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

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ANSA BENNY

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 30**

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 **ANSA BENNY (FIT21MCA-2030)** 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 Internal Examiner	Signature of External Examiner

CONTENT

34	13-01-2022	Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.	26-27	
35	20-01-2022	Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.	27	
36	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.	28-29	
CO5				
37	03-02-2022	Write a Python program to read a file line by line and store it into a list.	30	
38	03-02-2022	Write a Python program to read each row from a given csv file and print a list of string.	31	

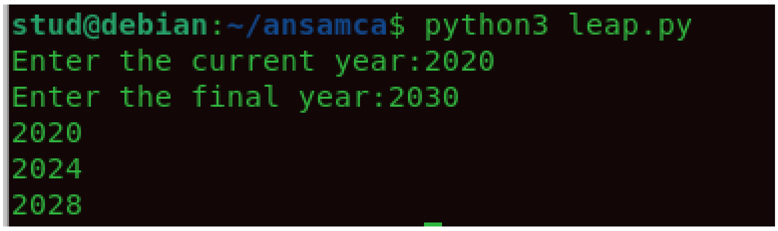
COURSE OUTCOME 1

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

Source code

```
currentyear=int(input("Enter the current year:"))
finalyear=int(input("Enter the final year:"))
for year in range(currentyear,finalyear):
        if(year%400==0)or(year%100!=0)and(year%4==0):
                print(year)
```

Output



```
stud@debian:~/ansamca$ python3 leap.py
Enter the current year:2020
Enter the final year:2030
2020
2024
2028
```

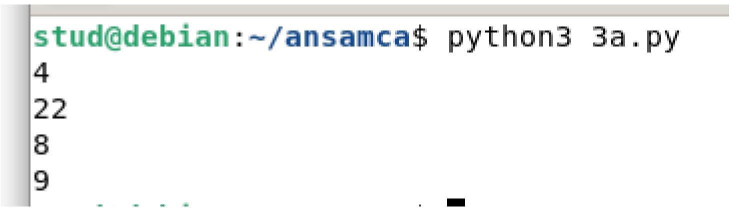
2) List comprehensions:

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

Source code

```
list1=[4,22,8,9]
for num in list1:
      if num>=0:
            print(num)
```

Output



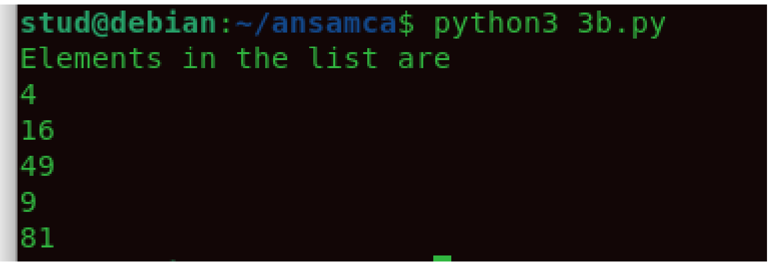
```
stud@debian:~/ansamca$ python3 3a.py
4
22
8
9
```

b. Square of N numbers

Source code

```
print("Elements in the list are")
list1=[2,4,7,3,9]
for s in list1:
     num=s*s
     print(num)
```

Output



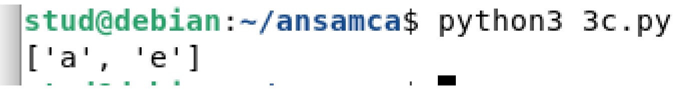
```
stud@debian:~/ansamca$ python3 3b.py
Elements in the list are
4
16
49
9
81
```

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

Source code

```
L=[]
s=("apple")
for i in s:
         if i in "AaEeIiOoUu":
                   L.append(i)
print(L)
```

Output



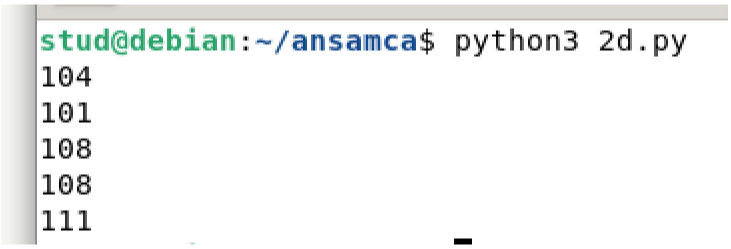
```
stud@debian:~/ansamca$ python3 3c.py
['a', 'e']
```

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

Source code

```
s=("hello")
for i in s:
        print(ord(i))
```

Output



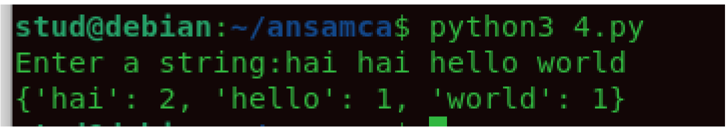
```
stud@debian:~/ansamca$ python3 2d.py
104
101
108
108
111
```

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

Source code

```
 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



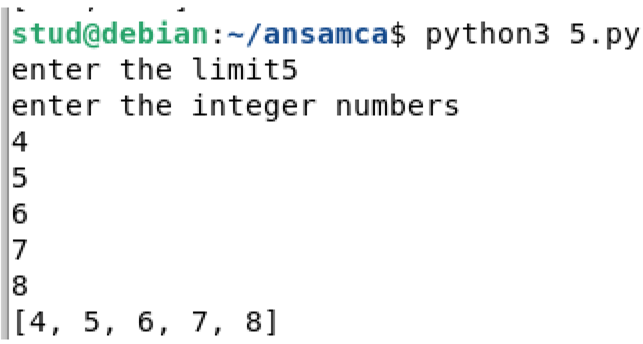
```
stud@debian:~/ansamca$ python3 4.py
Enter a string:hai hai hello world
{'hai': 2, 'hello': 1, 'world': 1}
```

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

Source code

```
list=[]
n=int(input(“enter the limit”))
print("Enter the integer numbers:")
for i in range(0,n):
                    j=int(input())
                    if j>=100:
                           list.append('over')
                   else:
                           list.append(j)
print(list)
```

Output



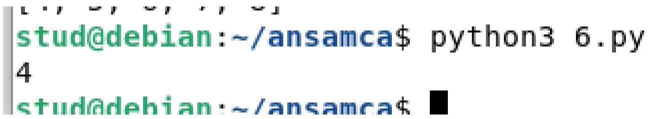
```
stud@debian:~/ansamca$ python3 5.py
enter the limit5
enter the integer numbers
4
5
6
7
8
[4, 5, 6, 7, 8]
```

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

Source code

```
 l=[]
 r=0
 s=(“anu,anna,ann”)
for i in s:
       if i in (“aA”):
              r=r+1
              l.append(i)
 print(r)
```

Output



```
stud@debian:~/ansamca$ python3 6.py
4
stud@debian:~/ansamca$ █
```

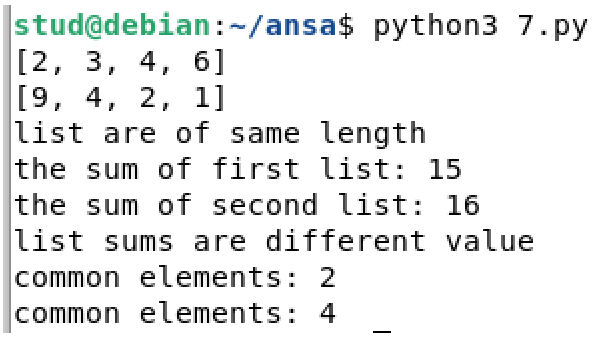
6) Enter 2 lists of integers.Check

- **a. whether list are of same length**
- **b. whetherlist sums of same value**
- **c. whether any value occur in both.**

Source code

```
 l1=[2,6,3,9,]
 l2=[4,5,6]
 print(l1)
 print(l2)
 x=len(l1)
 y=len(l2)
 if x==y:
              print("list are of same length")
else:
              print("list are of different length")
s1=0
s2=0
for i in range(x):
             s1=s1+l1[i]
print("the sum of 1st list:",s1)
for j in range(y):
             s2=s2+l2[j]
print("The sum of second list:",s2)
if s1==s2:
             print("sum of list are same")
else:
             print("sum of list are different")
flag==0;
for i in l1:
             if I in l2:
                    print("common elements:"i)
                    flag=1
if flag==0:
             print("no common elements")
```

Output



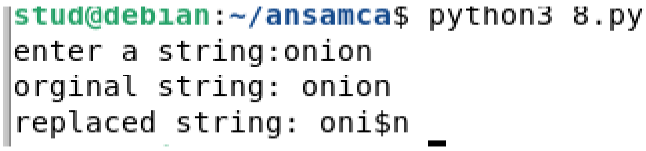
```
stud@debian:~/ansa$ python3 7.py
[2, 3, 4, 6]
[9, 4, 2, 1]
list are of same length
the sum of first list: 15
the sum of second list: 16
list sums are different value
common elements: 2
common elements: 4  _
```

7) Get a string from an input string where all occurrences of first character replaced with ‘$’,except first character.[eg:onion->oni$n]

Source code

```
str1=input("Enter the string:")
print("Original string:",str1)
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print("Replaced String:",str1)
```

Output



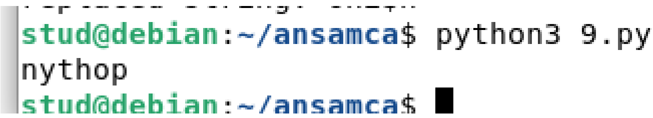
```
stud@debian:~/ansamca$ python3 8.py
enter a string:onion
orginal string: onion
replaced string: oni$n _
```

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

Source code

```
S=”python”
t=s[0]
t1=s[-1]
n=len(s)
ns=t1+s[1:n-1]+t
print("Replaced string:",ns);
```

Output



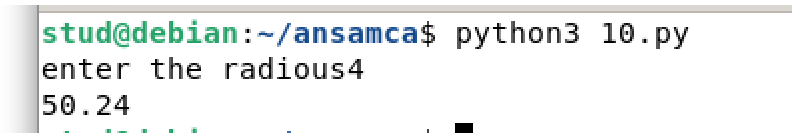
```
stud@debian:~/ansamca$ python3 9.py
nythop
stud@debian:~/ansamca$
```

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

Source code

```
p=int(input("Enter the radius"))
a=3.14*p*p
print(a)
```

Output



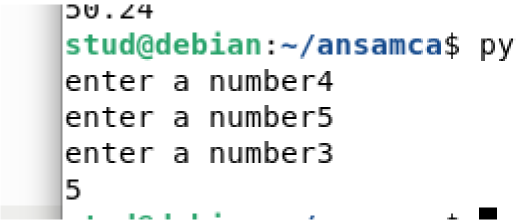
```
stud@debian:~/ansamca$ python3 10.py
enter the radious4
50.24
```

10) Find the biggest of 3 numbers

Source code

```
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



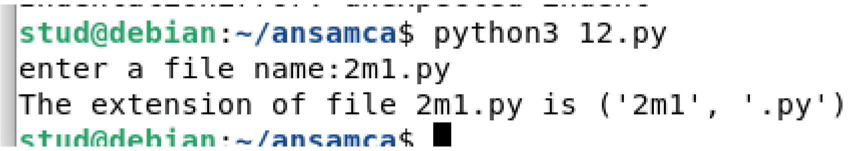
```
50.24
stud@debian:~/ansamca$ python3 11.py
enter a number4
enter a number5
enter a number3
5
```

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

Source code

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

Output



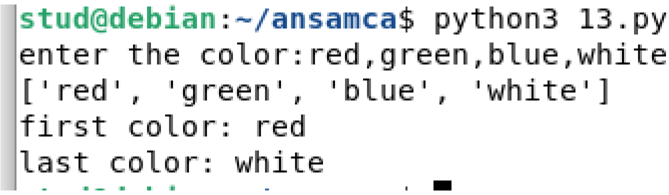
```
stud@debian:~/ansamca$ python3 12.py
enter a file name:2m1.py
The extension of file 2m1.py is ('2m1', '.py')
stud@debian:~/ansamca$ 
```

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

Source code

```
colors=[]
str=(input("Enter the color :"))
for i in str.split(','):
 colors.append(i)
print(colors)
print("first color:",colors[0],"Last color:",colors[-1])
```

Output



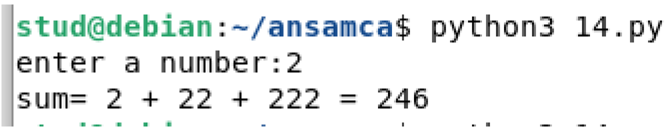
```
stud@debian:~/ansamca$ python3 13.py
enter the color:red,green,blue,white
['red', 'green', 'blue', 'white']
first color: red
last color: white
```

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

Source code

```
n=input("Enter a number:")
a=n*1
b=n*11
c=n*111
s1=a+b+c
print(“sum=”,a,”+”,b,”+”,c,”=”,sum)
```

Output



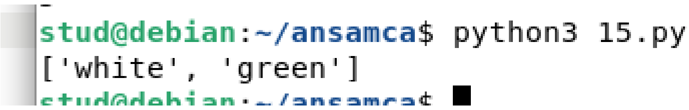
```
stud@debian:~/ansamca$ python3 14.py
enter a number:2
sum= 2 + 22 + 222 = 246
```

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

Source code

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

Output



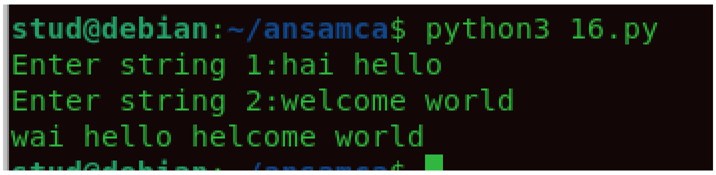
```
stud@debian:~/ansamca$ python3 15.py
['white', 'green']
stud@debian:~/ansamca$
```

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

Source 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



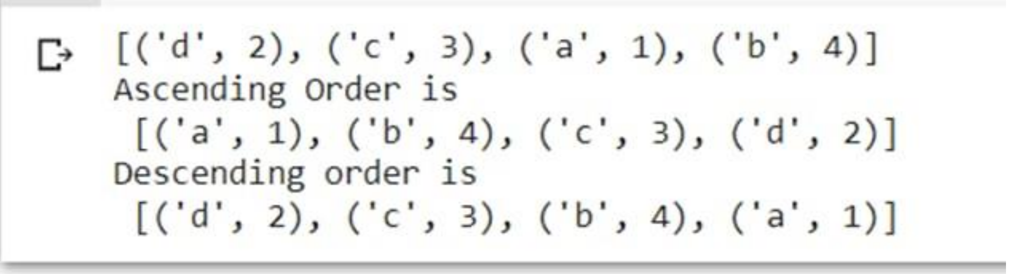
```
stud@debian:~/ansamca$ python3 16.py
Enter string 1:hai hello
Enter string 2:welcome world
wai hello helcome world
stud@debian:~/ansamca$
```

16) Sort dictionary in ascending and descending order.

Source code

```
dict1={"a":1,"c":3,"d":2,"b":4}
l=list(dict1.items())
print(l)
 l.sort()
 print("Ascending Order is \n",l)
 l=list(dict1.items())
 l.sort(reverse=True)
 print("Descending order is \n",l)
```

output



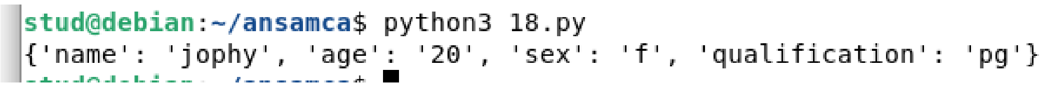
```
[('d', 2), ('c', 3), ('a', 1), ('b', 4)]
Ascending Order is
 [('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
 [('d', 2), ('c', 3), ('b', 4), ('a', 1)]
```

17) Merge two dictionaries.

Source code

```
d1={"name":"jophy","age":"20"}
d2={"sex":"f","qualification":"pg"}
d1.update(d2)
print(d1)
```

Output



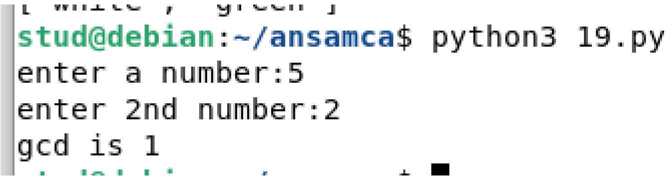
```
stud@debian:~/ansamca$ python3 18.py
{'name': 'jophy', 'age': '20', 'sex': 'f', 'qualification': 'pg'}
```

18) Find gcd of 2 numbers

Source code

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

Output



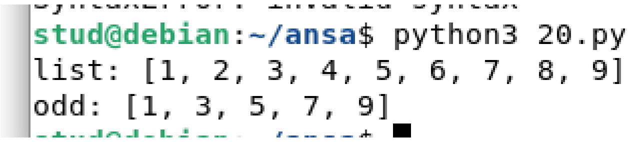
```
stud@debian:~/ansamca$ python3 19.py
enter a number:5
enter 2nd number:2
gcd is 1
```

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

Source code

```
list1=[1,2,3,4,5,6,7,8,9]
num=[]
print(list1)
for i in list1:
        if(i%2!=0):
                num.append(i)
print(num)
```

Output



```
SyntaxError: invalid syntax
stud@debian:~/ansa$ python3 20.py
list: [1, 2, 3, 4, 5, 6, 7, 8, 9]
odd: [1, 3, 5, 7, 9]
stud@debian:~/ansa$
```

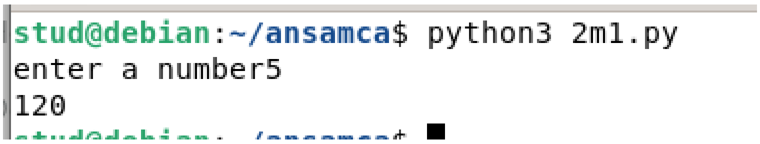
COURSE OUTCOME 2

1) Program to find the factorial of a number.

Source code

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

Output



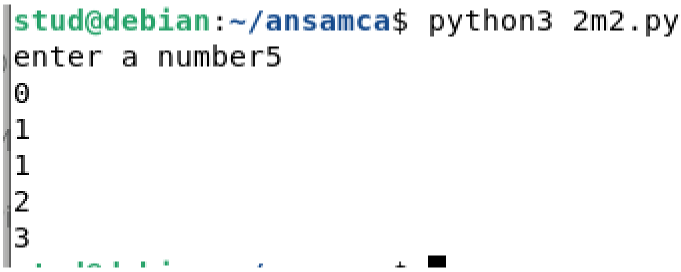
```
stud@debian:~/ansamca$ python3 2m1.py
enter a number5
120
stud@debian:~/ansamca$ ■
```

2) Generate fibonacci series of N terms.

Source code

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

Output



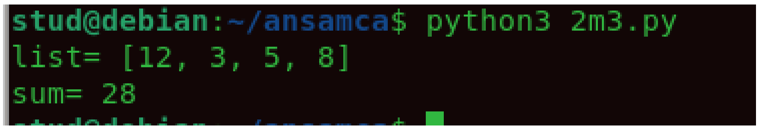
```
stud@debian:~/ansamca$ python3 2m2.py
enter a number5
0
1
1
2
3
```

3) Find the sum of all items in a list.

Source code

```
list=[12,3,5,8]
print("list=",list)
sum=0
for i in list:
        sum=sum+i
print("sum=",sum)
```

Output



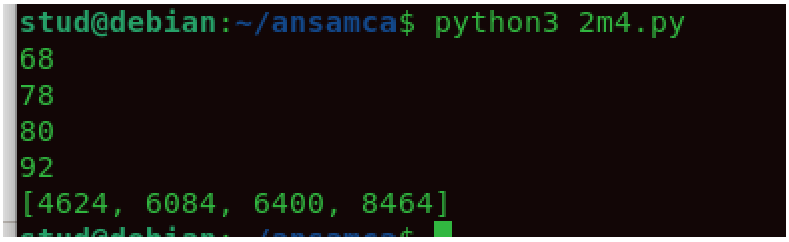
```
stud@debian:~/ansamca$ python3 2m3.py
list= [12, 3, 5, 8]
sum= 28
```

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

Source code

```
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



```
stud@debian:~/ansamca$ python3 2m4.py
68
78
80
92
[4624, 6084, 6400, 8464]
stud@debian:~/ansamca$
```

5) Display the given pyramid with step number accepted from user.

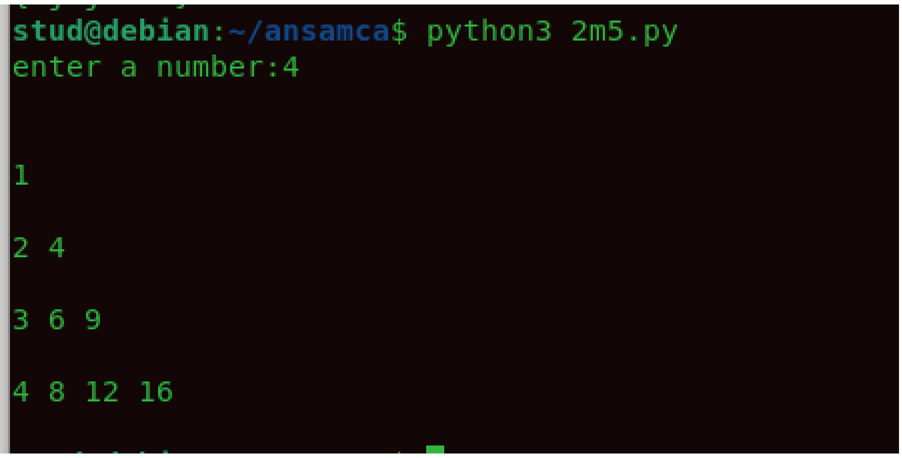
Eg n=4

```
 1
2   4
3   6    9
4   8    12   16
```

Source code

```
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



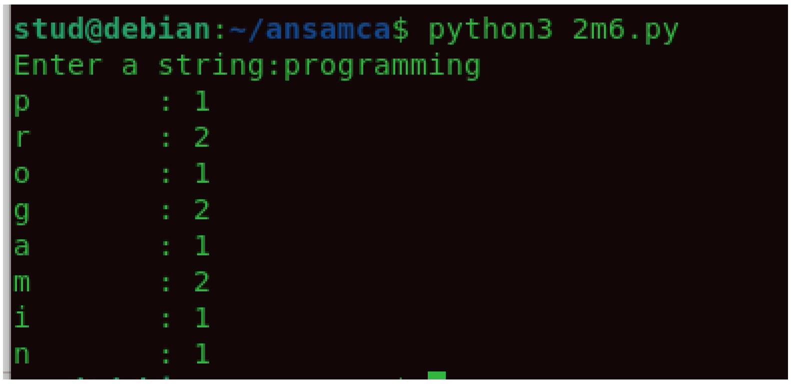
```
stud@debian:~/ansamca$ python3 2m5.py
enter a number:4


1

2 4

3 6 9

4 8 12 16
```

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

Source code

```
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



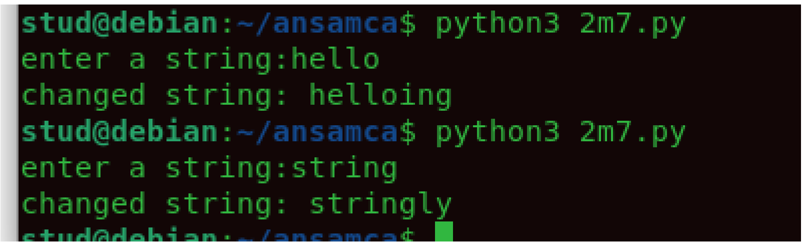
```
stud@debian:~/ansamca$ python3 2m6.py
Enter a string:programming
p       : 1
r       : 2
o       : 1
g       : 2
a       : 1
m       : 2
i       : 1
n       : 1
```

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

Source code

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

Output



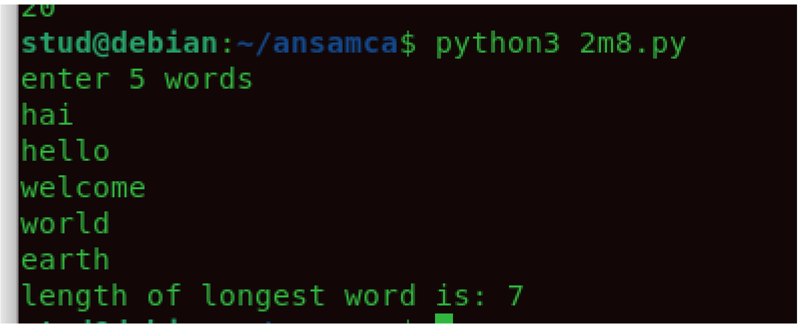
```
stud@debian:~/ansamca$ python3 2m7.py
enter a string:hello
changed string: helloing
stud@debian:~/ansamca$ python3 2m7.py
enter a string:string
changed string: stringly
stud@debian:~/ansamca$ 
```

8) Accept a list of words and return length of longest word.

Source code

```
list=[]
length=[]
print("enter 5 words")
for i in range (5):
  str=input()
  list.append(str)
for j in list:
  length.append(len(j))
print("length of longest word is:",max(length))
```

Output



```
20
stud@debian:~/ansamca$ python3 2m8.py
enter 5 words
hai
hello
welcome
world
earth
length of longest word is: 7
```

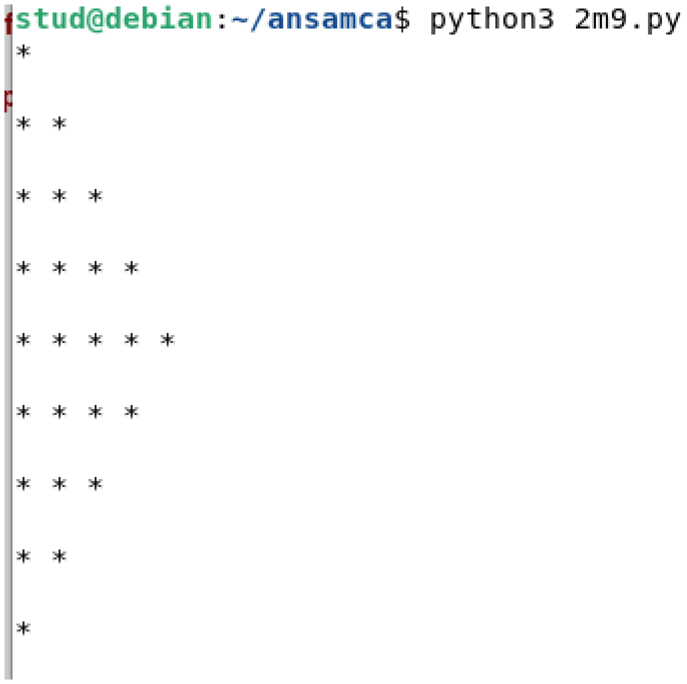
9) Construct following pattern using nested loop.

```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

Source code

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

output



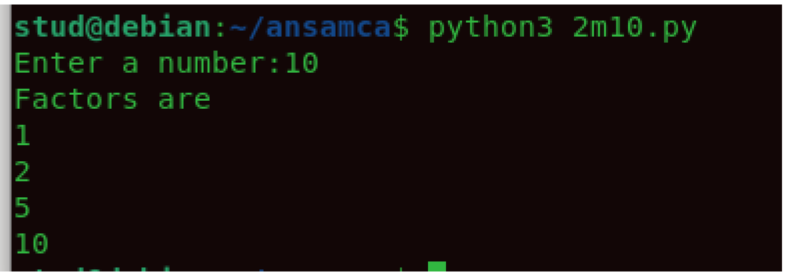
```
stud@debian:~/ansamca$ python3 2m9.py
*

*  *

*  *  *

*  *  *  *

*  *  *  *  *

*  *  *  *

*  *  *

*  *

*
```

10) Generate all factors of a number.

Source code

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

Output



```
stud@debian:~/ansamca$ python3 2m10.py
Enter a number:10
Factors are
1
2
5
10
```

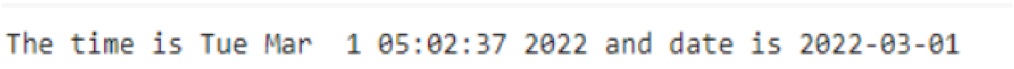
COURSE OUTCOME 3

1) Work with built-in packages.

SOURCE CODE

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

output



```
The time is Tue Mar  1 05:02:37 2022 and date is 2022-03-01
```

2) 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)

Source code

Graphice\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
```

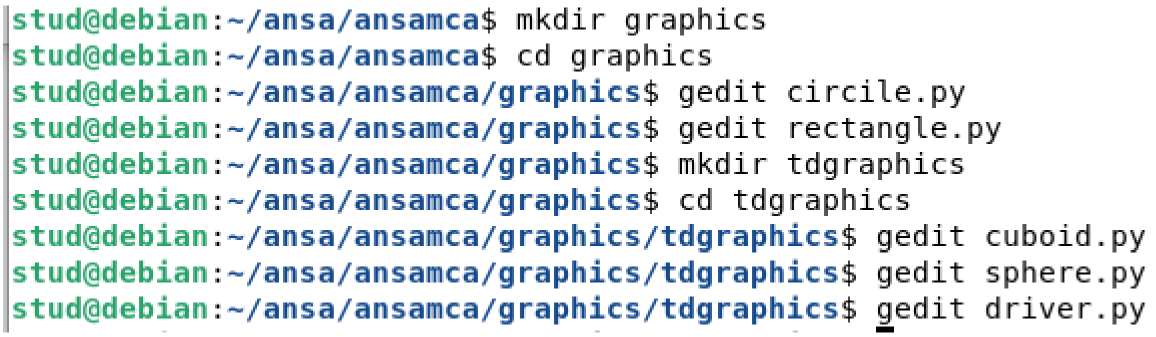
Graphics\tdgraphics\sphere.py

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

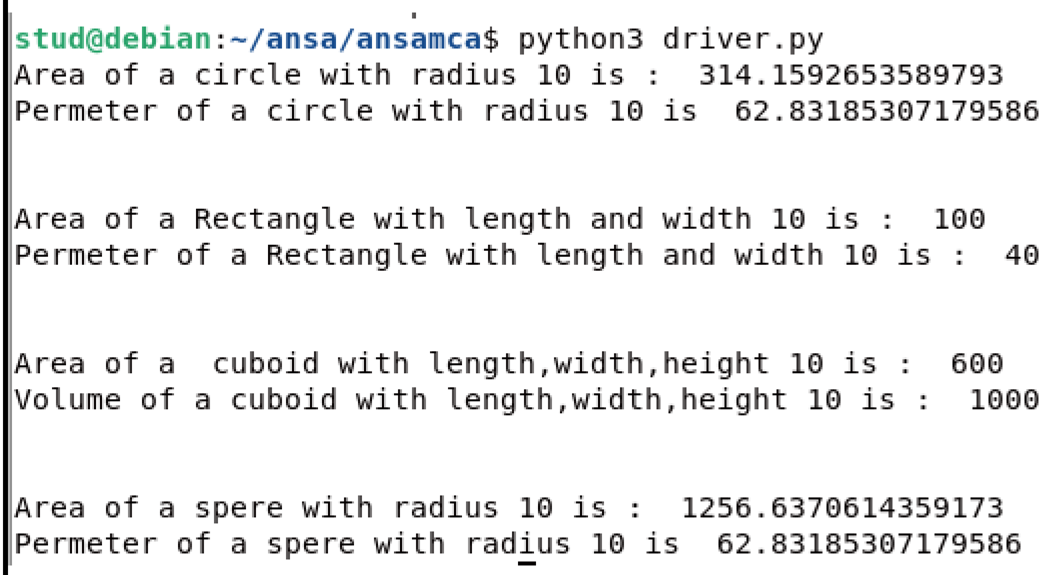
graphics.py (driver code)

```
import Graphics
from Graphics import circle,rectangle
from Graphics.tdgraphics import cuboid,sphere
from Graphics.circle import *
print("Area of a circle with radius 10 is : ",circle.area_circle(10))
print("Permeter of a circle with radius 10 is ",circle.perimeter_circle(10))
print("\n")

print("Area of a Rectangle with length and width 10 is : ",rectangle.area_rec(10,10))
print("Permeter of a Rectangle with length and width 10 is :
       ",rectangle.perimeter_rec(10,10))
print("\n")

print("Area of a  cuboid with length,width,height 10 is : ",cuboid.area_cuboid(10,10,10))
print("Volume of a cuboid with length,width,height 10 is :
       ",cuboid.volume_cuboid(10,10,10))
print("\n")

print("Area of a spere with radius 10 is : ",sphere.area_sphere(10))
print("Permeter of a spere with radius 10 is ",sphere.perimeter_sphere(10))
```

Output



```
stud@debian:~/ansa/ansamca$ mkdir graphics
stud@debian:~/ansa/ansamca$ cd graphics
stud@debian:~/ansa/ansamca/graphics$ gedit circile.py
stud@debian:~/ansa/ansamca/graphics$ gedit rectangle.py
stud@debian:~/ansa/ansamca/graphics$ mkdir tdgraphics
stud@debian:~/ansa/ansamca/graphics$ cd tdgraphics
stud@debian:~/ansa/ansamca/graphics/tdgraphics$ gedit cuboid.py
stud@debian:~/ansa/ansamca/graphics/tdgraphics$ gedit sphere.py
stud@debian:~/ansa/ansamca/graphics/tdgraphics$ gedit driver.py
```



```
stud@debian:~/ansa/ansamca$ python3 driver.py
Area of a circle with radius 10 is :  314.1592653589793
Permeter of a circle with radius 10 is  62.83185307179586


Area of a Rectangle with length and width 10 is :  100
Permeter of a Rectangle with length and width 10 is :  40


Area of a  cuboid with length,width,height 10 is :  600
Volume of a cuboid with length,width,height 10 is :  1000


Area of a spere with radius 10 is :  1256.6370614359173
Permeter of a spere with radius 10 is  62.83185307179586
```

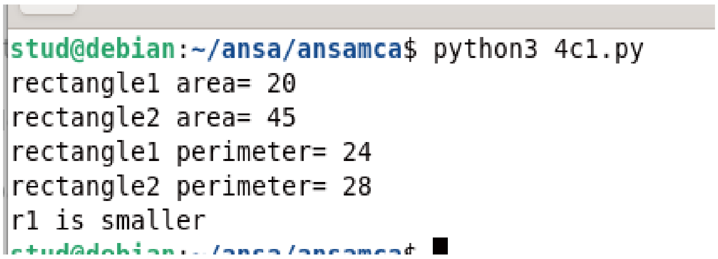
COURSE OUTCOME 4

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

Source code

```
class Rectangle():
  def __init__(self,l,b):
    self.length=l
    self.breadth=b
  def area(self):
    return self.length*self.breadth
  def peri(self):
    return 2*(self.length+self.breadth)
r1=Rectangle(10,2)
r2=Rectangle(5,9)
x=r1.area()
y=r2.area()
m=r1.peri()
n=r2.peri()
print("rectangle1 area=",x)
print("rectangle2 area=",y)
print("rectangle1 perimeter=",m)
print("rectangle2 perimeter=",n)
if(x<y):
  print("r1 is smaller")
else:
  print("r2 is smaller")
```

Output



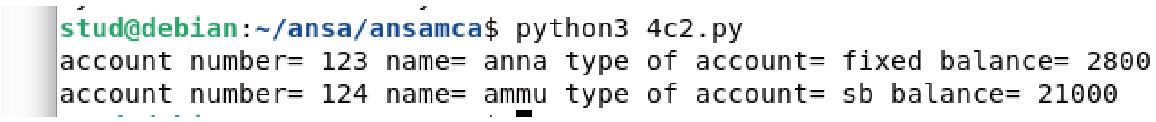
```
stud@debian:~/ansa/ansamca$ python3 4c1.py
rectangle1 area= 20
rectangle2 area= 45
rectangle1 perimeter= 24
rectangle2 perimeter= 28
r1 is smaller
stud@debian:~/ansa/ansamca$ █
```

2) **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.**

Source code

```
class bank:
  def __init__(self,account_no,name,type_of_account,balance):
    self.acno=account_no
    self.name=name
    self.toa=type_of_account
    self.b=balance
  def withdraw(self,x):
    self.b=self.b-x
  def deposit(self,y):
    self.b=self.b+y
  def print(self):
    print("account number=",self.acno,"name=",self.name,"type of account=",self.toa,"balance=",self.b)
acc1=bank(123,"anna","fixed",3800)
acc2=bank(124,"ammu","sb",1000)
acc1.withdraw(1000)
acc2.deposit(20000)
acc1.print()
acc2.print()
```

Output



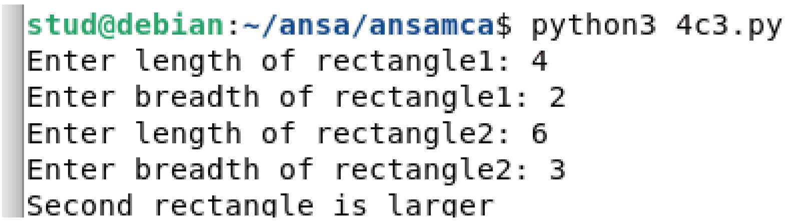
```
stud@debian:~/ansa/ansamca$ python3 4c2.py
account number= 123 name= anna type of account= fixed balance= 2800
account number= 124 name= ammu type of account= sb balance= 21000
```

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

Source code

```
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 length of rectangle1: "))
b=int(input("Enter breadth of rectangle1: "))
rect1 = Rectangle(l,b)
l=int(input("Enter length of rectangle2: "))
b=int(input("Enter breadth of rectangle2: "))
rect2 = Rectangle(l,b)
if rect1 < rect2:
  print("Second rectangle is larger")
else:
  print("First rectangle is larger")
```

output



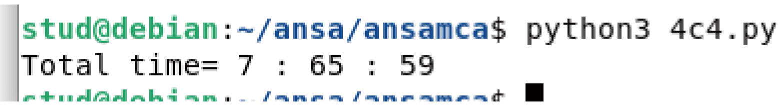
```
stud@debian:~/ansa/ansamca$ python3 4c3.py
Enter length of rectangle1: 4
Enter breadth of rectangle1: 2
Enter length of rectangle2: 6
Enter breadth of rectangle2: 3
Second rectangle is larger
```

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

Source code

```
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(3,45,56)
t2=Time(4,20,3)
t1+t2
```

Output



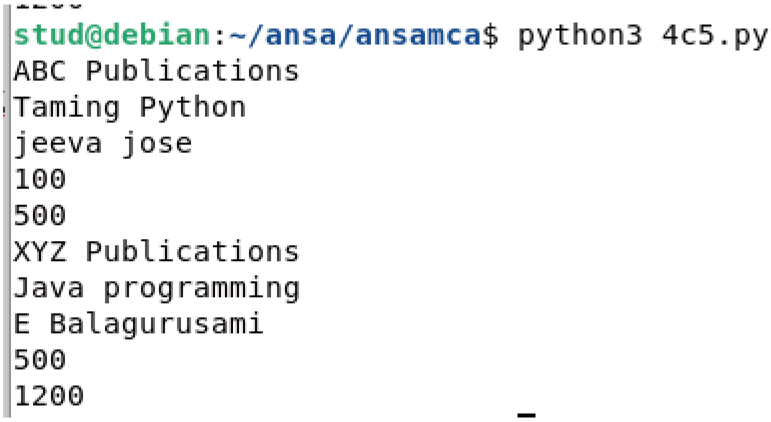
```
stud@debian:~/ansa/ansamca$ python3 4c4.py
Total time= 7 : 65 : 59
stud@debian:~/ansa/ansamca$
```

5) 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.

Source code

```
class Publisher(object):
    def __init__(self,name):
        self.name=name
    def display1(self):
        print(self.name)
        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.name)
        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("ABC Publications","Taming Python","jeeva jose",100,500)
p.display3()
q=Python("XYZ Publications","Java programming","E Balagurusami",500,1200)
q.display3()
```

Output



```
stud@debian:~/ansa/ansamca$ python3 4c5.py
ABC Publications
Taming Python
jeeva jose
100
500
XYZ Publications
Java programming
E Balagurusami
500
1200
```

COURSE OUTCOME 5

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

Source code

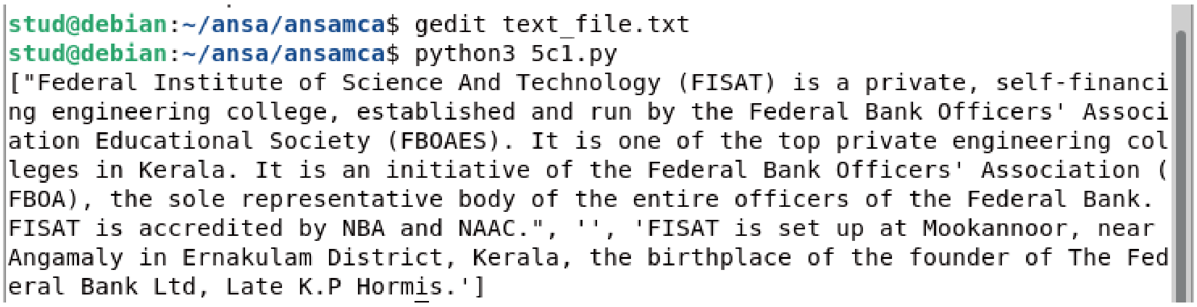
text_file.txt

Federal Institute of Science And Technology (FISAT) is a private, self-financing engineering college, established and run by the Federal Bank Officers' Association Educational Society (FBOAES). It is one of the top private engineering colleges in Kerala. It is an initiative of the Federal Bank Officers' Association (FBOA), the sole representative body of the entire officers of the Federal Bank. FISAT is accredited by NBA and NAAC.
FISAT is set up at Mookannoor, near Angamaly in Ernakulam District, Kerala, the birthplace of the founder of The Federal Bank Ltd, Late K.P Hormis.

5c1.py

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

output



```
stud@debian:~/ansa/ansamca$ gedit text_file.txt
stud@debian:~/ansa/ansamca$ python3 5c1.py
["Federal Institute of Science And Technology (FISAT) is a private, self-financi
ng engineering college, established and run by the Federal Bank Officers' Associ
ation Educational Society (FBOAES). It is one of the top private engineering col
leges in Kerala. It is an initiative of the Federal Bank Officers' Association (
FBOA), the sole representative body of the entire officers of the Federal Bank.
FISAT is accredited by NBA and NAAC.", '', 'FISAT is set up at Mookannoor, near
Angamaly in Ernakulam District, Kerala, the birthplace of the founder of The Fed
eral Bank Ltd, Late K.P Hormis.']
```

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

Source code

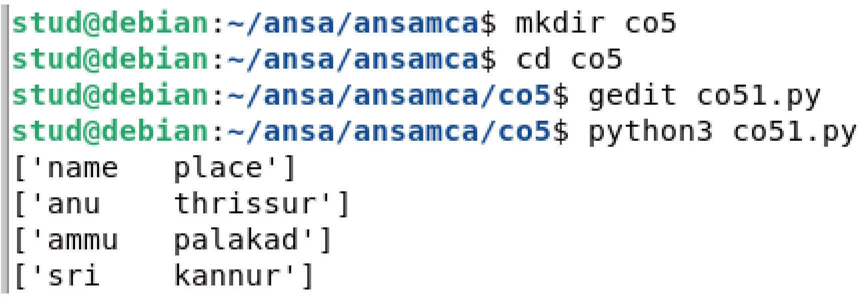
c05.csv

```
name   place
anu    thrissur
ammu   palakad
sri    kannur
```

co51.py

```
import csv
with open('c05.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

Output



```
stud@debian:~/ansa/ansamca$ mkdir co5
stud@debian:~/ansa/ansamca$ cd co5
stud@debian:~/ansa/ansamca/co5$ gedit co51.py
stud@debian:~/ansa/ansamca/co5$ python3 co51.py
['name   place']
['anu    thrissur']
['ammu   palakad']
['sri    kannur']
```