FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

PYTHON PROGRAMMING LAB

……..

LABORATORY RECORD

Name: ARAVIND M.A.

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: SEMESTER -1 A Roll No: 36

FEDERAL INSTITUTE OF

SCIENCE AND TECHNOLOGY

(FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name :ARAVIND M.A.

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 **Roll No: 36**

University Exam.Reg. No: FIT21MCA-2036

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 and submitted to Kerala Technological University *in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by* ***ARAVIND M. A.*** *in the* ***PYTHON*** *Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.*

Signature of Staff in Charge

Name: JOICE T

Date:

Signature of H.O.D

Name: DEEPA MARY MATHEWS

Date of University practical examination ……

Signature of

Internal Examiner

Signature of

External Examiner

CONTENT

SI No:	Date :	Name of Experiment:	Page No:	Signature of Staff –In – Charge:
1		Display future leap years from current year to a final year entered by user.		
2		Generate positive list of numbers from a given list of integers.		
3		Print square of N numbers.		
4		Form a list of vowels selected from a given word.		
5		List ordinal value of each element of a word (Hint: use ord() to get ordinal values)		
6		Count the occurrences of each word in a line of text.		
7		Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.		
8		Store a list of first names. Count the occurrences of ‘a’ within the list		
9		Get a string from an input string where all occurrences of first character replaced with “$’, except first character.		
10		Create a string from given string where first and last characters exchanged.		
11		Accept the radius from user and find area of circle.		
12		Find biggest of 3 numbers entered.		
13		Accept a file name from user and print extension of that.		

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

29		Construct following pattern using nested loop.		
30		Generate all factors of a number.		
31		Create a package graphics with modules rectangle, circle and sub-package 3D graphics with module cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements.		
32		Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare to rectangle objects by their area.		
33		Create 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.		
34		Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of two rectangles.		
35		Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of two time.		
36		Create a class Publisher(name). Derive class Book from Publisher with attributes title and author. Derive class python from Book with attributes price and number_of_pages. Write a program that displays information		

		about a Python book. Use base class constructor invocation and method overreading.		
37		Write a program to read a file line by line and store it into a list		
38		Write a Python program to read each row from a given csv file and print a list of strings.		

COURSE OUTCOME-1

EXPERIMENT 1

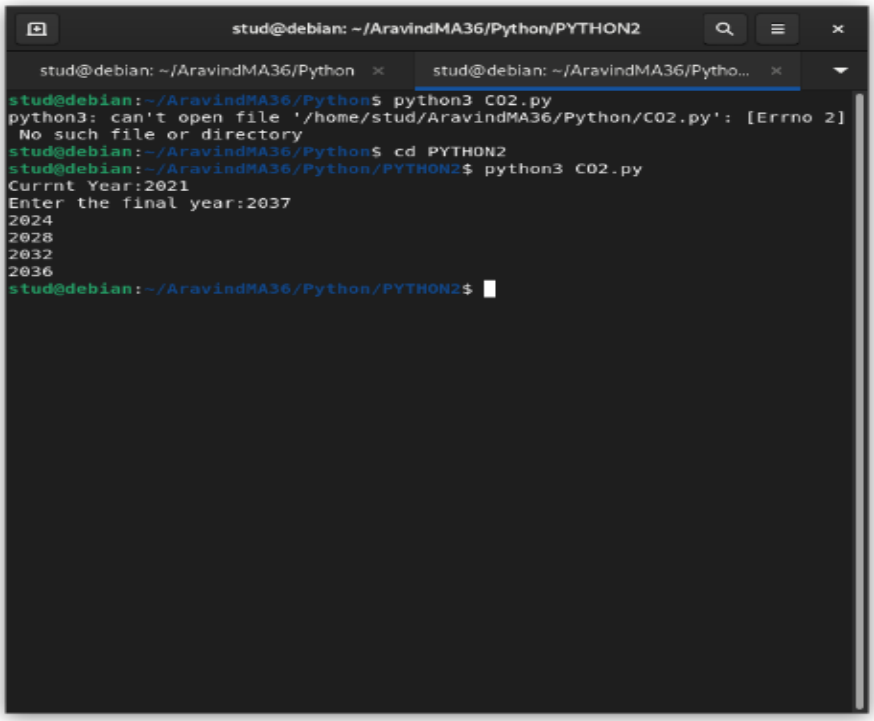
AIM: Display future leap years from current year to a final year entered by user.

CODE:

```
print("Current Year:2021")
p=int(input("Enter the final year:"))
if p<2021:
        print("Date should be greater than 2021!!")

for p in range(2021, p+1):
        if p%400==0 or p%4==0 and p%100!=0:
                print(p)
```

OUTPUT:



EXPERIMENT 2

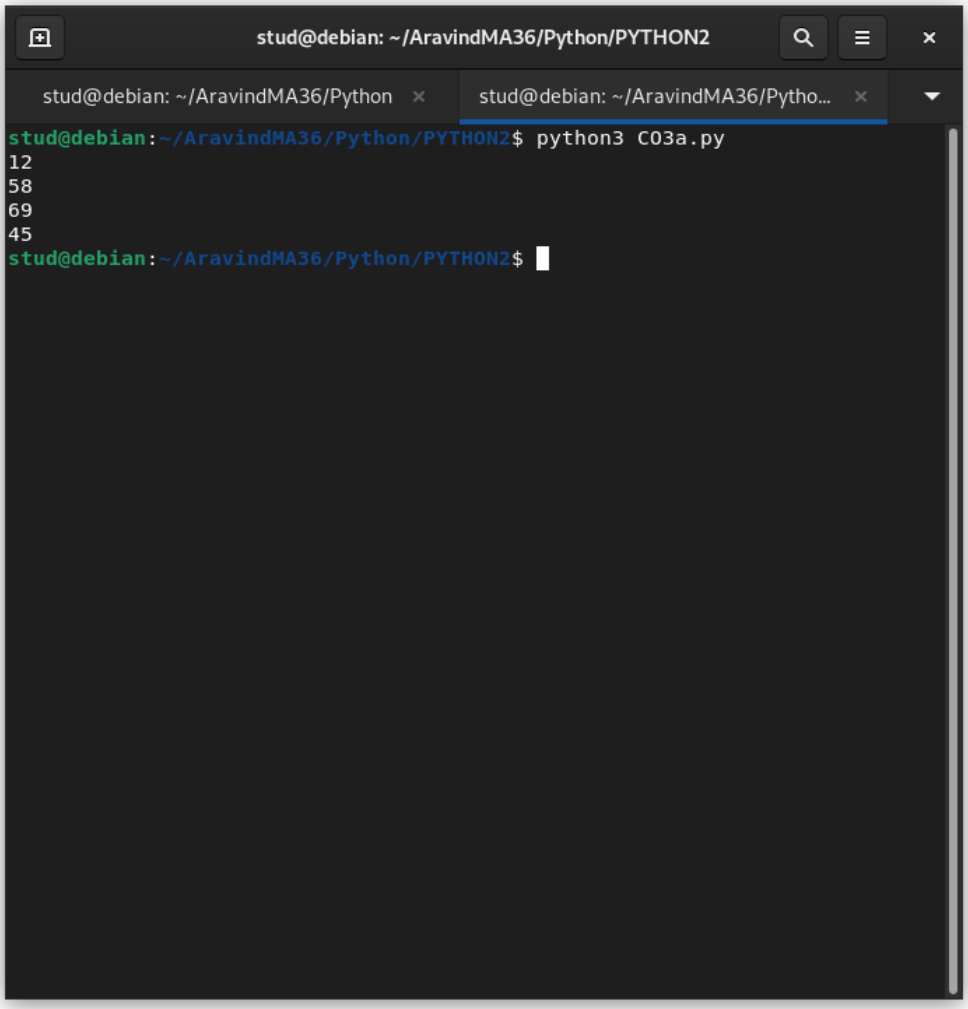
AIM: Generate positive list of numbers from a given list of integers.

CODE:

```
list=[12,-89,-98,58,69,-1,45]

for i in range(0,7):
        if list[i]>0:
                print(list[i])
```

OUTPUT



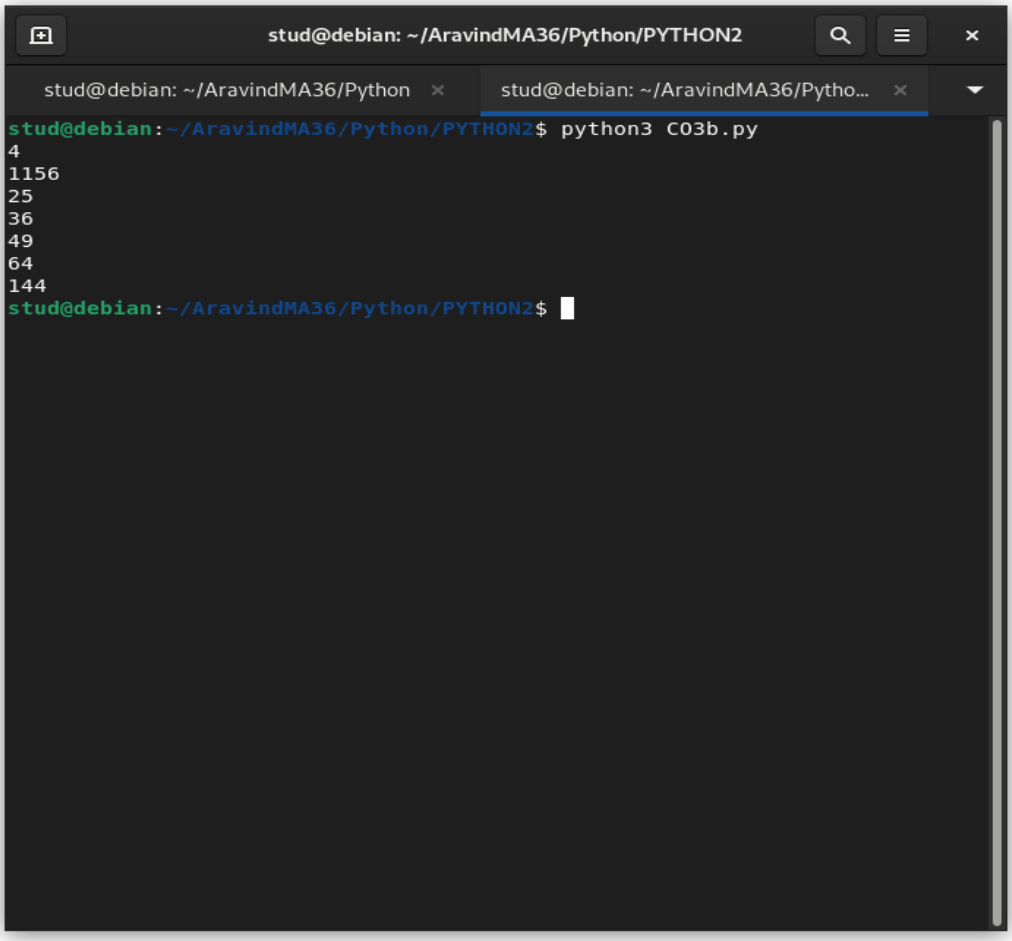
EXPERIMENT 3

AIM: Print square of N numbers.

CODE:

```
list=[2,34,5,6,7,8,12]
for i in range(0,7):
  print(list[i]*list[i])
```

OUTPUT:



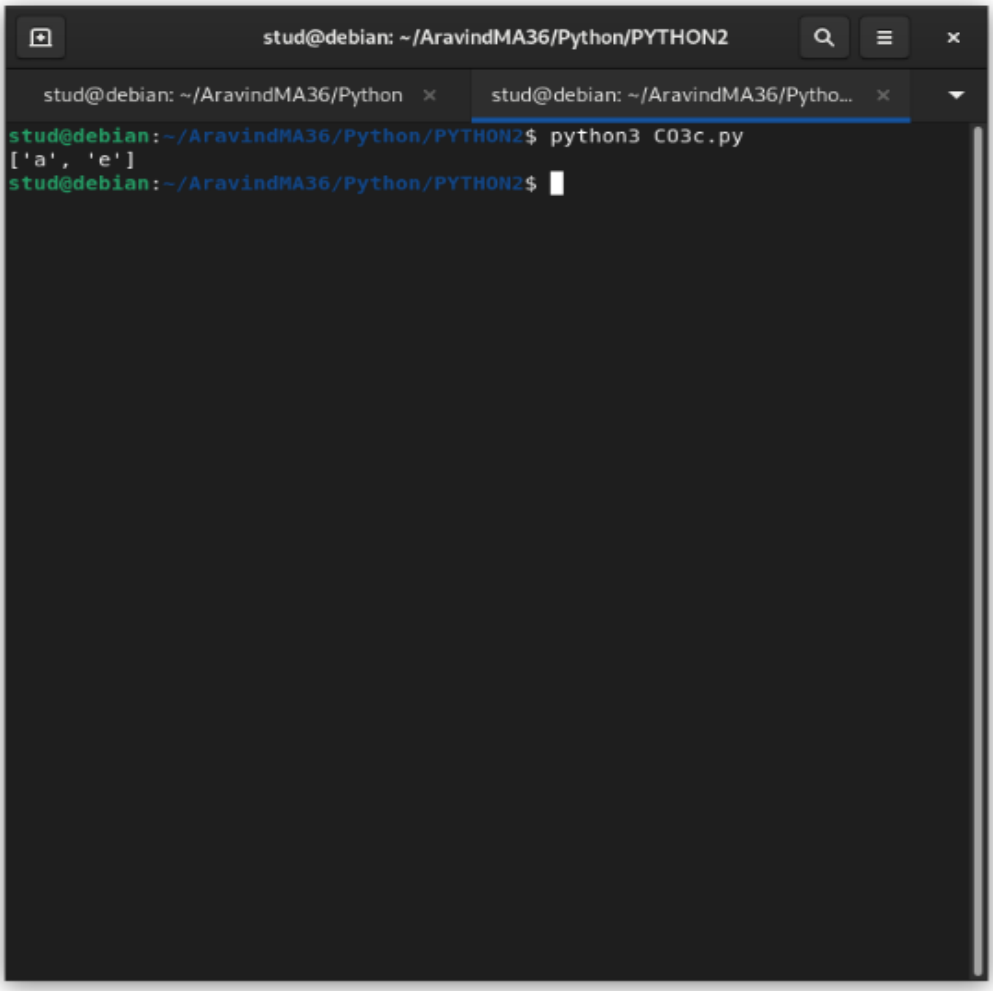
EXPERIMENT 4

AIM: Form a list of vowels selected from a given word.

CODE:

```
lst=[]
str='apple'
p=len(str)
for i in range(0,p):
        if str[i] in ("aeiouAEIOU"):
                lst.append(str[i])
print(lst)
```

OUTPUT:



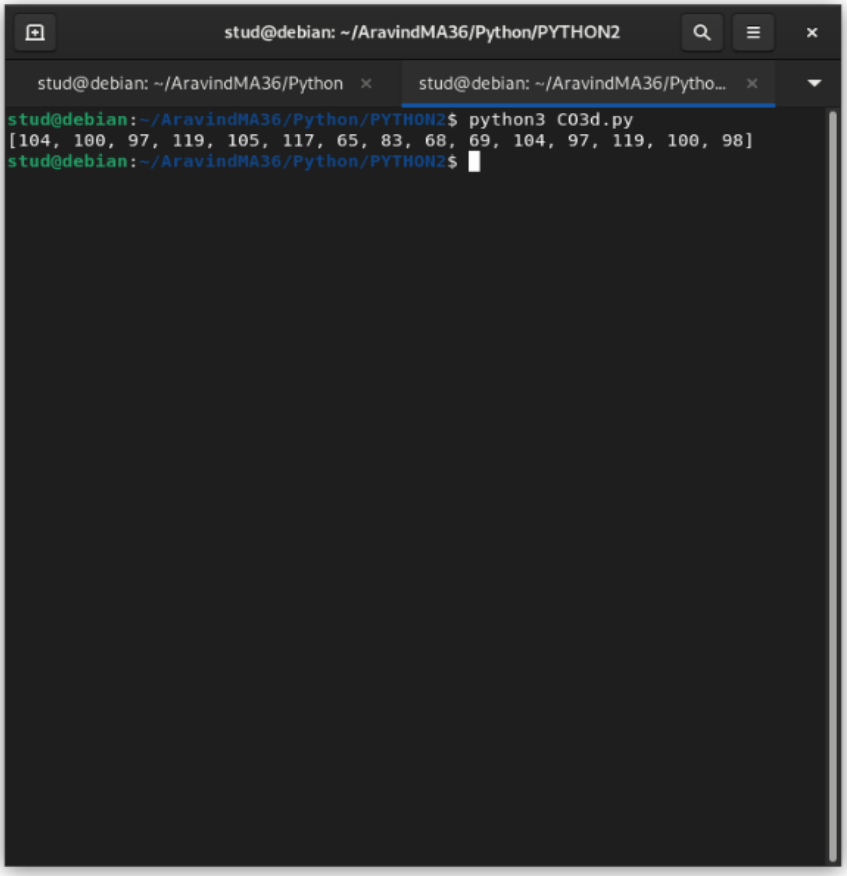
EXPERIMENT 5

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

CODE:

```
str='hdawiuASDEhawdb'
r=[]
p=len(str)
for i in range(0,p):
        r.append(ord(str[i]))
print(r)
```

OUTPUT:



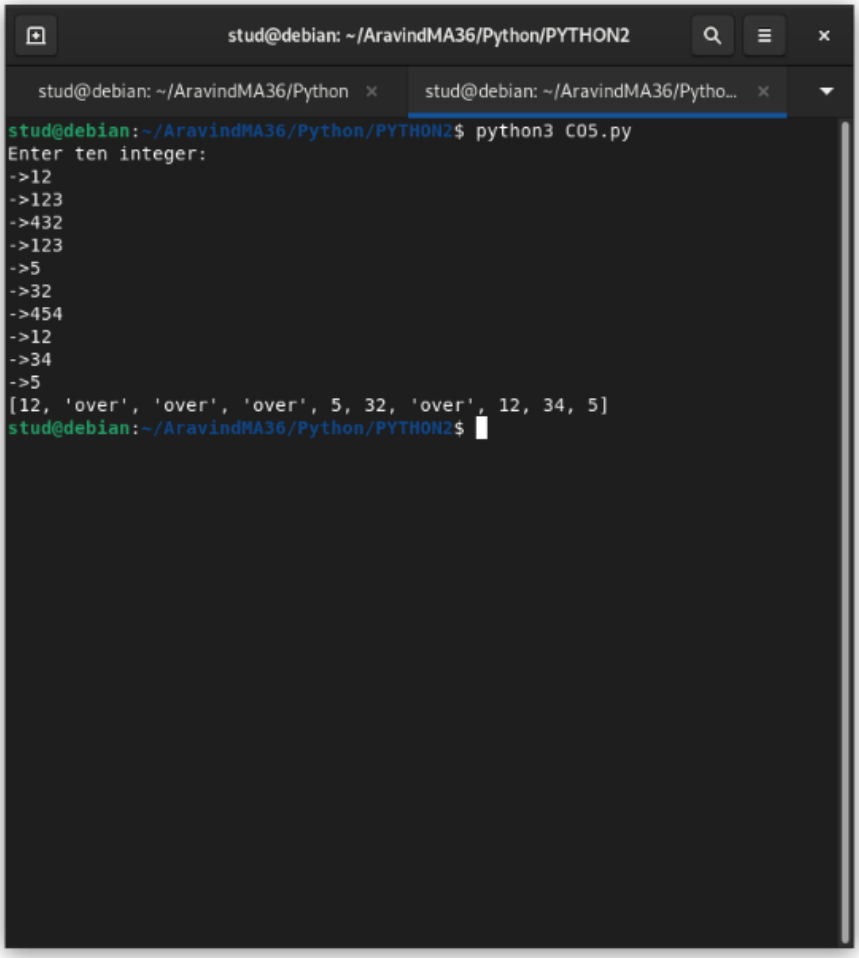
EXPERIMENT 6

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

CODE:

```
arr=[]
print("Enter ten integer:")
for i in range(0,10):
        k=int(input("->"))
                if k>100:
                        arr.append("over")
                else:
                        arr.append(k)
print(arr)
```

OUTPUT:



EXPERIMENT 7

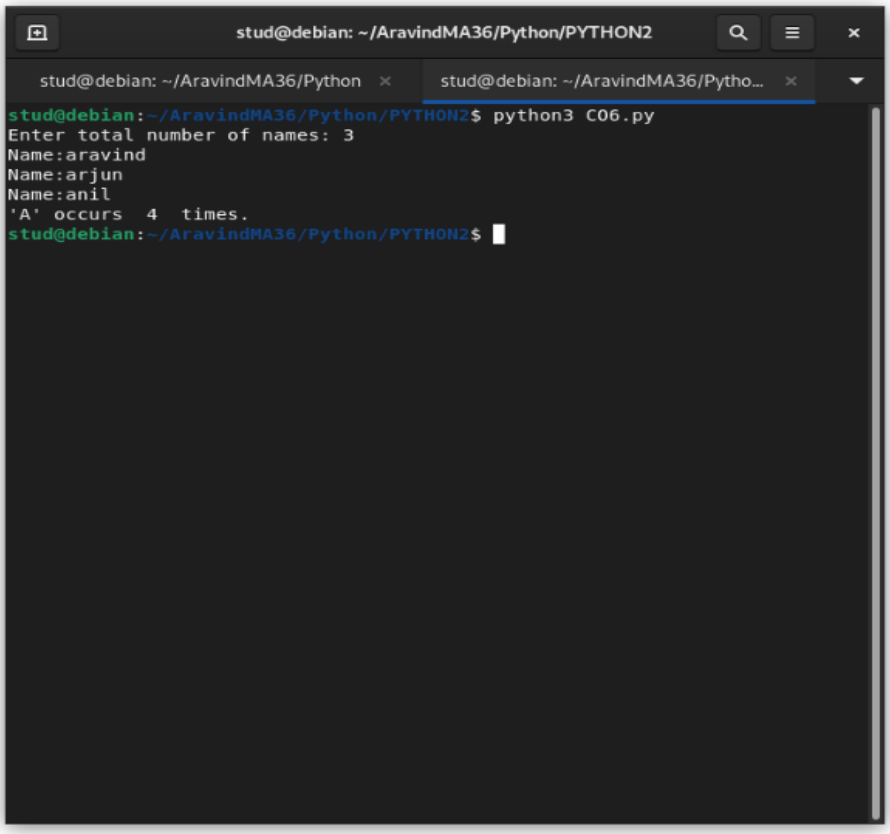
AIM: Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

CODE:

```
nme=[]
count=0
lim=int(input("Enter total number of names: "))
for i in range(0, lim):
        val=input("Name:")
        nme.append(val)

for i in range(0,lim):
        for y in nme[i]:
                if y=='a' or y=='A':
                count+=1
        print("'A' occurs ",count," times.")
```

OUTPUT:



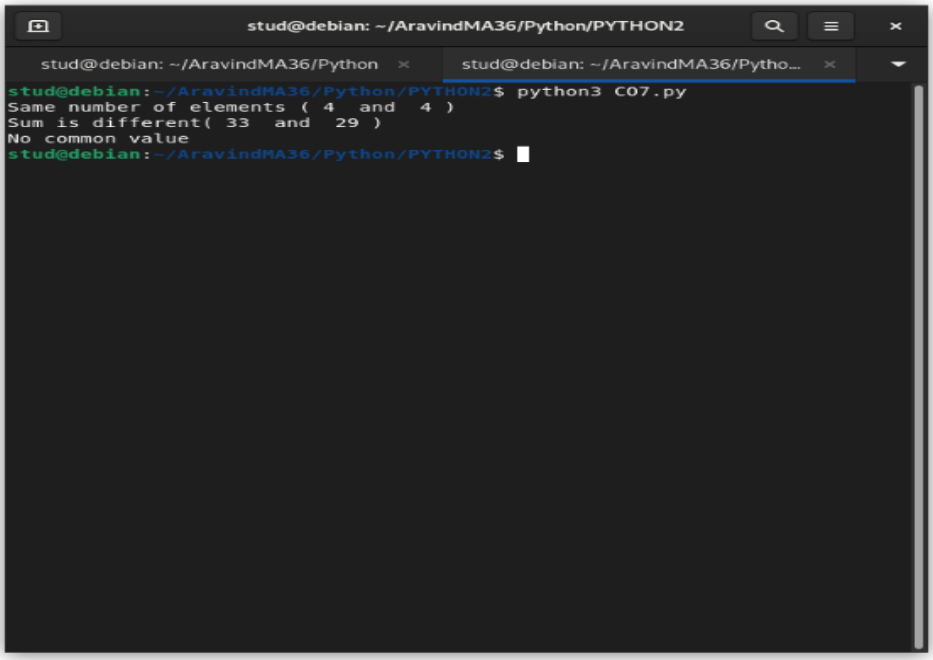
EXPERIMENT 8

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

CODE:

```
p1=[1,13,15,4]
p2=[12,5,5,7]
j=0
if len(p1)==len(p2):
        print("Same number of elements (",len(p1), " and ", len(p2),")")
else:
        print("Diffeernt number of elements")
sum1=sum(p1)
sum2=sum(p2)
if sum1==sum2:
        print("Sum is same")
else:
        print("Sum is different(",sum1," and ",sum2,")")
for i in p1:
        if i in p2:
                print("Same number-> ", i)
                j=j+1;
        if j==0:
                print("No common value")
```

OUTPUT:



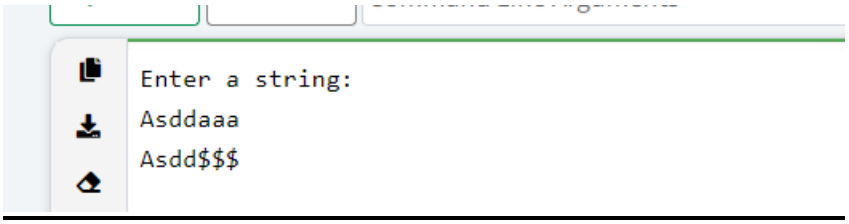
EXPERIMENT 9

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

CODE:

```
ch=input("Enter a string:")
f=ch[0]
print(ch[0],end="")
f=f.lower()
for i in range(1,len(ch)):
        if ch[i]==f:
                print("$", end="")
        else:
                print(ch[i],end="")
```

OUTPUT:



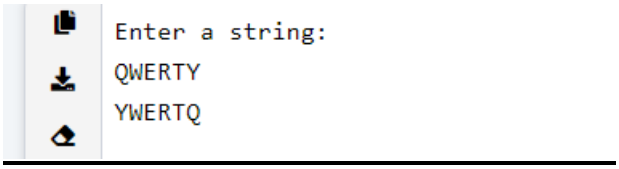
EXPERIMENT 10

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

CODE:

```
var=input("Enter a string: ")
beg=var[0]
end=var[len(var)-1]
dum=beg
beg=end
end=dum
print(beg+var[1:len(var)-1]+end)
```

OUTPUT:



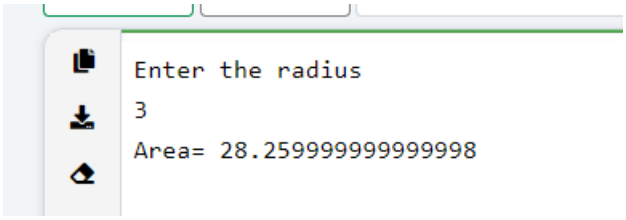
EXPERIMENT 11

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

CODE:

```
p=int (input("Enter the radius"))
ar=3.14*p*p
print("Area=",ar)
```

OUTPUT:



EXPERIMENT 12

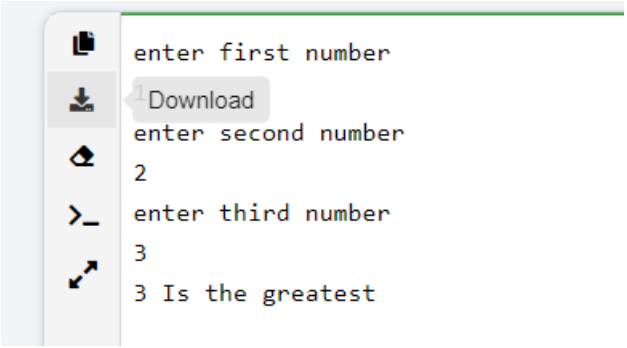
AIM: Find biggest of 3 numbers entered**.**

CODE:

```
p= int(input("enter first number "))
q= int(input("enter second number "))
r= int(input("enter third number "))

if p>q:
            if p>r:
             print(p,"Is the greatest")
            else:
             print(r,"Is the greatest")
else :
      if q>r:
        print(q, "Is the greatest")
      else:
        print(r,"Is the greatest")
```

OUTPUT:



EXPERIMENT 13

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

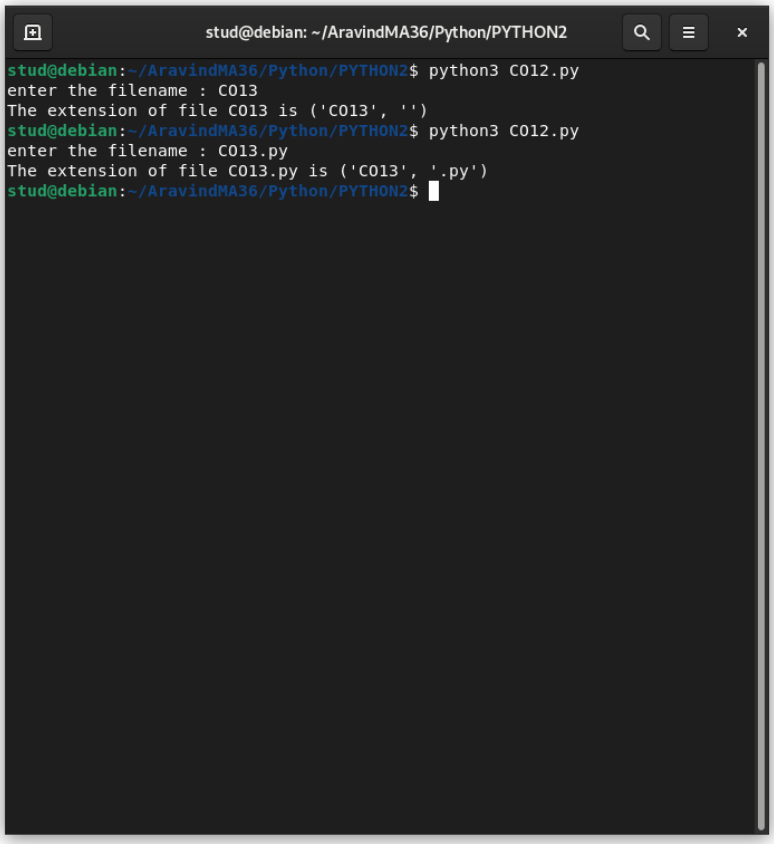
CODE:

```
import os

a=input("enter the filename : ")

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

OUTPUT:



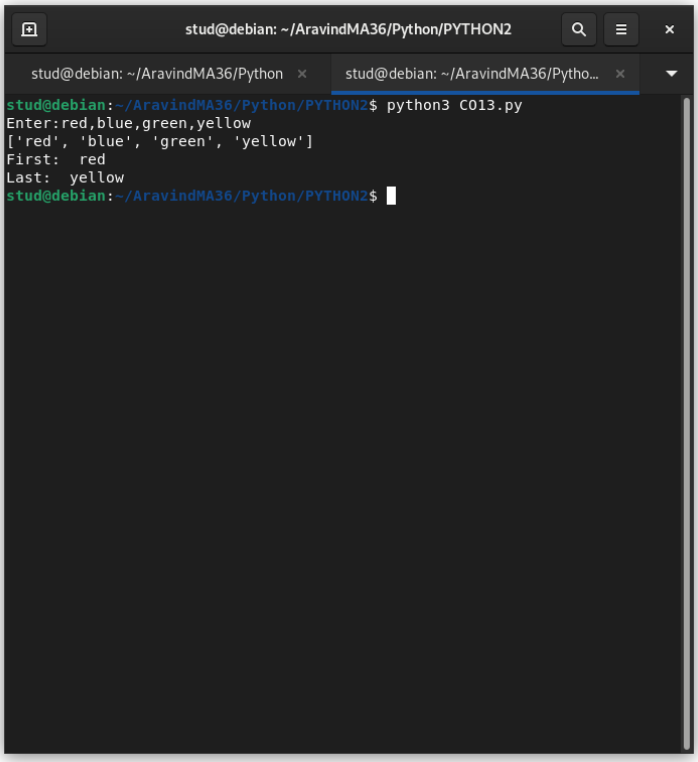
EXPERIMENT 14

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

CODE:

```
color=[]
color=[i for i in input("Enter:").split(',')]
print(color)
for i in color:
 m1=color[0]
 m2=color[len(color)-1]
print("First: ",m1,"\nLast: ",m2)
```

OUTPUT:



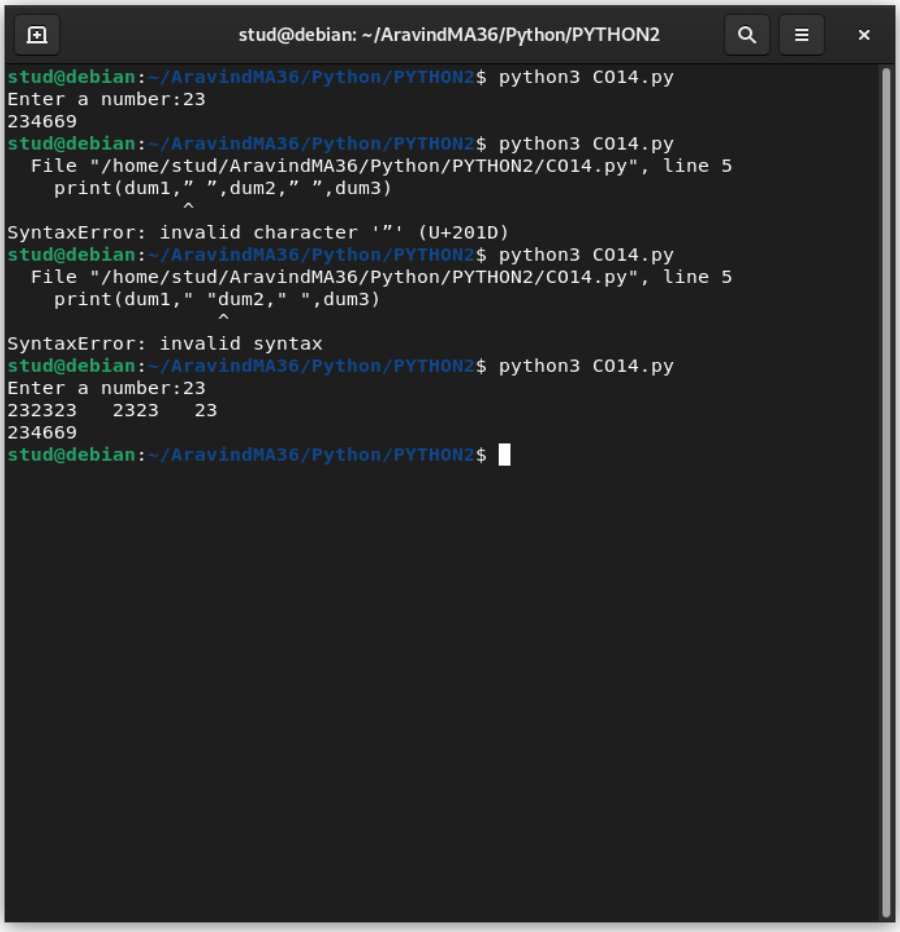
EXPERIMENT 15

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

CODE:

```
num=input("Enter a number:")
dum1=num+num+num
dum2=num+num
dum3=num
print(dum1,” ”,dum2,” ”,dum3)
print(int(dum1)+int(dum2)+int(dum3))
```

OUTPUT:



EXPERIMENT 16

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

CODE:

```
clr1=['red','blue','yellow','pink','green','grey']
clr2=['blue','brown','violet','pink','orange','green']
for i in clr1:
 if i not in clr2:
  print(i)
```

OUTPUT:

```
red
yellow
grey
```

EXPERIMENT 17

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

CODE:

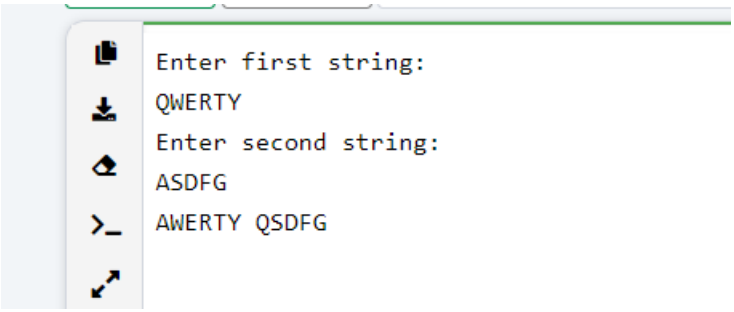
```
s1=input("Enter first string:")

s2=input("Enter second string:")

g=s2[0]+s1[1:]+" "+s1[0]+s2[1:]

print(g)
```

OUTPUT:



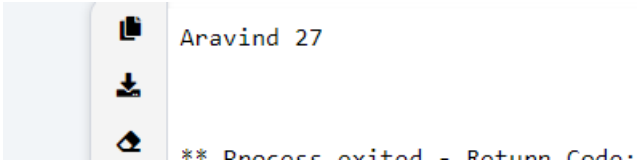
EXPERIMENT 18

AIM: Merge two dictionaries.

CODE:

```
d={"name":"Aravind", "age":"27"}
e={"sex":"Male", "Qual":"degree"}
d.update(e)
print(d["name"])
```

OUTPUT:



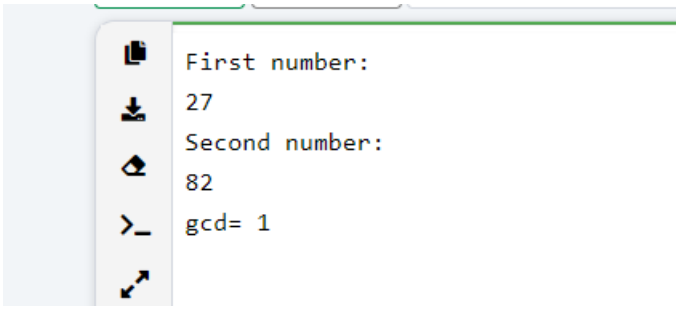
EXPERIMENT 19

AIM: Find gcd of 2 numbers.

CODE:

```
num1=int(input("First number: "))
num2=int(input("Second number: "))
if(num1<num2):
 dum=num1
 num1=num2
 num2=dum
for i in range(1,num2+1):
 if(num1%i==0) and (num2%i==0):
  gcd=i
print("gcd=",gcd)
```

OUTPUT:



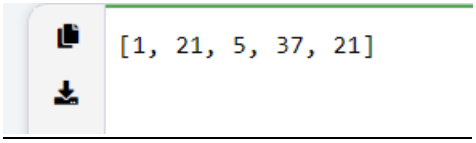
EXPERIMENT 20

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

CODE:

```
p=[1,12,21,24,4,5,37,21]
q=[]
for i in p:
 if i%2!=0:
  q.append(i)
print(q)
```

OUTPUT:



COURSE OUTCOME-2

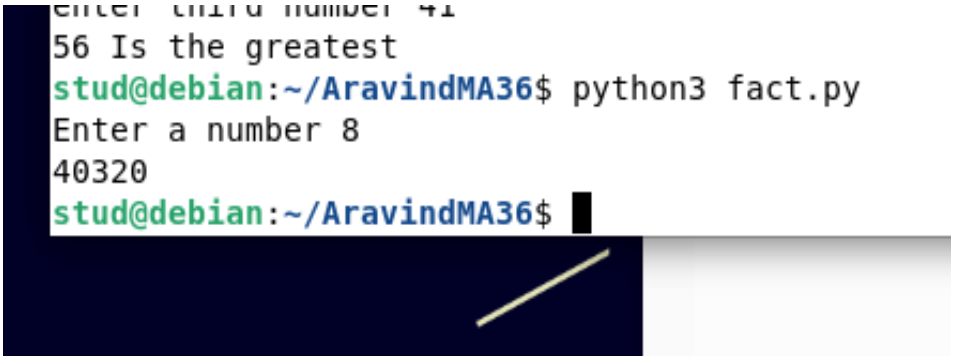
EXPERIMENT 21

AIM: Program to find the factorial of a number

CODE

```
p=int(input("Enter a number "))
fact=1
for p in range(1,p+1):
  fact=fact*p
print(fact)
```

OUTPUT



EXPERIMENT 22

AIM: Generate Fibonacci series of N terms

CODE

```
trm=int(input("Enter number of terms"))

p=0

q=1

print(p)

print(q)

for trm in range(1, trm-1):

        out=p+q

        print(out)

         p=q

        q=out
```

OUTPUT

```
enter first number 8
enter second number 56
enter third number 41
56 Is the greatest
stud@debian:~/AravindMA36$ python3 fact.py
Enter a number 8
40320
stud@debian:~/AravindMA36$ python3 fiban.py
Enter number of terms8
0
1
1
2
3
5
8
13
stud@debian:~/AravindMA36$
```

EXPERIMENT 23

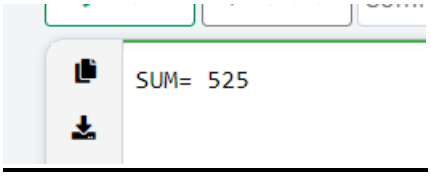
AIM: Find the sum of all items in a list

CODE

```
a=[15,58,66,-99,456,-66,95]

print(sum(a))
```

OUTPUT



EXPERIMENT 24

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

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:~/AravindMA36/Python/PYTHON2$ python3 CO2_4.py
68
78
80
92
[4624, 6084, 6400, 8464]
stud@debian:~/AravindMA36/Python/PYTHON2$
```

EXPERIMENT 25

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

Eg: N=4

1

2 4

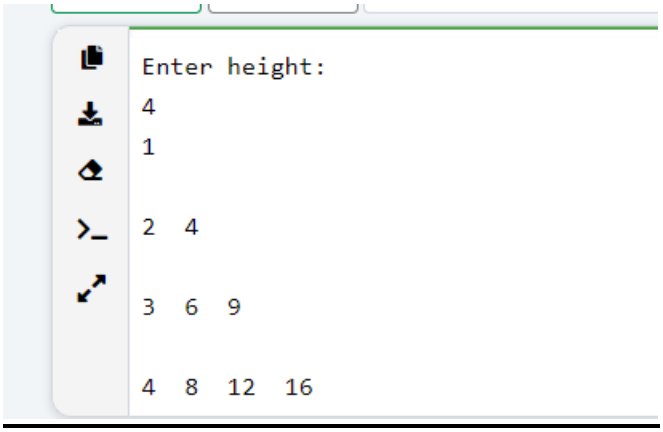
3 6 9

4 8 12 16

CODE:

```
ht=int(input("Enter height:"))
for i in range(1,ht+1):
 for j in range(1,i+1):
  print((j*i)," ",end="")
 print("\n")
```

OUTPUT



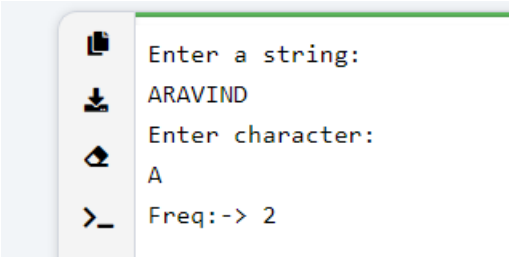
EXPERIMENT 26

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

CODE

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

OUTPUT



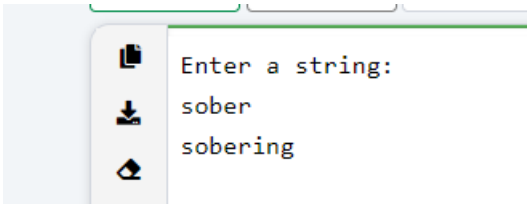
EXPERIMENT 27

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

CODE

```
a=input("Enter a string:")
l=len(a)
if a[l-3]=='i' and a[l-2]=='n' and a[l-1]=='g':
 print(a+'ly')
else:
 print(a+'ing')
```

OUTPUT



EXPERIMENT 28

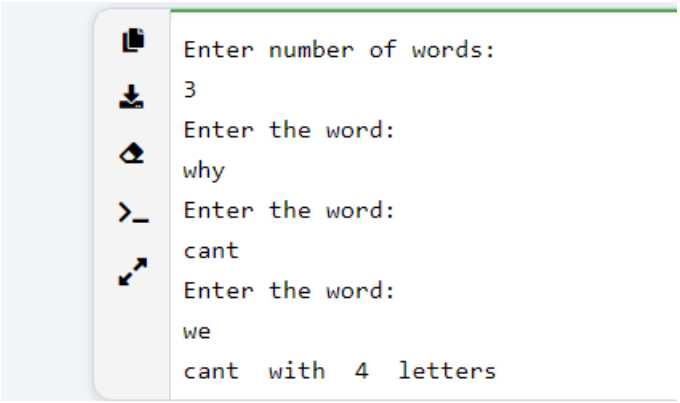
AIM: Accept a list of words and return length of longest word.

CODE

```
cnt=int(input("Enter number of words:"))
a=[]
com=0
for i in range(0,cnt):
 nw=input("Enter the word:")
 a.append(nw)


for i in range(0,cnt):
 cmp=len(a[i])
 if cmp>com:
  com=cmp
  j=i
print(a[j]," with ",com," letters")
```

OUTPUT



EXPERIMENT 29

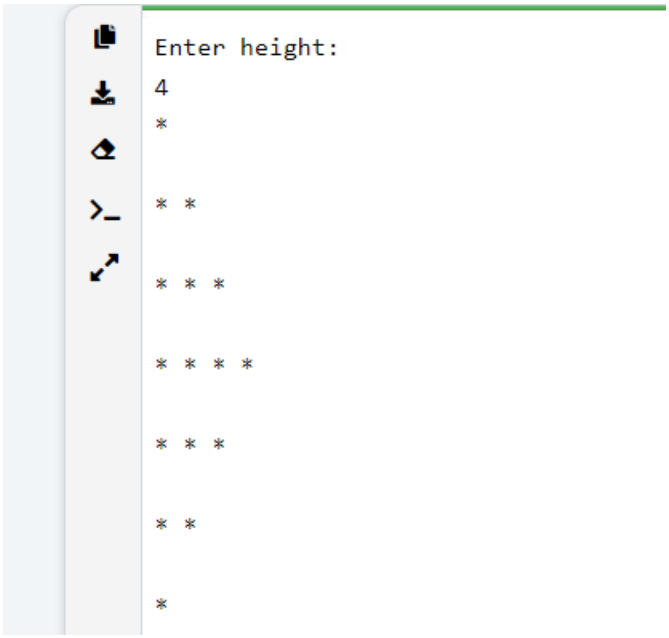
AIM: Construct following pattern using nested loop

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

CODE

```
ht=int(input("Enter height:"))
for i in range(0,ht):
 for j in range(0,i+1):
  print("* ",end="")
 print("\n")
for k in range(0,ht):
 for l in range(0,j-k):
  print("* ",end="")
 print("\n")
```

OUTPUT



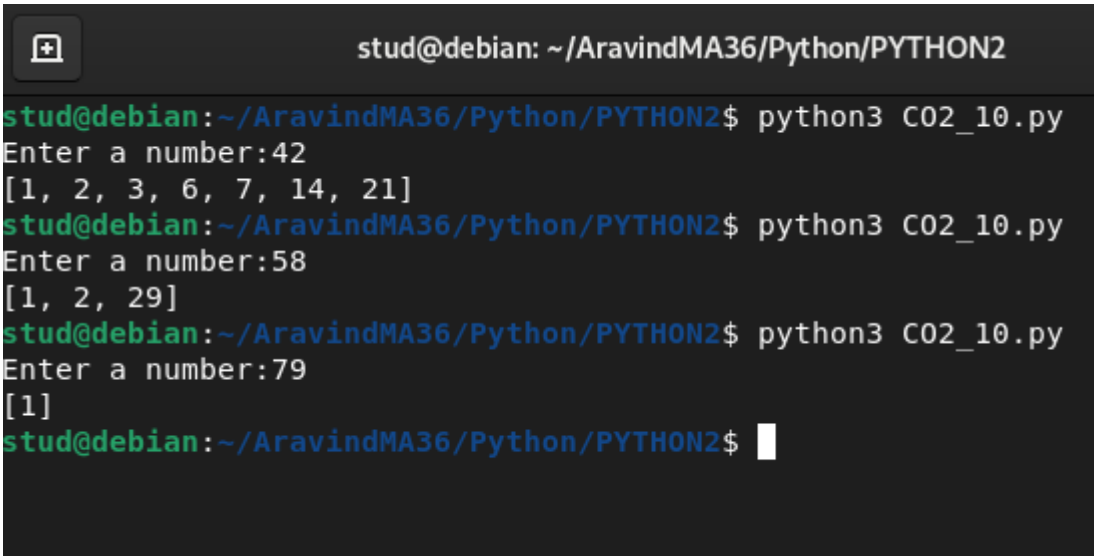
EXPERIMENT 30

AIM: Generate all factors of a number**.**

CODE

```
num=int(input("Enter a number:"))
fact=[]
for i in range(1,num):
 if num%i==0:
  fact.append(i)
print(fact)
```

OUTPUT



COURSE OUTCOME-3

EXPERIMENT 31

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

Terminal(Windows):

mkdir graphics

cd graphics

notepad rectangle

notepad circle

notepad __init__.py

mkdir dgraphics

cd dgraphics

notepad __init__.py

notepad cuboid.py

notepad sphere.py

CODE

1) Rectangle

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

2) Circle

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

3) Sphere

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

4) Cuboid

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

CODE

```
from graphics import rectangle as rt

from graphics import circle

from graphics dgraphics import *

r=rt.Rectangle(10,12) print("RECTANGLE\n")

print("length =",r.length)

print("width =",r.width)

print("area=",r.area())

print("perimeter=",r.perimeter())


c=circle.Circle(12)

print("CIRCLE\n")

print("radius =",c.radius)

print("area=",c.area())

print("perimeter=",c.perimeter())

s=sphere.Sphere(12)

print("SPHERE")

print("radius =",s.radius)

print("area=",s.area())

print("volume=",s.volume())

cu=cuboid.Cuboid(13,11,14) print("CUBOID\n")

print("length =",cu.l)

print("width =",cu.w)

print("height =",cu.h)

print("area=",cu.area())

print("volume=",cu.volume())
```

OUTPUT

```
RECTANGLE
length = 10
width = 12
area= 120
perimeter= 44

CIRCLE
radius = 12
area= 452.3904
perimeter= 75.3984

SPHERE
radius = 12
area= 1809.5616
volume= 7238.246399999999
CUBOID

length = 13
width = 11
height = 14
area= 958
volume= 2002
```

COURSE OUTCOME-4

EXPERIMENT 32

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

CODE

```
class Rectangle:
        def __init__(self,ln,br):
                self.ln=ln
                self.br=br
        def area(self):
                p=self.ln*self.br
                return p
        def perim(self):
                q=2*(self.ln+self.br)
                return q
a=int(input("Enter length of the first rectangle:"))
b=int(input("Enter breadth of the first rectangle:"))
r1=Rectangle(a,b)
a=int(input("Enter length of the second rectangle:"))
b=int(input("Enter breadth of the second rectangle:"))
r2=Rectangle(a,b)

print("Perimeter of frist rectangle= ",r1.perim())
print("Perimeter of second rectangle= ",r2.perim())
m=r1.area()
n=r2.area()
if m>n:
        print("First rectangle has the largest area")
else:
        print("Second rectangle has the largest area")
```

OUTPUT

```
Enter length of the first rectangle:12
Enter breadth of the first rectangle:34
Enter length of the second rectangle:55
Enter breadth of the second rectangle:12
Perimeter of frist rectangle=  92
Perimeter of second rectangle=  134
Second rectangle has the largest area
>
```

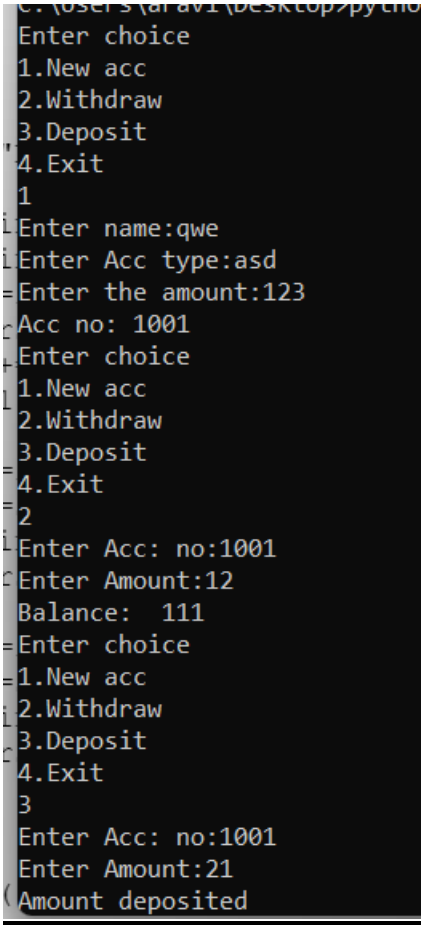
EXPERIMENT 33

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

CODE

```
bnkno=1000
holders=[10]
ptr=0
class Bank:
        def __init__(self,accno,accnme,acctype,accbal):
                self.accno=accno
                self.accnme=accnme
                self.acctype=acctype
                self.accbal=accbal
        def deposit(self,dep):
                self.accbal=self.accbal+dep
                print("Amount deposited")
        def withdraw(self,wit):
                if self.accbal>=wit:
                        self.accbal=self.accbal-wit
                        print("Balance: ",self.accbal)
                else:
                        print("Insufficient balance.")
l=0
while(l==0):
        ch=int(input("Enter choice\n1.New
acc\n2.Withdraw\n3.Deposit\n4.Exit\n"))
        if ch==1:
                name=input("Enter name:")
                tipe=input("Enter Acc type:")
                start=int(input("Enter the amount:"))
                holders[ptr]=(Bank(bnkno,name,tipe,start))
                bnkno+=1
                print("Acc no:",bnkno)
                ptr+=1
        if ch==2:
                srch1=int(input("Enter Acc: no:"))
                srch1=srch1-1001
                amt1=int(input("Enter Amount:"))
                holders[srch1].withdraw(amt1)
        if ch==3:
                srch1=int(input("Enter Acc: no:"))
                srch1=srch1-1001
                amt1=int(input("Enter Amount:"))
                holders[srch1].deposit(amt1)
        if ch==4:
                l+=1
```

OUTPUT



```
C:\Users\aravi\Desktop\pytho
Enter choice
1.New acc
2.Withdraw
3.Deposit
4.Exit
1
Enter name:qwe
Enter Acc type:asd
Enter the amount:123
Acc no: 1001
Enter choice
1.New acc
2.Withdraw
3.Deposit
4.Exit
2
Enter Acc: no:1001
Enter Amount:12
Balance:  111
Enter choice
1.New acc
2.Withdraw
3.Deposit
4.Exit
3
Enter Acc: no:1001
Enter Amount:21
Amount deposited
```

EXPERIMENT 34

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

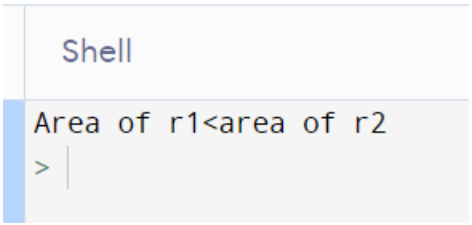
CODE

```
class Rectangle:
    def __init__(self,l,b):
        self.__length=l
        self.__width=b
    def __lt__(self,ob):

        if((self.__length*self.__width)<(ob.__length*ob.__width)):
            return True
        else:
            return False

r1=Rectangle(15,12)
r2=Rectangle(34,44)
if(r1<r2):
    print("Area of r1<area of r2")
elif(r2<r1):
    print("Area of r2<area of r1")
else:
    print("Area of r1=area of r2")
```

OUTPUT



EXPERIMENT 35

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

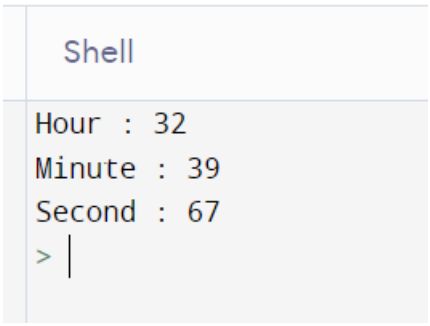
CODE

```
class Time:
        def __init__(self,h,m,s):
                self.__hour=h
                self.__minute=m
                self.__second=s
        def __add__(self,ob):
                hour=self.__hour+ob.__hour
                minute=self.__minute+ob.__minute
                second=self.__second+ob.__second
                t=Time(hour,minute,second)
                return t

        def print_it(self):
                print("Hour :",self.__hour)
                print("Minute :",self.__minute)
                print("Second :",self.__second)

t1=Time(10,10,10)
t2=Time(20,20,20)
t3=t1+t2
t3.print_it()
```

OUTPUT



EXPERIMENT 36

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

CODE

```
class Publisher:
        def __init__(self,name):
                self.name=name
                class Book(Publisher):
        def __init__(self,name,title,auther):
                super().__init__(name)
                self.title=title
                self.auther=auther
        def print_function(self):
                print("This Fuction is a member fuction of class Publisher")
class Python(Book):
        def __init__(self,name,title,auther,price,nop):
                super().__init__(name,title,auther)
                self.price=price
                self.nop=nop
        def print_function(self):
                print("Name :",self.name)
                print("Title :",self.title)
                print("Auther :",self.auther)
                print("Price :",self.price)
                print("Number of Pages :",self.nop)
p1=Python("Text book","Python Programming","Mr.abc",100,500)
p1.print_function()
p2=Book("a","b","c")
p2.print_function()
```

OUTPUT

```
Shell
Name : Novel
Title : Roadside Picnic
Auther : Arkady Strugatsky
Price : 200
Number of Pages : 350
```

COURSE OUTCOME-5

EXPERIMENT 37

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

CODE

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

a.txt

On January 15, 2001, Jimmy Wales[6] and Larry Sanger launched Wikipedia; Sanger coined its name as a portmanteau of "wiki" and "encyclopedia."[7][8] Wales was influenced by the "spontaneous order" ideas associated with Friedrich Hayek and the Austrian School of economics, after being exposed to these ideas by Austrian economist and Mises Institute Senior Fellow Mark Thornton.[9] Initially available only in English, versions in other languages were quickly developed. Its combined editions comprise more than 58 million articles, attracting around 2 billion unique device visits per month and more than 17 million edits per month (1.9 edits per second) as of November 2020.[10][11] In 2006, *Time* magazine stated that the policy of allowing anyone to edit had made Wikipedia the "biggest (and perhaps best) encyclopedia in the world."[12]

OUTPUT

```
C:\Users\aravi\Desktop>python3 prog.py
['On January 15, 2001, Jimmy Wales[6] and Larry Sanger launched Wikipedia; Sanger coined its name as a portmanteau of "w
iki" and "encyclopedia."[7][8] Wales was influenced by the "spontaneous order" ideas associated with Friedrich Hayek and
 the Austrian School of economics, after being exposed to these ideas by Austrian economist and Mises Institute Senior F
ellow Mark Thornton.[9] Initially available only in English, versions in other languages were quickly developed. Its com
bined editions comprise more than 58 million articles, attracting around 2 billion unique device visits per month and mo
re than 17 million edits per month (1.9 edits per second) as of November 2020.[10][11] In 2006, Time magazine stated tha
t the policy of allowing anyone to edit had made Wikipedia the "biggest (and perhaps best) encyclopedia in the world."[1
2]']
```

EXPERIMENT 38

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

CODE

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

a.csv

```
Id,Name,Desig,Salary
001,Aravind M,Manager,100000
002,Aravind S,Secretary,30000
003,Arjun,Deputy Manager,25000
```

OUTPUT

```
C:\Users\aravi\Desktop>python3 prog.py
['Id', 'Name', 'Desig', 'Salary']
['001', 'Aravind M', 'Manager', '100000']
['002', 'Aravind S', 'Secretary', '30000']
['003', 'Arjun', 'Deputy Manager', '25000']
```