FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

‘FOCUS ON EXCELLENCE’

LABORATORY RECORD
20MCA131-PROGRAMMING LAB

Name: CHRISTEENA PAUL

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch:2021- A Roll No: 48

FEDERAL INSTITUTE OF

SCIENCE AND TECHNOLOGY

(FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

‘FOCUS ON EXCELLENCE’

Name :CHRISTEENA PAUL

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 Roll No: 48

University Exam.Reg. No: FIT21MCA-2048

<u>CERTIFICATE</u>

Certified that this is a Bonafide record of the Practical work done by **Ms.CHRISTEENA PAUL** in the **20MCA131-PROGRAMMING** Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.

Signature of Staff in Charge	Signature of H.O.D
Name:	Name:
Date:	

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

Signature of	Signature of
Internal Examiner	External Examiner

Sl No :	Date	Name of the Experiment	Page No:	Signature of Staff –In -Charge
1	**10/11/21**	Display future leap years from current year to a final year entered by user.		
2	**10/11/21**	List comprehensions:(a)Generate positive list of numbers from a given list of integers(b)Square of N numbers(c)Form a list of vowels selected from a given word(d)List ordinal value of each element of a word		
3	**25/11/21**	Count the occurrences of each word in a line of text.		
4	**11/11/21**	Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.		
5	**11/11/21**	Store a list of first names. Count the occurrences of ‘a’ within the list		
6	**17/11/21**	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		
7	**25/11/21**	Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.		
8	**25/11/21**	Create a string from given string where first and last characters exchanged.		
9	**28/11/21**	Accept the radius from user and find area of circle.		
10	**28/11/21**	Find biggest of 3 numbers entered.		
11	**25/11/21**	Accept a file name from user and print extension of that.		
12	**17/11/21**	Create a list of colors from comma-separated color names entered by user. Display first and last colors.		
13	**2/12/21**	Accept an integer n and compute n+nn+nun.		
14	**17/11/21**	Print out all colors from color-list 1 not contained in color-list2.		
15	**2/12/21**	Create a single string separated with space from two strings by swapping the character at position 1.		

16	**2/12/21**	Sort dictionary in ascending and descending order.		
17	**2/12/21**	Merge two dictionaries.		
18	**25/11/21**	Find gcd of 2 numbers.		
19	**17/11/21**	From a list of integers, create a list removing even numbers.		
20	**28/10/21**	Program to find the factorial of a number		
21	**28/10/21**	Generate Fibonacci series of N terms		
22	**2/12/21**	Find the sum of all items in a list		
23	**2/12/21**	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.		
24	**2/12/21**	Display the given pyramid with step number accepted from user		
25	**2/12/21**	Count the number of characters (characterfrequency) in a string.		
26	**2/12/21**	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’		
27	**2/12/21**	Accept a list of words and return length of longest word.		
28	**2/12/21**	Construct following pattern using nested loop		
29	**2/12/21**	Generate all factors of a number.		
30	**20/01/22**	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.		
31	**9/12/21**	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.		
32	**9/12/21**	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.		
33	**16/12/21**	Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.		

34	**13/01/22**	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.		
35	**20/01/22**	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 ages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.		
36	**27/01/22**	Write a Python program to read a file line by line and store it into a list.		
37	**27/01/22**	Write a Python program to read each row from a given csv file and print a list of strings.		

Course Outcome 1(CO1)

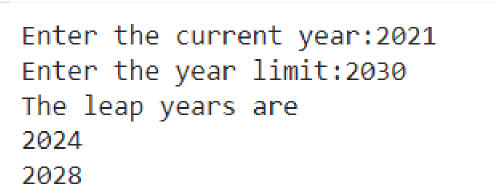
AIM

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

Input

```
    #leap
print ("Enter last year")
endYear = int(input())
print ("List of leap years:")
for year in range(2021, endYear+1):
        if (year % 4 == 0 ) and (year % 100 != 0) or (year % 400 == 0):
                print (year)
```

Output



```
Enter the current year:2021
Enter the year limit:2030
The leap years are
2024
2028
```

AIM

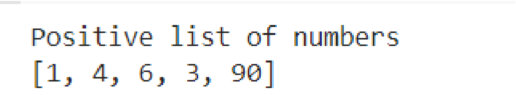
2.List comprehensions:

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

Input

```
#positive no in list
list1=[0,-1,-2,3,4,1,2,5]
        for num in list1:
               if (num>=0):
        print(num)
```

Output



```
Positive list of numbers
[1, 4, 6, 3, 90]
```

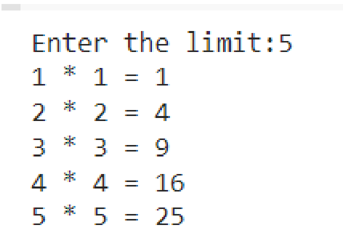
AIM

(b)Square of N numbers

Input

```
#Square of n numbers
list1=[3,56,6]
list2=[]
print("The squares of the given numbers are:")
for i in list1:
        s=(i*i) print(s)
```

Output



```
Enter the limit:5
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
```

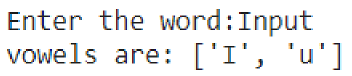
AIM

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

Input

```
#vowel in string
word=input('Enter the string:')
vowel=['a','e','i','o','u']
 list1=[]
 for d in word:
if(d in vowel and d not in list1):
        list1.append(d)
 print('vowels are:',list1)
```

Output



```
Enter the word:Input
vowels are: ['I', 'u']
```

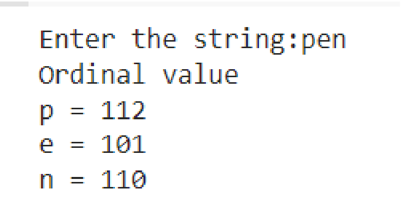
AIM

(d)List ordinal value of each element of a word

Input

```
#ordinal value
word=input('Enter the string:')
print([ord(x) for x in word])
```

Output



```
Enter the string:pen
Ordinal value
p = 112
e = 101
n = 110
```

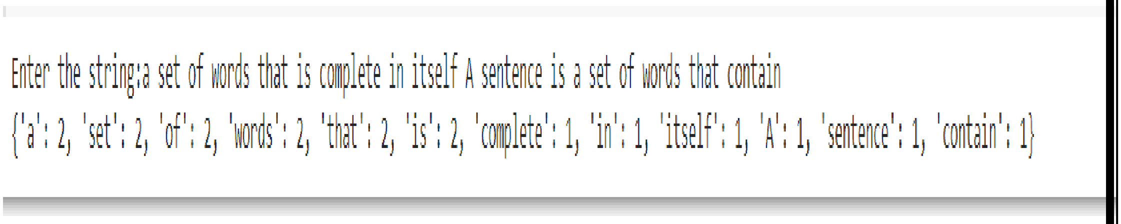
AIM

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

Input

```
list1=[]

list2=[]

x=input("Enter a string:")

for i in x.split(" "):

        list1.append(i)

        if i not in list2:

                list2.append(i)

for i in list2:

        print(i,"\t",list1.count(i))
```

Output



```
Enter the string:a set of words that is complete in itself A sentence is a set of words that contain
{'a': 2, 'set': 2, 'of': 2, 'words': 2, 'that': 2, 'is': 2, 'complete': 1, 'in': 1, 'itself': 1, 'A': 1, 'sentence': 1, 'contain': 1}
```

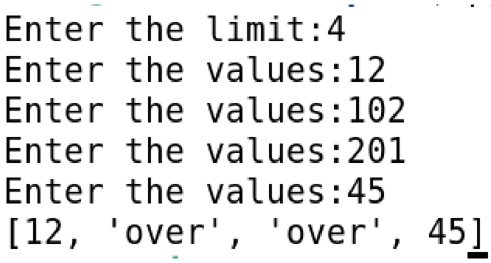
AIM

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

Input

```
#store over for values>100
w=int(input('Enter the limit:'))
list1=[]
for i in range(0,w):
    value=int(input('Enter the values:'))
    if(value>100):
        list1.append('over')
    else:
        list1.append(value)
print(list1)
```

Output



```
Enter the limit:4
Enter the values:12
Enter the values:102
Enter the values:201
Enter the values:45
[12, 'over', 'over', 45]
```

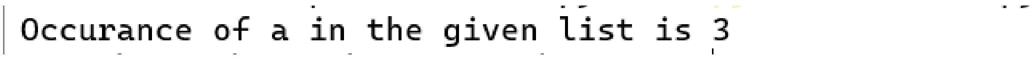
AIM

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

Input

```
#counta word=['anna','anu']
r=0
for d in word:
    for c in d:
        if (c=='a'):
            r=r+1
print(r)
```

Output



```
Occurance of a in the given list is 3
```

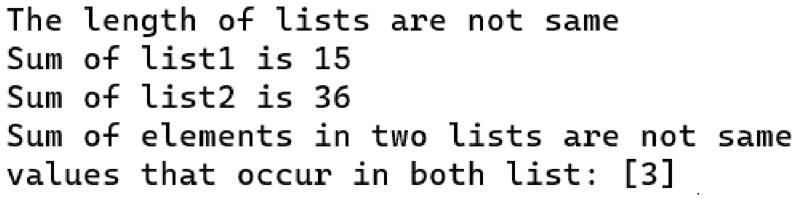
AIM

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

Input

```
#lensame l1=[1,2,3,4,5]
l2=[6,3,21,6,]
p=len(l1)
q=len(l2)
if(p==q):
    print("The length of two lists are same")
else:
    print("The length of lists are not same")
s=0
p=0
for i in l1:
    s=s+i
print("Sum of list1 is",s)
for r in l2:
    p=p+r
print("Sum of list2 is",p)
  if(s==p):
    print("Sum of elements in two lists are same")
else:
    print("Sum of elements in two lists are not same")
l3=[]
f=0
for i in l1:
    if i in l2:
        l3.append(i)
         f=f+1
print(l3)
if(f==0):
    print("no element is same")
```

Output



```
The length of lists are not same
Sum of list1 is 15
Sum of list2 is 36
Sum of elements in two lists are not same
values that occur in both list: [3]
```

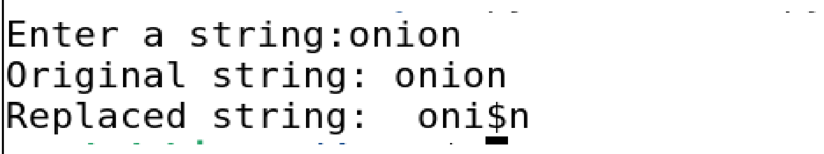
AIM

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

Input

```
#character replace

str1=input("Enter a string:")

print("Original string:",str1)

char=str1[0]

str1=str1.replace(char,'$')

str1=char+str1[1: ]

print("Replaced string: ",str1)
```

Output



```
Enter a string:onion
Original string: onion
Replaced string:  oni$n
```

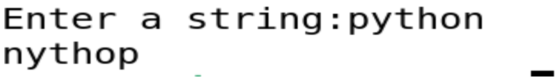
AIM

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

Input

```
#First and last character exchange
str=input("Enter a string:")
char=str[0]
char1=str[-1]
n=len(str)
ns=char1+str[1:n-1]+char print(ns)
```

Output



```
Enter a string:python
nythop
```

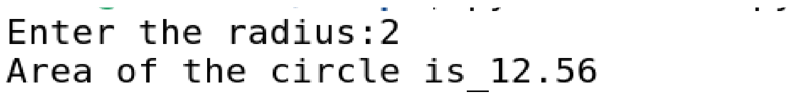
AIM

Accept the radius from user and find area of circle.

Input

```
#Area of the circle

x=int(input('Enter the radius:'))

A=3.14*x*x

print("Area of the circle is",A)
```

Output



```
Enter the radius:2
Area of the circle is 12.56
```

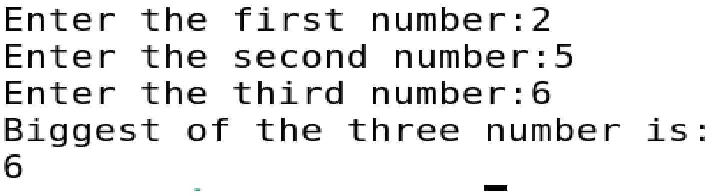
AIM

Find biggest of 3 numbers entered.

Input

```
#biggest of 3 nos
a=int(input('Enter the first number:'))
b=int(input('Enter the second number:'))
c=int(input('Enter the third number:'))
if a>b:
        if a>c:
                print(a)
        else:
                if(b>c):
                        print(b)
else:
        print(c)
```

Output



```
Enter the first number:2
Enter the second number:5
Enter the third number:6
Biggest of the three number is:
6
```

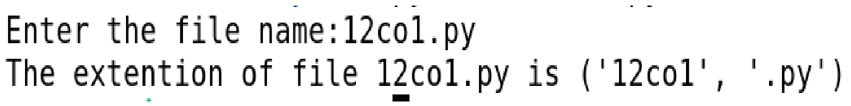
AIM

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

Input

```
#extention of file
import os
a=input("Enter the file name\n")
print(os.path.splitext(a))
```

Output



```
Enter the file name:12co1.py
The extention of file 12co1.py is ('12co1', '.py')
```

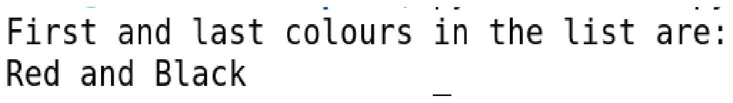
AIM

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

Input

```
#first and last colours
l1=['Red','blue','white','yellow','Black']
print("First and last colours in the list are:")
print(l1[0],'and',l1[-1])
```

Output



```
First and last colours in the list are:
Red and Black
```

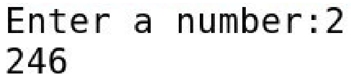
AIM

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

Input

```
#computing n+nn+nnn
x=int(input("enter the numbers"))
a=str(x)
b=a+a
c=a+a+a
d=x+int(b)+int(c)
print(d)
```

Output



```
Enter a number:2
246
```

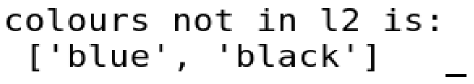
AIM

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

Input

```
#colours not in list2
l1=['red','blue','black']
l2=['red','white','pink']
l3=[]
for i in l1:
        if i not in l2:
                l3.append(i)
print('colours not in l2 is:\n',l3)
```

Output



```
colours not in l2 is:
 ['blue', 'black']   _
```

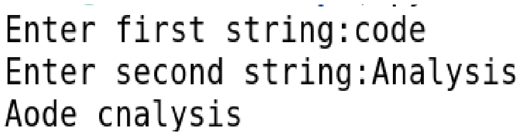
AIM

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

Input

```
#swapping
str1=input("Enter first string:")
str2=input("Enter second string:")
str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]
print(str3)
```

Output



```
Enter first string:code
Enter second string:Analysis
Aode cnalysis
```

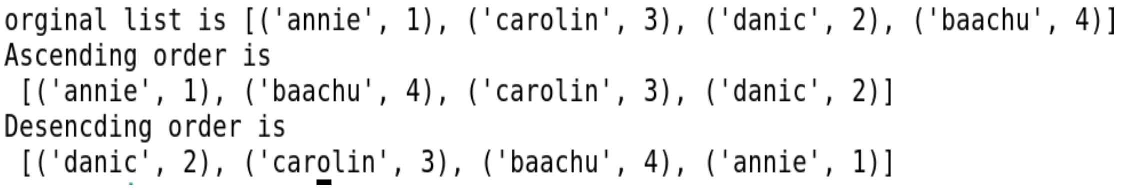
AIM

Sort dictionary in ascending and descending order.

Input

```
#ascending and descending order
d1={"annie":1,"carolin":3,"danic":2,"baachu":4}
l=list(d1.items())

print("orginal list is",l)
l.sort()
print("Ascending order is\n",l)
l=list(d1.items())
l.sort(reverse=True)
print("Desencding order is\n",l)
```

Output



```
orginal list is [('annie', 1), ('carolin', 3), ('danic', 2), ('baachu', 4)]
Ascending order is
 [('annie', 1), ('baachu', 4), ('carolin', 3), ('danic', 2)]
Desencding order is
 [('danic', 2), ('carolin', 3), ('baachu', 4), ('annie', 1)]
```

Course Outcome 2(CO2):

AIM

Program to find the factorial of a number.

Input

```
#co21factorial of a number
n=int(input('Enter the number:'))
fact=1
for i in range(1,n+1):
     fact=fact*i
print(fact)
```

Output

```
Enter the number:4
24
```

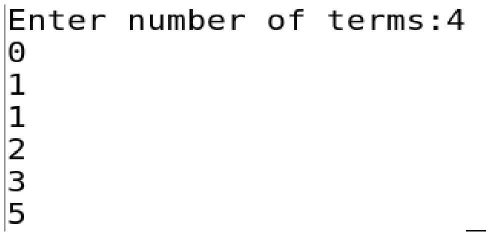
AIM

Generate Fibonacci series of N terms.

Input

```
#co22fibanocciseries
n=int(input('Enter number of terms:'))
f1=0
f2=1
print(f1,f2)
for i in range(0,n):
    f3=f1+f2
    print(f3)
    f1=f2
    f2=f3
```

Output



```
Enter number of terms:4
0
1
1
2
3
5
```

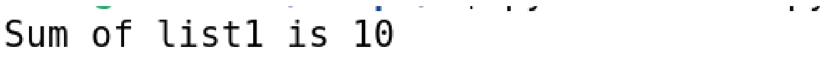
AIM

Find the sum of all items in a list.

Input

```
#sum of items in list
l1=[1,2,3,4]
l3=[]
s=0
p=0
for i in l1:
        s=s+i
print("Sum of list1 is",s)
```

Output



```
Sum of list1 is 10
```

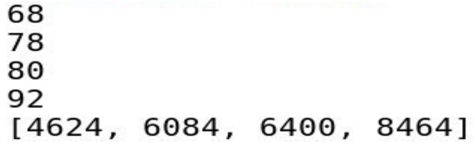
AIM

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

Input

```
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



```
68
78
80
92
[4624, 6084, 6400, 8464]
```

AIM

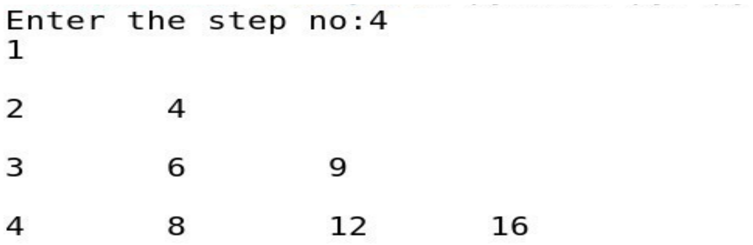
Display the given pyramid with step number accepted from the user.

```
Eg: N=4 1
2 4
3 6 9
4 8 12 16
```

Input

```
#pyramid with step no
n=int(input('enter the step number'))
for i in range(1,n+1):
        for j in range(1,i+1):
                s=i*j
                print(s,'\t',end="")
        print("\n")
```

Output



```
Enter the step no:4
1

2       4

3       6       9

4       8       12      16
```

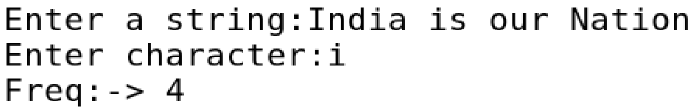
AIM

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

Input

```
#character frequency
str=input("Enter a string:")
fnd=input("Enter character:")
cnt=0
fnd=fnd.lower()
str=str.lower()
for i in str:
    if i==fnd:
        cnt=cnt+1
print("Freq:→",cnt)
```

Output



```
Enter a string:India is our Nation
Enter character:i
Freq:-> 4
```

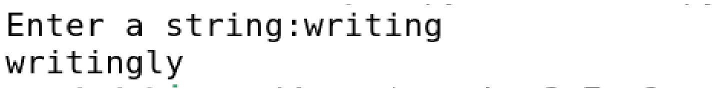
AIM

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

Input

```
#add ing
a=input("Enter a word\n")
l=len(a)
ll=a[l-3:l]
if(ll=="ing"):
        s=a+"ly"
else:
        s=a+"ing"
print (s)
```

Output



```
Enter a string:writing
writingly
```

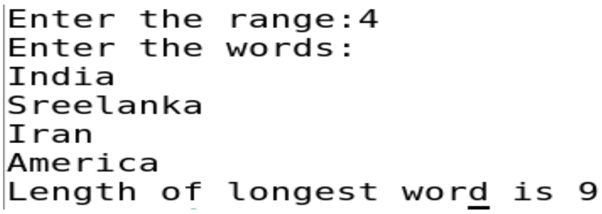
AIM

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

Input

```
#Length of the longest
word

          lis=[]
          n=int(input("Enter
          the range:"))
          print("Enter the
          words:")

          for i in
          range(0,
          n):

          lis.appe
          nd(input
          (""))
          longest
          =lis[0]

          for i in
            range(1,n):

            if(len(lis[i])
            >len(longe
            st)):

            longest=lis
            [i]

          print("Length of longest word is",len(longest))
```

Output



```
Enter the range:4
Enter the words:
India
Sreelanka
Iran
America
Length of longest word is 9
```

AIM

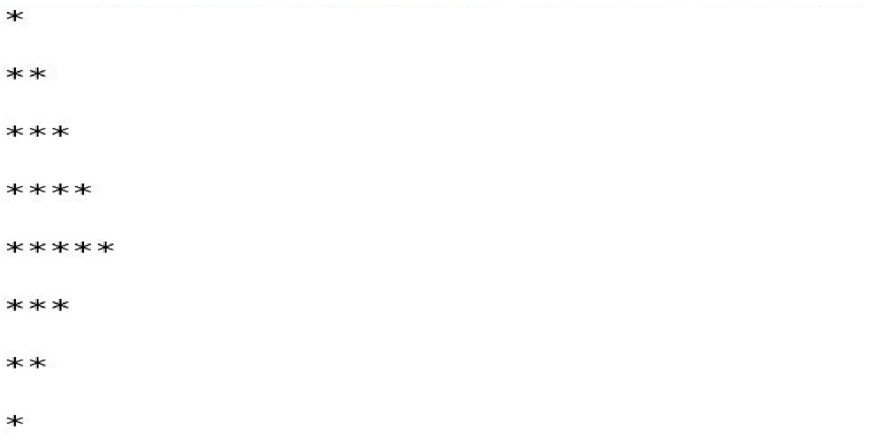
Construct following pattern using nested loop.

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

Input

```
#pattern star
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(i,0,-1):
                print("*",end="")
        print("\n")
```

Output



```
*

**

***

****

*****

***

**

*
```

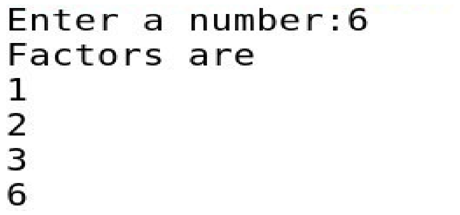
AIM

Generate all factors of a number.

Input

```
#All factors of a no
n=int(input("Enter a number:"))
print("Factors are")

for i in         range(1,n+1):
        if(n%i==0):
                print(i)
```

Output



```
Enter a number:6
Factors are
1
2
3
6
```

Course Outcome 3(CO3):

AIM

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.

Input

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
```

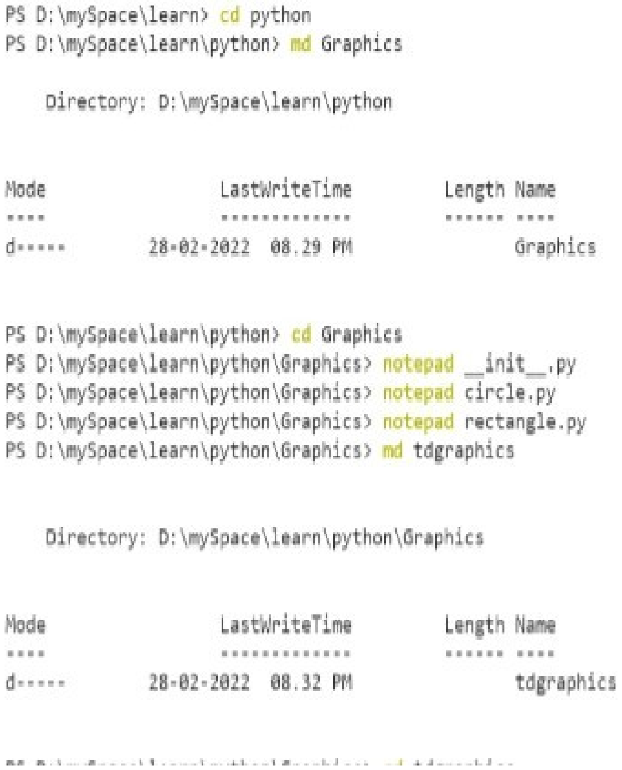
Graphics\tdgraphics\sphere.py

```
from math import pi

def area_sphere(radius):
        return 4*(pi*radius*radius)

def perimeter_sphere(radius):
        return 2*pi*radius
```

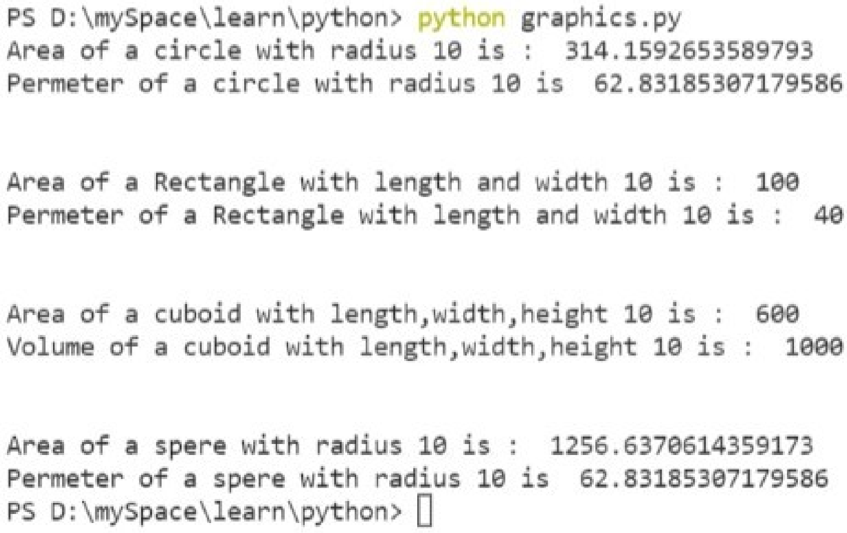
Graphics.py (driver code)

```
import graphics
from graphics import circle,rectangle
from graphics.tdgraphics import cuboid,sphere
from graphics.circle import *

print("Area of a circle with radius 20 is : ",circle.area_circle(20))
print("Permeter of a circle with radius 20 is ",circle.perimeter_circle(20))
print("\n")

print("Area of a Rectangle with length 20 and width 10 is : ",rectangle.area_rec(20,10))
print("Permeter of a Rectangle with length 20 and width 10 is : ",rectangle.perimeter_rec(20,10))
print("\n")

print("Area of a cuboid with length,width,height 8 is : ",cuboid.area_cuboid(8,8,8))
print("Volume of a cuboid with length,width,height 12 is : ",cuboid.volume_cuboid(12,12,12))
print("\n")

print("Area of a spere with radius 20 is : ",sphere.area_sphere(20))
print("Permeter of a spere with radius 20 is ",sphere.perimeter_sphere(20))
```

Output



```
PS D:\mySpace\learn> cd python
PS D:\mySpace\learn\python> md Graphics

    Directory: D:\mySpace\learn\python


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        28-02-2022  08.29 PM                Graphics


PS D:\mySpace\learn\python> cd Graphics
PS D:\mySpace\learn\python\Graphics> notepad __init__.py
PS D:\mySpace\learn\python\Graphics> notepad circle.py
PS D:\mySpace\learn\python\Graphics> notepad rectangle.py
PS D:\mySpace\learn\python\Graphics> md tdgraphics


    Directory: D:\mySpace\learn\python\Graphics


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        28-02-2022  08.32 PM                tdgraphics
```



```
PS D:\mySpace\learn\python> python graphics.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
PS D:\mySpace\learn\python> 
```

Course Outcome 4(CO4):

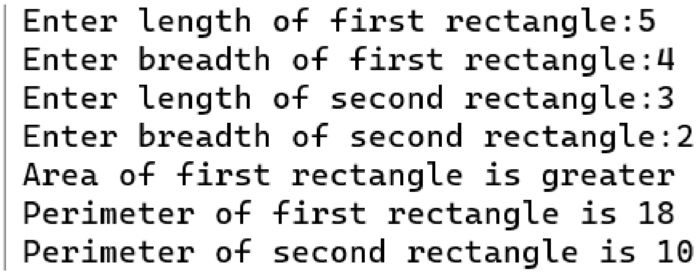
AIM

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

Input

```
class Rectangle:
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def area(self):
        return (self.l*self.b)
    def perimeter(self):
        return 2*(self.l+self.b)
p=int(input("Enter length of first rectangle:"))
q=int(input("Enter breadth of first rectangle:"))
r=int(input("Enter length of second rectangle:"))
s=int(input("Enter breadth of second rectangle:"))
r1=Rectangle(p,q)
r2=Rectangle(r,s)
x=r1.area()
y=r2.area()
z=r1.perimeter()
h=r2.perimeter()
if(x>y):
    print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
print("Perimeter of first rectangle is",z)
print("Perimeter of second rectangle is",h)
```

Output



```
Enter length of first rectangle:5
Enter breadth of first rectangle:4
Enter length of second rectangle:3
Enter breadth of second rectangle:2
Area of first rectangle is greater
Perimeter of first rectangle is 18
Perimeter of second rectangle is 10
```

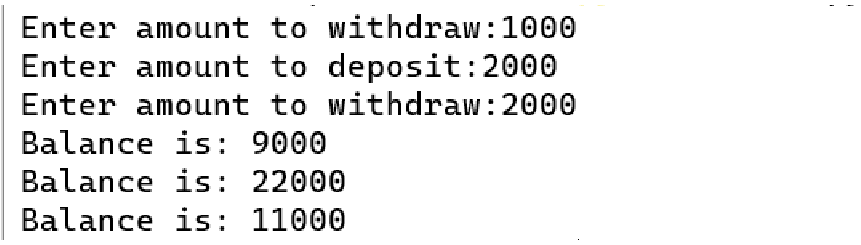
AIM

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.

Input

```
class Bank:
        def __init__(self,acno,name,typeofac,balance):
                self.acno=acno
                self.name=name
                self.typeofac=typeofac
                self.balance=balance
         def withdraw(self,x):
                self.balance=self.balance-x
                print("Balance is:",self.balance)
        def deposit(self,y):
                self.balance=self.balance+y
                print("Balance is:",self.balance)


ac1=Bank(1,"Aiswarya","SB",10000)
ac2=Bank(2,"Krishnenthu","SB",20000)
p=int(input("Enter amount to withdraw:"))
q=int(input("Enter amount to deposit:"))
r=int(input("Enter amount to withdraw:"))
ac1.withdraw(p)
ac2.deposit(q)
ac1.deposit(r)
```

Output



```
Enter amount to withdraw:1000
Enter amount to deposit:2000
Enter amount to withdraw:2000
Balance is: 9000
Balance is: 22000
Balance is: 11000
```

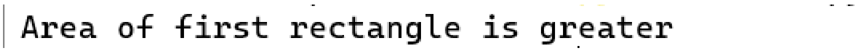
AIM

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

Input

```
class Rectangle:
    def __init__(self,length,breadth):
        self.__length=length
        self.__breadth=breadth
    def area(self):
        a=self.__length*self.__breadth
        print("area",a)
        return a
    def perimeter(self):
        p=2*(self.__length+self.__breadth)
        print("perimeter",p)
    def __lt__(self,rr):

    if(self.__breadth*self.__length>rr.__breadth*rr.__length):
            return True
        else:
            return False
r1=Rectangle(5,7)
r2=Rectangle(4,6)
if(r1<r2):
    print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
```

Output



```
Area of first rectangle is greater
```

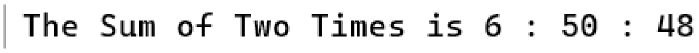
AIM

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

Input

```
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("The Sum of Two Times is",hr,":",min,":",sec)
t1=Time(2,30,46)
t2=Time(4,20,2)
t1+t2
```

Output



```
The Sum of Two Times is 6 : 50 : 48
```

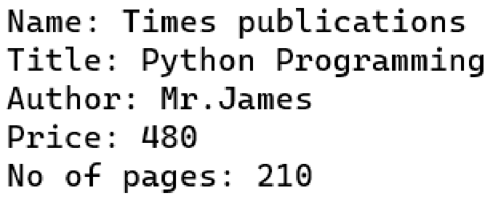
AIM

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 ages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Input

```
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,no_of_pages):
        super().__init__(name,title,author)
        self.price=price
        self.no_of_pages=no_of_pages
    def display(self):
        print("Name:",self.name)
        print("Title:",self.title)
        print("Author:",self.author)
        print("Price:",self.price)
        print("No of pages:",self.no_of_pages)
p1=Python("Times publications","Python Programming","Mr.James",480,210)
p1.display()
```

Output



```
Name: Times publications
Title: Python Programming
Author: Mr.James
Price: 480
No of pages: 210
```

Course Outcome 5(CO5):

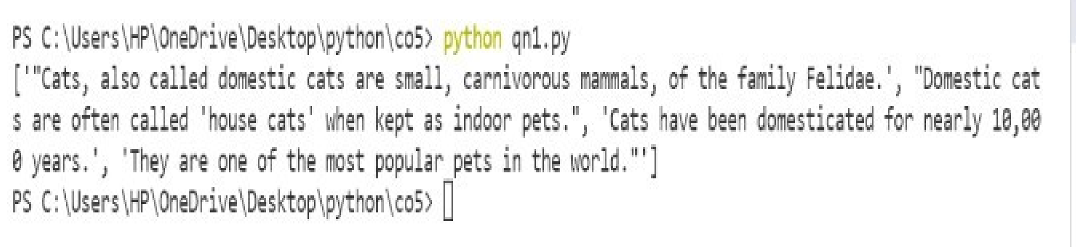
AIM

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

Input

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

Output



```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn1.py
['"Cats, also called domestic cats are small, carnivorous mammals, of the family Felidae.', "Domestic cat
s are often called 'house cats' when kept as indoor pets.", 'Cats have been domesticated for nearly 10,00
0 years.', 'They are one of the most popular pets in the world."']
PS C:\Users\HP\OneDrive\Desktop\python\co5> 
```

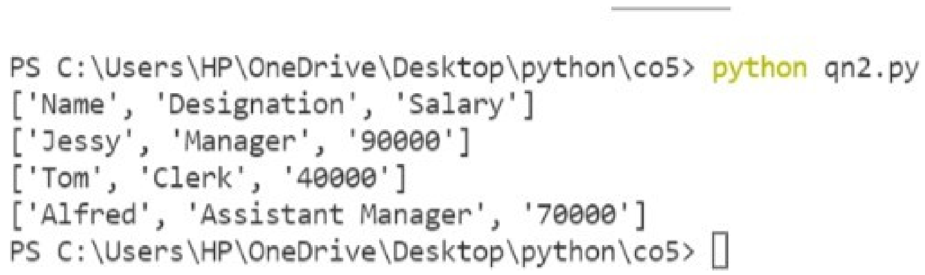
AIM

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

Input

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

Output



```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn2.py
['Name', 'Designation', 'Salary']
['Jessy', 'Manager', '90000']
['Tom', 'Clerk', '40000']
['Alfred', 'Assistant Manager', '70000']
PS C:\Users\HP\OneDrive\Desktop\python\co5> []
```