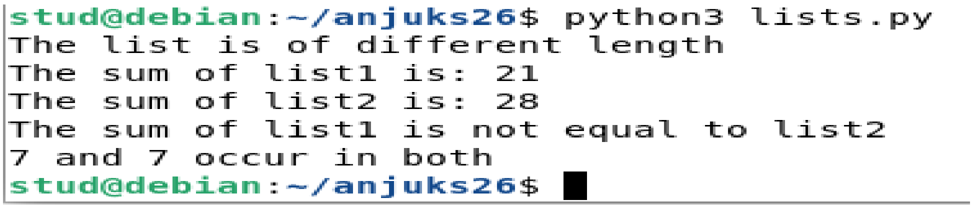
AIM

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

PROGRAM

```
l1=[5,6,3,7]
l2=[2,1,7,10,8]
x=len(l1)
y=len(l2)
if x==y:
   print("The list is of same length")
else:
   print("The list is of different length")
sum1=0
sum2=0
for i in range(len(l1)):
         sum1=sum1+l1[i]
print("The sum of list1 is:",sum1)
for j in range(len(l2)):
                         sum2=sum2+l2[j]
print("The sum of list2 is:",sum2)
if sum1=sum2:
            print("The sum of list1 is equal to list2")
else:
            print("The sum of list1 is not equal to list2")
for i in range(x):
            for j in range(y):
           if l1[i]==l2[j]:
              print(l1[i],"and",l2[j],"occur in both")
```

OUTPUT



```
stud@debian:~/anjuks26$ python3 lists.py
The list is of different length
The sum of list1 is: 21
The sum of list2 is: 28
The sum of list1 is not equal to list2
7 and 7 occur in both
stud@debian:~/anjuks26$ 
```

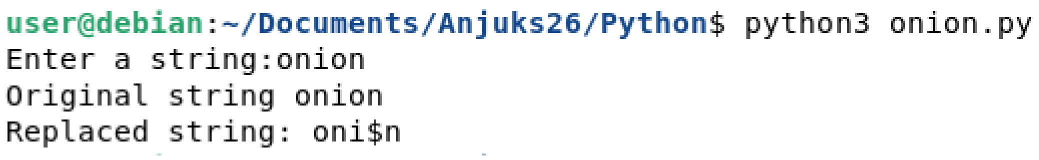
AIM

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:")
print("Original string",str)
char=str[0]
str=str.replace(char,'$')
str=char+str[2:]
print("Replaced string:",str)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 onion.py
Enter a string:onion
Original string onion
Replaced string: oni$n
```

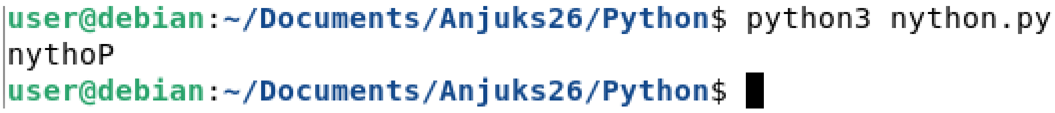
AIM

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

PROGRAM

```
s="Python"

temp=s[0]

temp1=s[-1]

n=len(s)

ns=temp1+s[1:n-1]+temp

print(ns)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 nython.py
nythoP
user@debian:~/Documents/Anjuks26/Python$ 
```

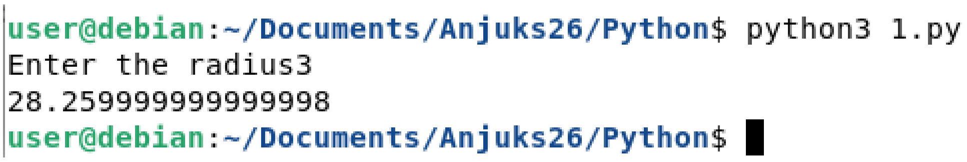
AIM

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

PROGRAM

```
r=int(input('Enter the radius'))

A=3.14*r*r

print(A)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 1.py
Enter the radius3
28.259999999999998
user@debian:~/Documents/Anjuks26/Python$ █
```

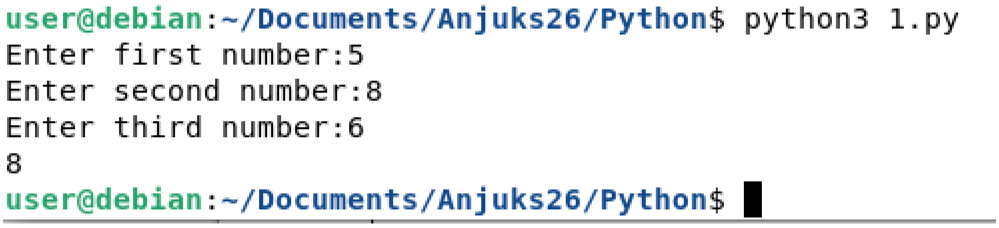
AIM

10 . Find biggest of 3 numbers entered.

PROGRAM

```
a=int(input('Enter first number:'))
     b=int(input('Enter second number:'))
     c=int(input('Enter third number:'))
     if a>b and a>c:
          print(a)
     if b>a and b>c:
          print(b)
     if c>a and c>b:
          print(c)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 1.py
Enter first number:5
Enter second number:8
Enter third number:6
8
user@debian:~/Documents/Anjuks26/Python$ █
```

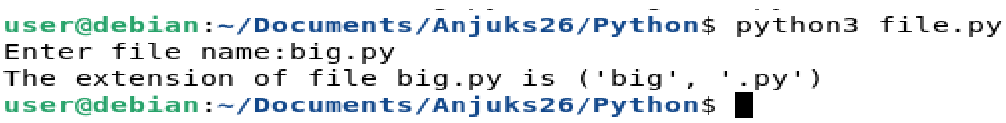
AIM

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

PROGRAM

```
import os

a=input("Enter file name:")

print("The extension of file",a,"is",os.path.splitext(a))
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 file.py
Enter file name:big.py
The extension of file big.py is ('big', '.py')
user@debian:~/Documents/Anjuks26/Python$ 
```

AIM

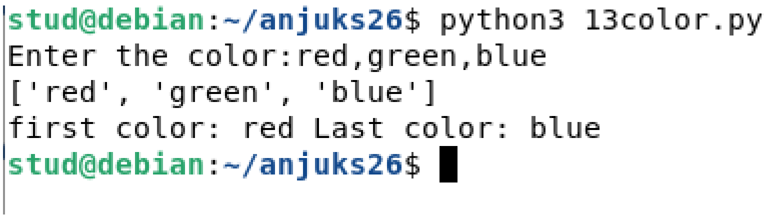
12 . Create a list of colors from comma-separated color names entered by user.

Display first and last colors.

PROGRAM

```
color=[]

str=(input("Enter the color:"))

for i in str.split(','):

        color.append(i)

print(color)

print("first color:",color[0],"Last color:",color[-1])
```

OUTPUT



```
stud@debian:~/anjuks26$ python3 13color.py
Enter the color:red,green,blue
['red', 'green', 'blue']
first color: red Last color: blue
stud@debian:~/anjuks26$ 
```

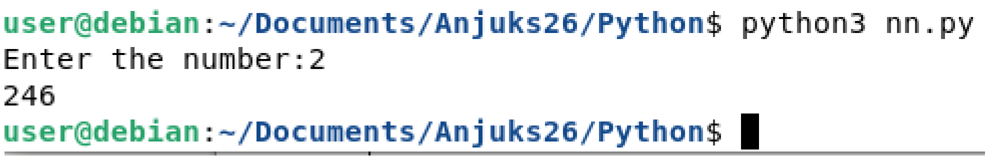
AIM

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

PROGRAM

```
n=int(input("Enter the number:"))
a=n*1
b=n*11
c=n*111
s=a+b+c
print(s)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 nn.py
Enter the number:2
246
user@debian:~/Documents/Anjuks26/Python$
```

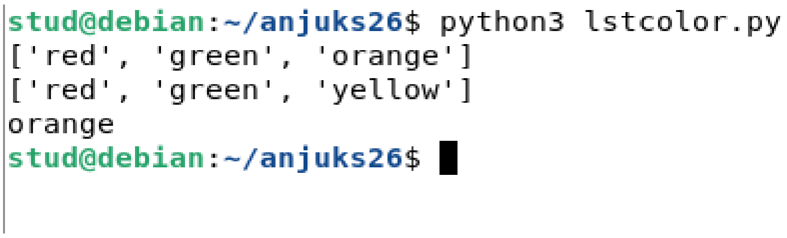
AIM

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

PROGRAM

```
l1=['red','green','orange']
l2=['red','green','yellow']
print(l1)
print(l2)
for i in l1:
    if i not in l2:
        print(i)
```

OUTPUT



```
stud@debian:~/anjuks26$ python3 lstcolor.py
['red', 'green', 'orange']
['red', 'green', 'yellow']
orange
stud@debian:~/anjuks26$
```

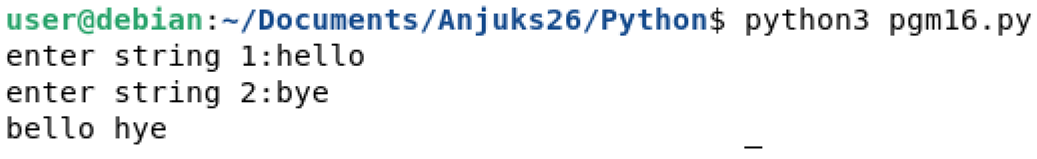
AIM

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

PROGRAM

```
a=input("enter string 1:")
b=input("enter string 2:")
new_a = b[:1] + a[1:]
new_b = a[:1] + b[1:]
c=new_a+ ' ' + new_b
print(c)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pgm16.py
enter string 1:hello
enter string 2:bye
bello hye
```

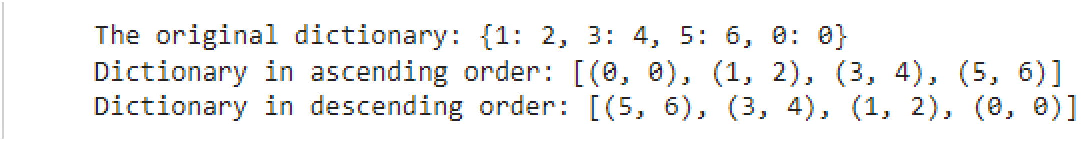
AIM

16. Sort dictionary in accenting and descending order.

PROGRAM

```
dic={1:2,3:4,5:6,0:0}

print("The original dictionary:",dic)

sorted_dic=sorted(dic.items())

print("Dictionary in ascending order:",sorted_dic)

sorted_reverse=sorted(dic.items(),reverse=True)

print("Dictionary in descending order:",sorted_reverse)
```

OUTPUT



```
The original dictionary: {1: 2, 3: 4, 5: 6, 0: 0}
Dictionary in ascending order: [(0, 0), (1, 2), (3, 4), (5, 6)]
Dictionary in descending order: [(5, 6), (3, 4), (1, 2), (0, 0)]
```

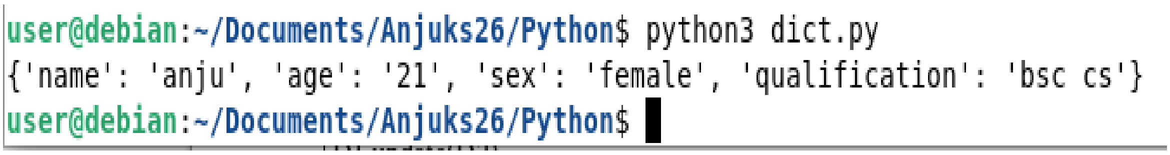
AIM

17. Merge two dictionaries

PROGRAM

```
D1={"name":"anju","age":"21"}

D2={"sex":"female","qualification":"bsc cs"}

D1.update(D2)

print(D1)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 dict.py
{'name': 'anju', 'age': '21', 'sex': 'female', 'qualification': 'bsc cs'}
user@debian:~/Documents/Anjuks26/Python$ 
```

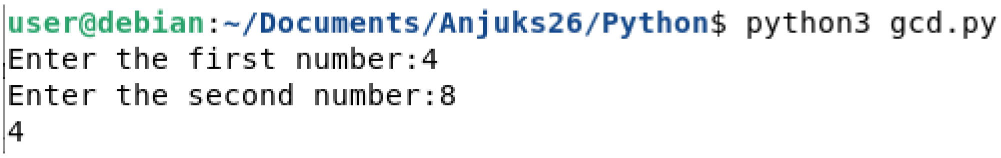
AIM

18. Find gcd of 2 numbers.

PROGRAM

```
x=int(input("Enter the first number:"))
y=int(input("Enter the second number:"))
if x>y:
  small=y
else:
  small=x
for i in range(1,small+1):
  if x%i==0 and y%i==0:
    hcf=i
print(hcf)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 gcd.py
Enter the first number:4
Enter the second number:8
4
```

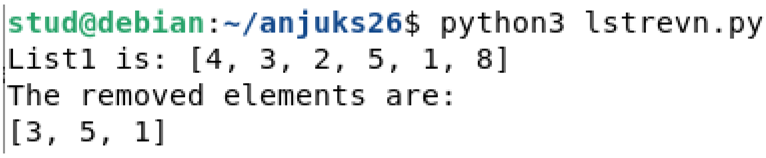
AIM

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

PROGRAM

```
l1=[4,3,2,5,1,8]
print("List1 is:",l1)
l2=[]
for i in range(len(l1)):
   if l1[i]%2!=0:
      l2.append(l1[i])
print("The removed elements are:")
print(l2)
```

OUTPUT



```
stud@debian:~/anjuks26$ python3 lstrevn.py
List1 is: [4, 3, 2, 5, 1, 8]
The removed elements are:
[3, 5, 1]
```

COURSE OUTCOME 2

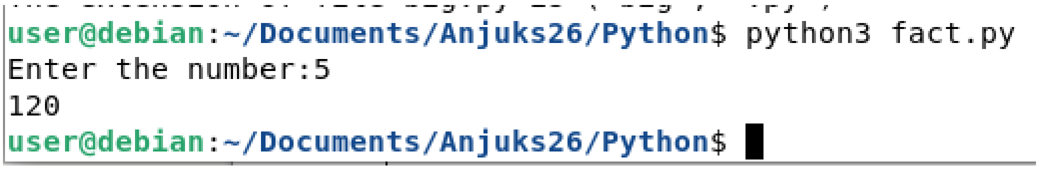
AIM

20. Program to find factorial of a number.

PROGRAM

```
n=int(input('Enter the number:'))
 fact=1
 for i in range(1,n+1):
            fact=fact*i
 print(fact)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 fact.py
Enter the number:5
120
user@debian:~/Documents/Anjuks26/Python$ █
```

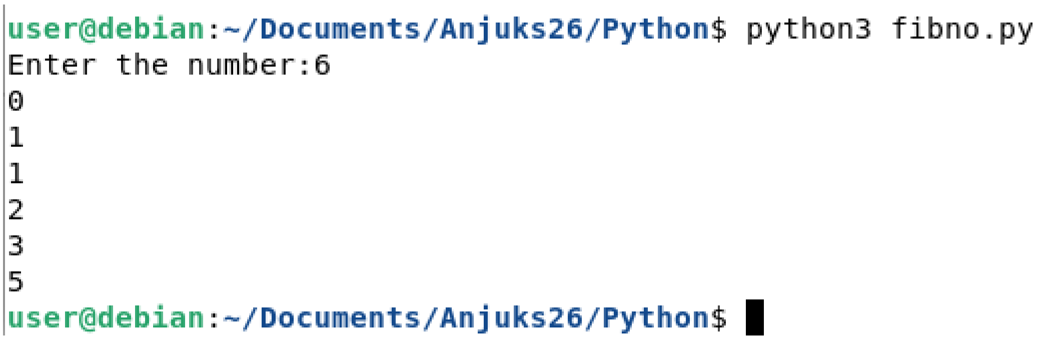
AIM

20. Generate Fibnocci series of N terms.

PROGRAM

```
n=int(input('Enter a limit:'))
        a=0
        b=1
        print(a)
        print(b)
        for i in range (2,n):
                c=a+b
                print(c)
                a=b
                b=c
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 fibno.py
Enter the number:6
0
1
1
2
3
5
user@debian:~/Documents/Anjuks26/Python$ █
```

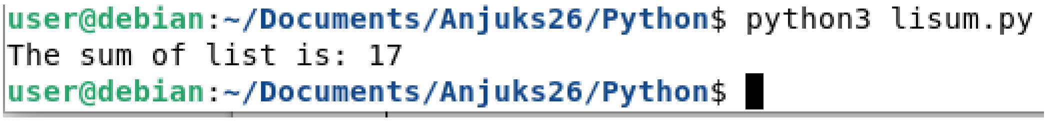
AIM

21. Find the sum of all items in a list.

PROGRAM

```
list=[2,6,9]
sum=0
for i in list:
    sum=sum+i
print("The sum of list is:",sum)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 lisum.py
The sum of list is: 17
user@debian:~/Documents/Anjuks26/Python$ 
```

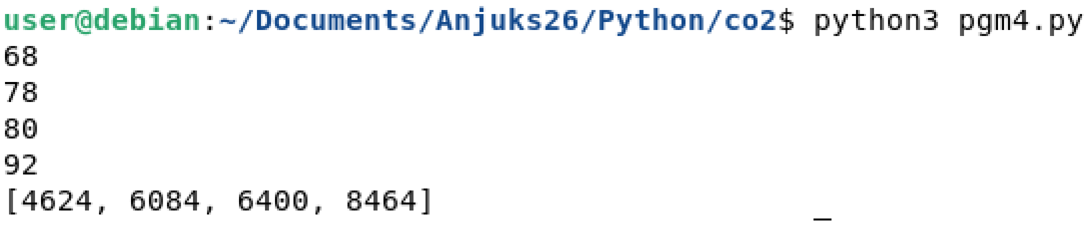
AIM

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

PROGRAM

```
limit1=1000
limit2=9999
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



```
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm4.py
68
78
80
92
[4624, 6084, 6400, 8464]
```

AIM

23. Display the given pyramid with step number accepted from user.

Eg:N=4

1

2 4

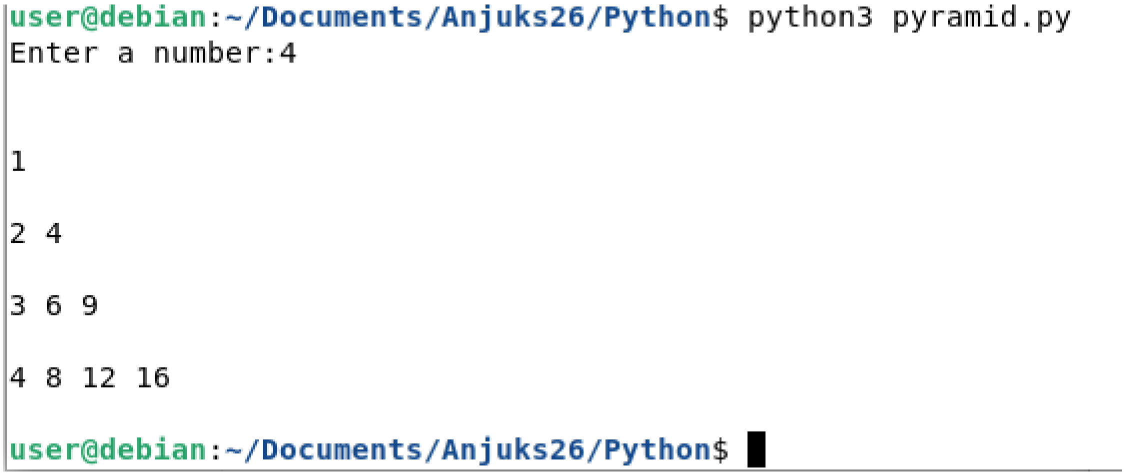
3 6 9

4 8 12 16

PROGRAM

```
n=int(input("Enter a number:"))
for j in range(0,n+1):
  for i in range(1,j+1):
    i=j*i
    print(i,end=" ");
  print("\n");
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pyramid.py
Enter a number:4


1

2 4

3 6 9

4 8 12 16

user@debian:~/Documents/Anjuks26/Python$
```

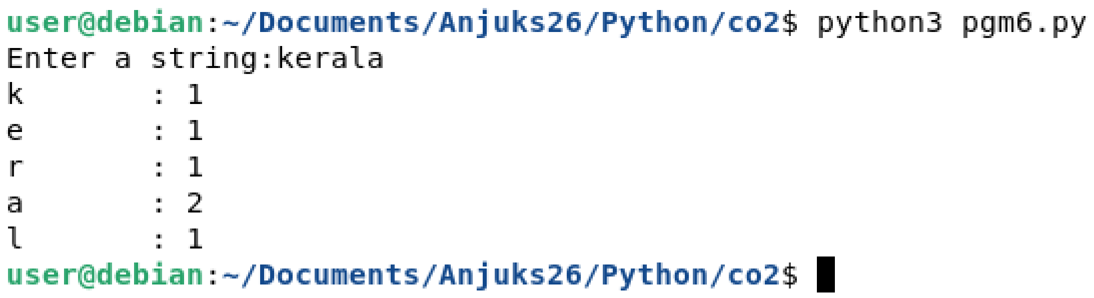
AIM

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

PROGRAM

```
string=input("Enter a string:")
list1=[]
for i in string:
 if i not in list1:
  list1.append(i)
for i in list1:
 count=0
 for j in string:
  if(i==j):
   count=count+1
 print(i,"\t:",count)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm6.py
Enter a string:kerala
k       : 1
e       : 1
r       : 1
a       : 2
l       : 1
user@debian:~/Documents/Anjuks26/Python/co2$ 
```

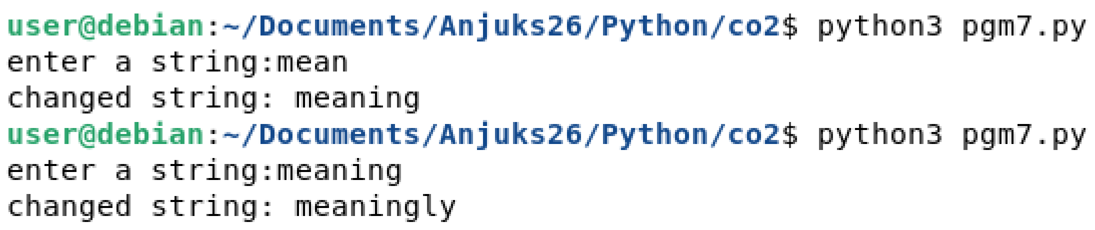
AIM

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

PROGRAM

```
str1=input("enter a string:")
if str1[-3:]=='ing':
  str1=str1+'ly'
else:
  str1=str1+'ing'
print("changed string:",str1)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm7.py
enter a string:mean
changed string: meaning
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm7.py
enter a string:meaning
changed string: meaningly
```

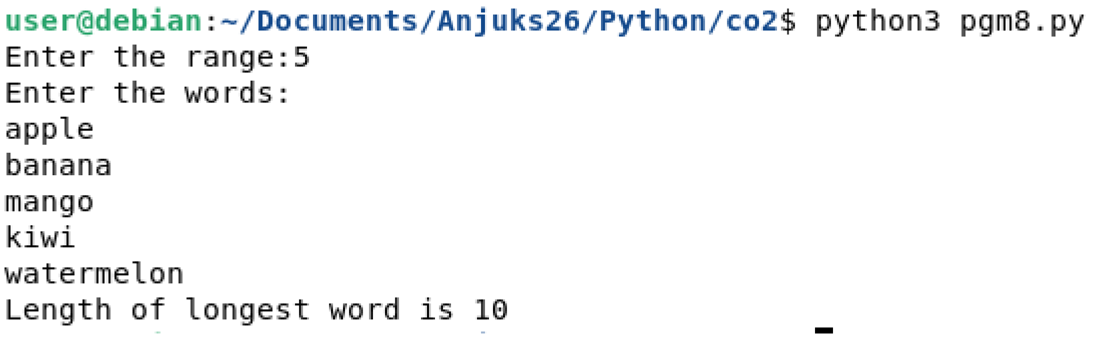
AIM

26 . Accept a list of words and retrun length of longest word.

PROGRAM

```
list1=[]
n=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,n):
  list1.append(input(""))
longest=list1[0]
for i in range(1,n):
  if(len(list1[i])>len(longest)):
    longest=list1[i]
print("Length of longest word is",len(longest))
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm8.py
Enter the range:5
Enter the words:
apple
banana
mango
kiwi
watermelon
Length of longest word is 10
```

AIM

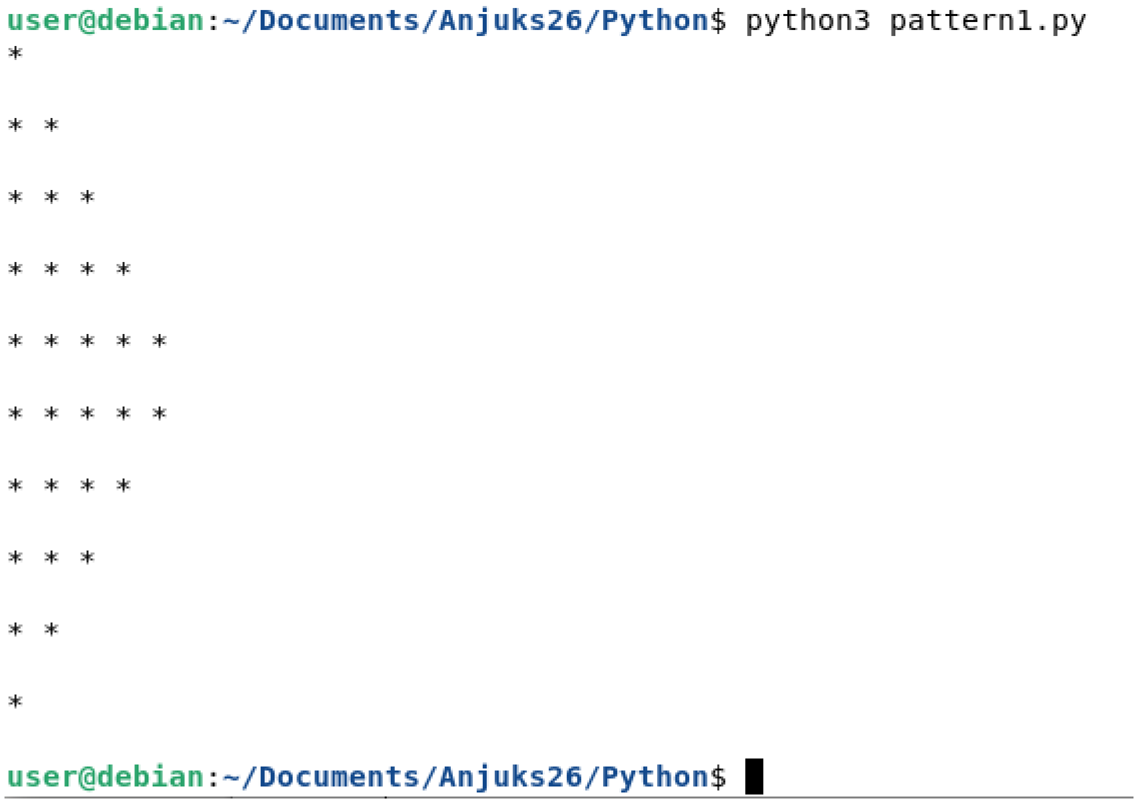
27. Construct following pattern using nested loop.

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

PROGRAM

```
for i in range(1,6):
    for j in range(1,i+1):
            print("*",end=" ")
    print("\n")
for i in range(5,0,-1):
    for j in range(1,i+1):
            print("*",end=" ")
    print("\n")
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python$ python3 pattern1.py
*

* *

* * *

* * * *

* * * * *

* * * * *

* * * *

* * *

* *

*

user@debian:~/Documents/Anjuks26/Python$ 
```

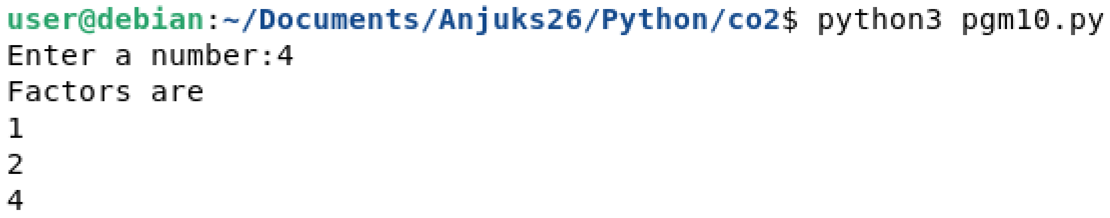
AIM

28. Generate all factors of a number.

PROGRAM

```
n=int(input("Enter a number:"))
print("Factors are")
for i in range(1,n+1):
  if(n%i==0):
    print(i)
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co2$ python3 pgm10.py
Enter a number:4
Factors are
1
2
4
```

COURSE OUTCOME 3

AIM

29. 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)

PROGRAM

graphics\circle.py

```
from math import pi
def area_circle(radius):
    return pi*radius*radius
def perimeter_circle(radius):
    return 2*pi*radius
```

graphics\rectangle.py

```
def area_rec(length,width):
    return length*width
def perimeter_rec(length,width):
    return 2*(length+width)
```

graphics\tdgraphics\cuboid.py

```
def area_cuboid(l,b,h):
    return 2*(l*h + b*h + l*b)
def volume_cuboid(l,b,h):
    return l*b*h
```

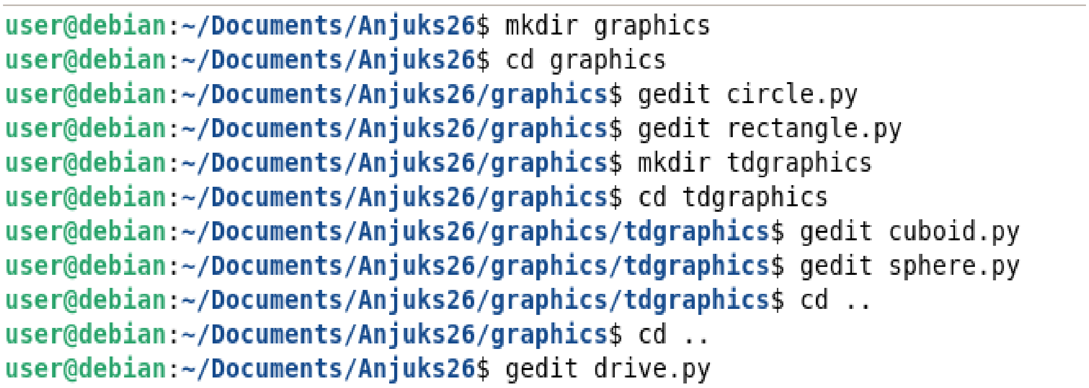
graphics\tdgraphics\sphere.py

```
from math import pi
def area_sphere(radius):
    return 4*(pi*radius*radius)
def perimeter_sphere(radius):
    return 2*pi*radius
```

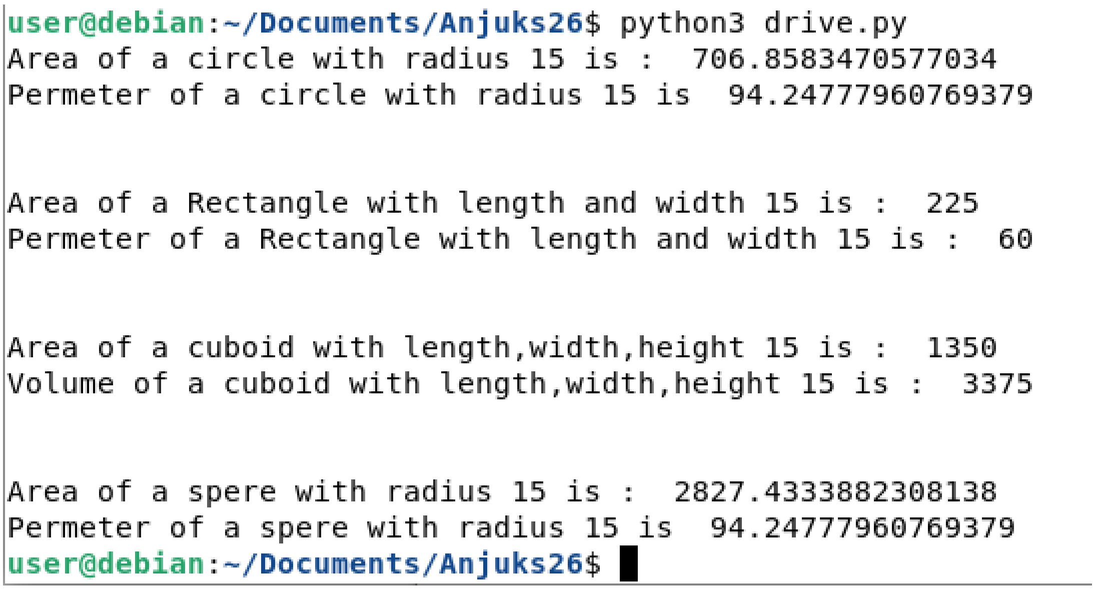
graphics.py (driver code)

```
import graphics
from graphics import circle,rectangle
from graphics.tdgraphics import cuboid,sphere
from graphics.circle import *
print("Area of a circle with radius 15 is : ",circle.area_circle(15))
print("Permeter of a circle with radius 15 is ",circle.perimeter_circle(15))
print("\n")

print("Area of a Rectangle with length and width 15 is : ",rectangle.area_rec(15,15))
print("Permeter of a Rectangle with length and width 15 is :
",rectangle.perimeter_rec(15,15))
print("\n")

print("Area of a cuboid with length,width,height 15 is :
",cuboid.area_cuboid(15,15,15))
print("Volume of a cuboid with length,width,height 15 is :
",cuboid.volume_cuboid(15,15,15))
print("\n")

print("Area of a spere with radius 15 is : ",sphere.area_sphere(15))
print("Permeter of a spere with radius 15 is ",sphere.perimeter_sphere(15))
```

OUTPUT



```
user@debian:~/Documents/Anjuks26$ mkdir graphics
user@debian:~/Documents/Anjuks26$ cd graphics
user@debian:~/Documents/Anjuks26/graphics$ gedit circle.py
user@debian:~/Documents/Anjuks26/graphics$ gedit rectangle.py
user@debian:~/Documents/Anjuks26/graphics$ mkdir tdgraphics
user@debian:~/Documents/Anjuks26/graphics$ cd tdgraphics
user@debian:~/Documents/Anjuks26/graphics/tdgraphics$ gedit cuboid.py
user@debian:~/Documents/Anjuks26/graphics/tdgraphics$ gedit sphere.py
user@debian:~/Documents/Anjuks26/graphics/tdgraphics$ cd ..
user@debian:~/Documents/Anjuks26/graphics$ cd ..
user@debian:~/Documents/Anjuks26$ gedit drive.py
```



```
user@debian:~/Documents/Anjuks26$ python3 drive.py
Area of a circle with radius 15 is :  706.8583470577034
Permeter of a circle with radius 15 is  94.24777960769379


Area of a Rectangle with length and width 15 is :  225
Permeter of a Rectangle with length and width 15 is :  60


Area of a cuboid with length,width,height 15 is :  1350
Volume of a cuboid with length,width,height 15 is :  3375


Area of a spere with radius 15 is :  2827.4333882308138
Permeter of a spere with radius 15 is  94.24777960769379
user@debian:~/Documents/Anjuks26$ █
```

COURSE OUTCOME 4

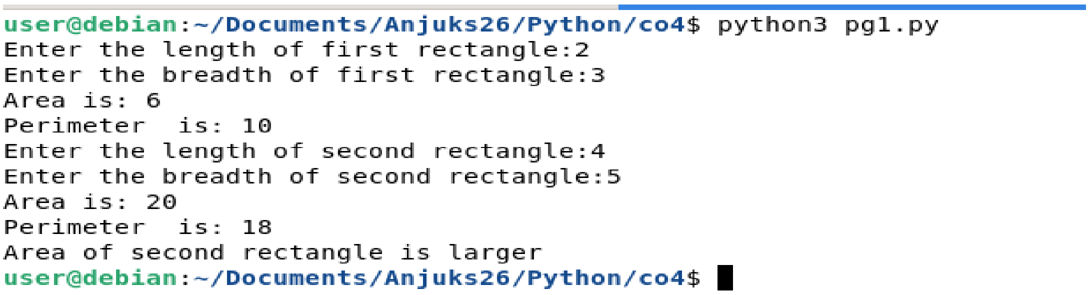
AIM

30 . 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 __init__(self,length,breadth):

                self.length=length
        self.breadth=breadth
    def area(self):
        return self.length*self.breadth

    def perimeter(self):
        return 2*(self.length+self.breadth)

l=int(input("Enter the length of first rectangle:"))
b=int(input("Enter the breadth of first rectangle:"))
rect1 = rectangle(l,b)
r1=rect1.area()
p1=rect1.perimeter()
print("Area is:",r1)
print("Perimeter  is:",p1)
l=int(input("Enter the length of second rectangle:"))
b=int(input("Enter the breadth of second rectangle:"))
rect2 = rectangle(l,b)
r2=rect2.area()
p2=rect2.perimeter()
print("Area is:",r2)
print("Perimeter  is:",p2)

if  r1 > r2 :
   print("Area of first rectangle is larger")
else:
   print("Area of second rectangle is larger")
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co4$ python3 pg1.py
Enter the length of first rectangle:2
Enter the breadth of first rectangle:3
Area is: 6
Perimeter  is: 10
Enter the length of second rectangle:4
Enter the breadth of second rectangle:5
Area is: 20
Perimeter  is: 18
Area of second rectangle is larger
user@debian:~/Documents/Anjuks26/Python/co4$ 
```

AIM

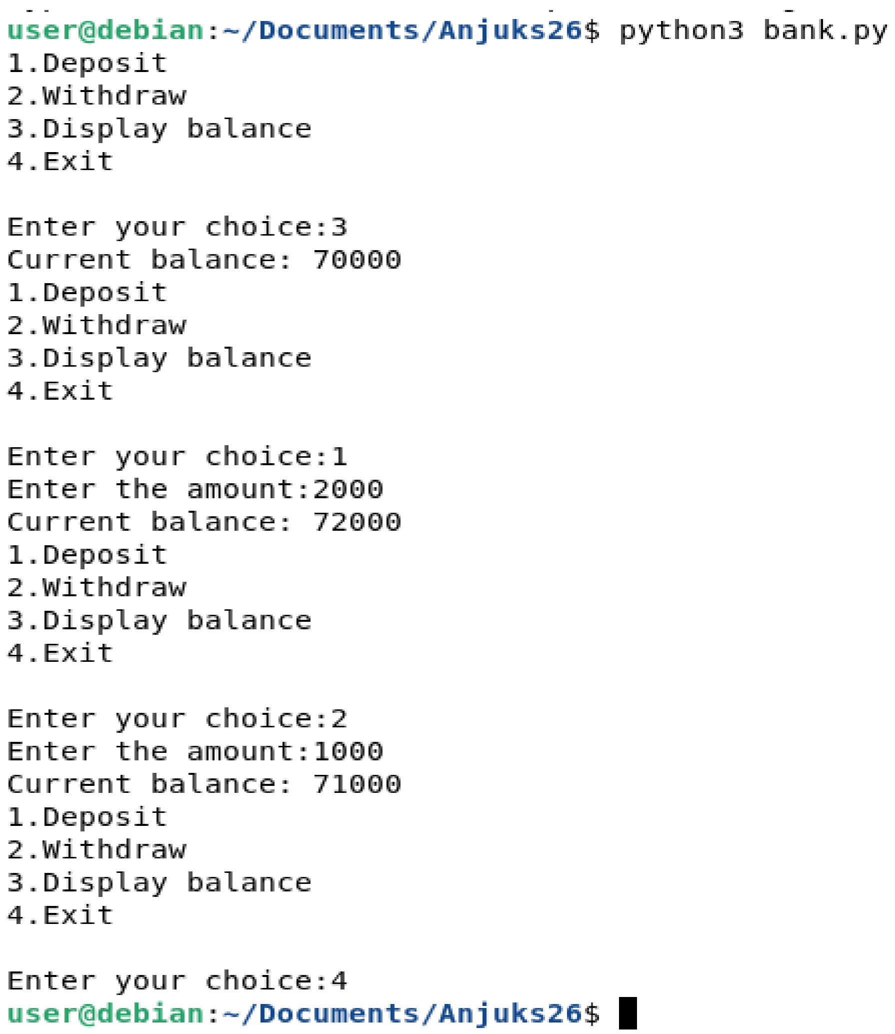
31. Create a Bank account with members account number, name, type of account and balance.Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

PROGRAM

```
class bank:
        def __init__(self,acc_no,name,acc_type,bal):
                self.acc_no=acc_no
                self.name=name
                self.acc_type=acc_type
                self.bal=bal

        def deposit(self,y):
                self.bal=self.bal+y
                return self.bal
        def withdraw(self,y):
                return self.bal-y
        def display_balance(self):
                return self.bal
acc1=bank("b11","Anju","Savings",70000)
while(1):
```

```
print("1.Deposit\n2.Withdraw\n3.Display balance\n4.Exit\n")
ch=int(input("Enter your choice:"))
if ch==1:
        amt=int(input("Enter the amount:"))
        b=acc1.deposit(amt)
        print("Current balance:",b)
elif ch==2:
        amt=int(input("Enter the amount:"))
        b=acc1.withdraw(amt)
        print("Current balance:",b)
elif ch==3:
        cb=acc1.display_balance()
        print("Current balance:",cb)
elif ch==4:
        exit(1)
else:
        print("Invalid choice")
```

OUTPUT



```
user@debian:~/Documents/Anjuks26$ python3 bank.py
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:3
Current balance: 70000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:1
Enter the amount:2000
Current balance: 72000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:2
Enter the amount:1000
Current balance: 71000
1.Deposit
2.Withdraw
3.Display balance
4.Exit

Enter your choice:4
user@debian:~/Documents/Anjuks26$ 
```

AIM

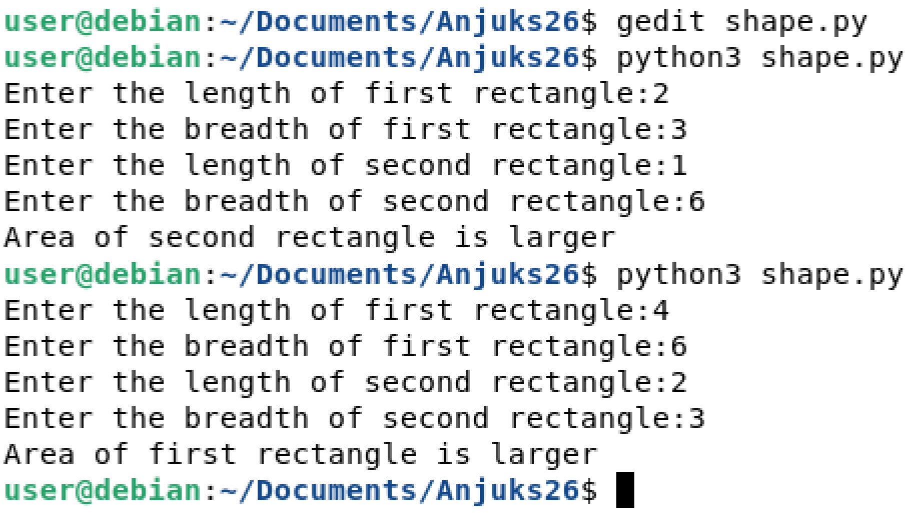
32. Create a class Rectanglr with private attributes length and width.Overload ‘ <’ operator to compare the area of 2 rectangles.

PROGRAM

```
class rectangle:
        def __init__(self,length,breadth):
                self.length=length
                self.breadth=breadth
        def __lt__(self,rect2):
```

```
        if self.length*self.breadth < rect2.length*rect2.breadth:
            return True
        else:
            return False

l=int(input("Enter the length of first rectangle:"))
b=int(input("Enter the breadth of first rectangle:"))
rect1 = rectangle(l,b)

l=int(input("Enter the length of second rectangle:"))
b=int(input("Enter the breadth of second rectangle:"))
rect2 = rectangle(l,b)

if  rect1 > rect2 :
  print("Area of first rectangle is larger")
else:
  print("Area of second rectangle is larger")
  print("Area of second rectangle is larger")
```

OUTPUT



```
user@debian:~/Documents/Anjuks26$ gedit shape.py
user@debian:~/Documents/Anjuks26$ python3 shape.py
Enter the length of first rectangle:2
Enter the breadth of first rectangle:3
Enter the length of second rectangle:1
Enter the breadth of second rectangle:6
Area of second rectangle is larger
user@debian:~/Documents/Anjuks26$ python3 shape.py
Enter the length of first rectangle:4
Enter the breadth of first rectangle:6
Enter the length of second rectangle:2
Enter the breadth of second rectangle:3
Area of first rectangle is larger
user@debian:~/Documents/Anjuks26$ 
```

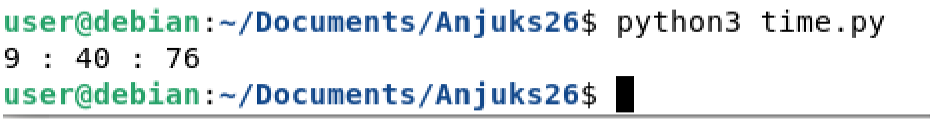
AIM

33. 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__(t1,t2):
                hr=t1.__hr+t2.__hr
                min=t1.__min+t2.__min
                sec=t1.__sec+t2.__sec
                print(hr,":",min,":",sec)
t1=Time(4,30,56)
t2=Time(5,10,20)
t1+t2
```

OUTPUT



```
user@debian:~/Documents/Anjuks26$ python3 time.py
9 : 40 : 76
user@debian:~/Documents/Anjuks26$ 
```

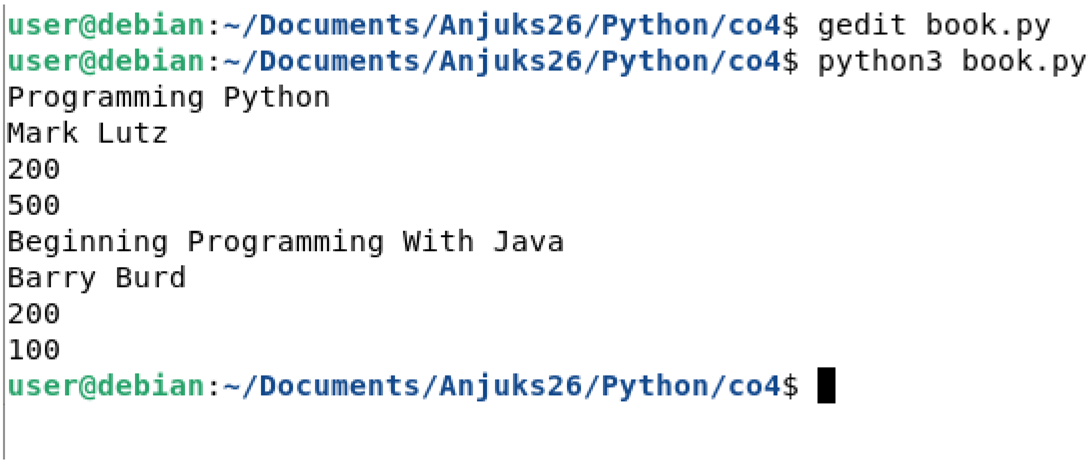
AIM

34 . 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(object):
    def __init__(self,name):
        self.name=name
    def display1(self):
        print(self.title)
        print(self.author)


class Book(Publisher):
    def __init__(self,name,title,author):
        super().__init__(name)
        self.title=title
        self.author=author
    def display2(self):
        #super().display1()
        print(self.title)
        print(self.author)
class Python(Book):
    def __init__(self,name,title,author,price,no_of_pages):
        super().__init__(name,title,author)
        self.price=price
        self.no_of_pages=no_of_pages
    def display3(self):
        super().display2()
        print(self.price)
        print(self.no_of_pages)
p=Python("AB Publications","Programming Python","Mark Lutz",200,500)
p.display3()
q=Python("CH Publications","Beginning Programming With Java","Barry Burd",200,100)
q.display3()
```

OUTPUT



```
user@debian:~/Documents/Anjuks26/Python/co4$ gedit book.py
user@debian:~/Documents/Anjuks26/Python/co4$ python3 book.py
Programming Python
Mark Lutz
200
500
Beginning Programming With Java
Barry Burd
200
100
user@debian:~/Documents/Anjuks26/Python/co4$ 
```

COURSE OUTCOME 5

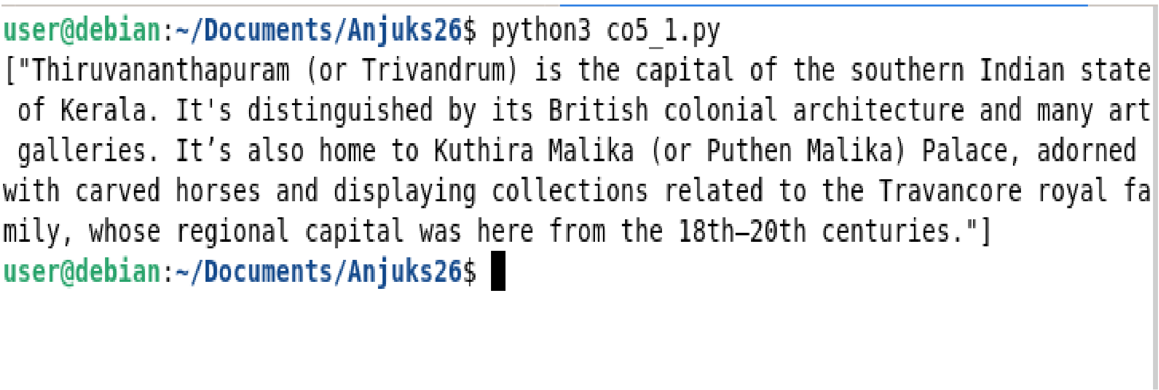
AIM

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

PROGRAM

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

OUTPUT



```
user@debian:~/Documents/Anjuks26$ python3 co5_1.py
["Thiruvananthapuram (or Trivandrum) is the capital of the southern Indian state
 of Kerala. It's distinguished by its British colonial architecture and many art
 galleries. It’s also home to Kuthira Malika (or Puthen Malika) Palace, adorned
with carved horses and displaying collections related to the Travancore royal fa
mily, whose regional capital was here from the 18th–20th centuries."]
user@debian:~/Documents/Anjuks26$
```

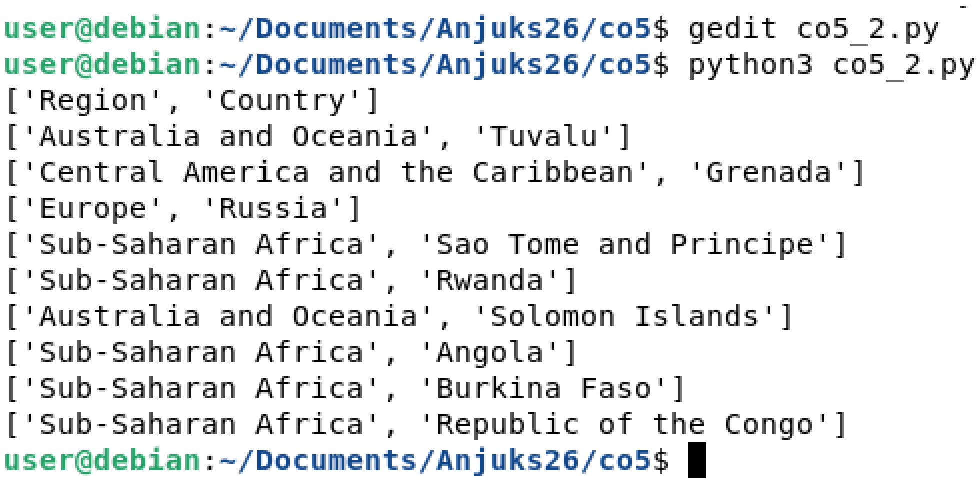
AIM

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

PROGRAM

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

OUTPUT



```
user@debian:~/Documents/Anjuks26/co5$ gedit co5_2.py
user@debian:~/Documents/Anjuks26/co5$ python3 co5_2.py
['Region', 'Country']
['Australia and Oceania', 'Tuvalu']
['Central America and the Caribbean', 'Grenada']
['Europe', 'Russia']
['Sub-Saharan Africa', 'Sao Tome and Principe']
['Sub-Saharan Africa', 'Rwanda']
['Australia and Oceania', 'Solomon Islands']
['Sub-Saharan Africa', 'Angola']
['Sub-Saharan Africa', 'Burkina Faso']
['Sub-Saharan Africa', 'Republic of the Congo']
user@debian:~/Documents/Anjuks26/co5$
```