

## Program No:- 21

Aim:- python programs to enter 2 lists of integers.  
check whether lists are same length, list sum to same value, whether any value occur in both.

## Program Code:-

```
From collections import defaultdict
```

```
def areEqual (arr1, arr2, n, m):
```

```
if (n != m):
```

```
    return False
```

```
Count = defaultdict(int)
```

```
for i in arr1:
```

```
    Count[i] += 1
```

```
for i in arr2:
```

```
    if (Count[i] == 0):
```

```
        return False
```

```
    else:
```

```
        Count[i] -= 1
```

```
return True
```

```
arr1 = [3, 5, 2, 5, 2]
```

```
arr2 = [2, 3, 5, 5, 2]
```

```
n = len(arr1)
```

```
m = len(arr2)
```

```
If (areEqual (arr1, arr2), n, m));
```

```
Print ("Yes")
```

```
else:  
    print("No")
```

```
list1 = [3, 5, 2, 5, 2]
```

```
list2 = [2, 3, 5, 5, 2]
```

```
if set(list1).intersection(list2):
```

```
    print('lists have elements in common')
```

```
else:
```

```
    print('No elements in common')
```

Result:

The program has been executed and the output was verified.

Output:

Yes

List have elements in common

possible to merge sorted

: (0, 2, 4, 6, 8, 10, 12) l1,

; (0, 1,

several answers

(0, 1) + (0, 2, 4, 6, 8, 10, 12)

; 100% ok

i =  $\lceil \frac{1}{2} \rceil$  takes

; 6 ms ok

: (0, 1, [i] takes) ? i

several answers

ok

i =  $\lceil \frac{1}{2} \rceil$  takes

ok

[0, 1, 2, 3]

{1, 2}

{0, 1}

; ((0, 1, 2, 3), 100%) l1

l2

Date:- 27.1.21

## Program No:- 20

Aim:- Python program to display future leap years from current year to a final year entered by user.

Program code:-

```
import calendar  
  
def leap_year (year, number_of_years):  
    leap_year_counter = 0  
  
    while leap_year_counter < number_of_years:  
        if calendar.isleap (year):  
            print ("{} is a leap year!".format (year))  
            leap_year_counter += 1  
  
        year += 1  
  
leap_year (2016, 20)
```

Result:-

The program has been executed and the output was verified.

Output:

2016 is a leap year!

2020 is a leap year!

2024 is a leap year!

2028 is a leap year!

2032 is a leap year!

2036 is a leap year!

2040 is a leap year!

2044 is a leap year!

2048 is a leap year!

2052 is a leap year!

2056 is a leap year!

2060 is a leap year!

2064 is a leap year!

2068 is a leap year!

2072 is a leap year!

2076 is a leap year!

2080 is a leap year!

2084 is a leap year!

2088 is a leap year!

2092 is a leap year!

program No:- 23

Date:- 3.2.21

Aim:- python program to form a list of integers, create a list removing even numbers.

Program code:-

```
list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
print ("original list")
```

```
print (list)
```

```
for i in list:
```

```
    if (i % 2 == 0)
```

```
        list.remove(i)
```

```
print ("list after removing EVEN Numbers:")
```

```
print (list)
```

Result:-

The program has been executed and the output was verified.

Output:

Original list:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

List after removing EVEN numbers:

[1, 3, 5, 7, 9]

## Program No:- 24

Aim:- python program to generate a list of four digit numbers in a given range will all their digits even and the number is a perfect square.

Program Code:-

```
def call():
```

```
n=0
```

```
for x in range(1000, 10000, 1):
```

```
    num = str(x)
```

```
    number = int(x)
```

```
    first = int(num[0])
```

```
    second = int(num[1])
```

```
    third = int(num[2])
```

```
    fourth = int(num[3])
```

```
    if first % 2 == 0:
```

```
        if second % 2 == 0:
```

```
            if third % 2 == 0:
```

```
                if fourth % 2 == 0:
```

```
                    for i in range(2, number):
```

```
                        if i * i == number:
```

```
                            print(number)
```

```
call()
```

Result:-

The program has been executed and the output was verified.

Output:-

4624

6084

6400

8464

[0, P, E, T, E, Z, H, S, C]

(tail loop)

(tail)

: tail

(o == o + i) ? i

(i) : moves tail

(tail after removed even numbers):

(tail)

nothing left bad situations good and more  
better in?

Program No:- 05

Date: 3.2.21

Aim:- Python program to count the number of characters (character frequency) in a string.

Program code:

```
def char_frequency(str1):  
    dict = {}  
  
    for n in str1:  
        keys = dict.keys()  
        if n in keys:  
            dict[n] += 1  
        else:  
            dict[n] = 1  
  
    return dict  
  
print(char_frequency('python program'))
```

Result:-

The program has been executed and the output was verified.

Output:

{'id': 1, 'name': 'John', 'age': 25, 'city': 'New York', 'isEmployed': True}

Code:

:6) class

: (1,000001,0001) open ai  
(x) r+w = mode

(x) t+ai = mode

([0] mode) +ai = terit

([1] mode) +ai = baozi

([2] mode) +ai = briar

([3] mode) +ai = dianzi

: O == C .! +terit ?!

: O == C .! +baozi ?!

: O == C .! +briar ?!

: O == C .! +dianzi ?!

(mode, 0) open ai rot

readmode = i \* i ?!

(readmode) print

docs further add more information

Program No.: - 26

Aim:- python program to adding 'ly' at the end of a given string. If it already end with 'ing', then add 'ly'. ~~to break the loop~~

Program code:-

```
def add_string(str1):
    length = len(str1)
    if length > 0:
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else:
            str1 += 'ing'
    return str1

print(add_string('program'))
print(add_string('programm'))
print(add_string('programming'))
```

Result:-

The program has been executed and the output was verified.

Output:

programming

is the process of giving  
instructions to a computer  
in a language it can understand

Program

Programming

```
int i, j, k;
char a[10], b[10];
for(i = 0; i < 10; i++)
    a[i] = 'a' + i;
for(j = 0; j < 10; j++)
    b[j] = 'b' + j;
for(k = 0; k < 10; k++)
    cout << a[k] << b[k];
cout << endl;
```

copy - character by character

Input output operations  
are carried out through  
standard input and output streams

program No:- 27

Aim:- Python program to accept a list of words and return length of longest word.

program code:-

```
def find_longest_word(words_list):
    word_len = []
    for n in words_list:
        word_len.append(len(n))
    word_len.sort()
    return word_len[-1][0], word_len[-1][1]

result = find_longest_word(["python", "PHP", "Java"])
print("The longest word:", result[0])
print("Length of the longest word:", result[1])
```

Result:-

The program has been executed and the output was verified.

Output:

Longest word : python

Length of the longest word : 6

(int) points to

(int) n =

: C < dpo

point = [::e-] int i

'p' = +int

: int

'point' = +int

(int) arr

((complicated)) points - int

((complicated)) points - int

((complicated)) points - int

arr[0] with two instances need each component  
variable

Program No:- 28

Aim:- Python program to check the length of lists.

Program code:-

```
List1 = [11, 13, 15, 19, 18, 15]
```

```
List2 = [16, 14, 13, 15, 12, 11]
```

```
len1 = len(List1)
```

```
len2 = len(List2)
```

```
if len1 == len2:
```

```
    print('both list have equal length')
```

```
else:
```

```
    print('both list doesnot have equal length')
```

Result:-

The program has been executed and the output was verified.

Maya -

Output:-

both list have equal length.

so we can compare them  
if both list have equal length  
then we can compare them

if both list have equal length  
then we can compare them

(tail\_brow) brow - respond - hair  
[ ] = ast\_brow

[ ] = tail\_brow or ast\_brow

((a,m)post) brow - col\_brow

( ) brow . ast\_brow

[1][1-] ast\_brow , [0][1-] ast\_brow or

( "BROW" , "GHQ" , "moothsq" ) brow - respond - hair : true

( [1] + user " : brow respond( ) " )

( 0 ) user " : brow respond with to depend ) "

user with has between mood and morphology  
between

Program No:- 29

Date : 14.0.21

Aim:- python programs to generate positive list of numbers from a given list of integers.

Program code:-

```
list1 = [1, -1, 2, -2, -7, 7, -10, 13, -10, 11, 4]
pos = list()
for i in list1:
    if i > 0:
        pos.append(i)
print('Original list:', list1)
print('positive integer list:', pos)
```

Result:-

The program has been executed and the output was verified.

Output:-

original list :  $[1, -1, 2, -2, -7, 7, -10, 13, -10, 11, 4]$

positive integer:  $[1, 2, 7, 13, 11, 4]$

$[1, 2, 7, 13, 11, 4]$

$[1, 2, 7, 13, 11, 4]$

$(1 \text{ tail})_{\text{odd}}$

$(0 \text{ tail})_{\text{odd}}$

: cost = 16 ml

('drop tails and tail add') thing

('tail tails and tails add tail add') thing

two add two numbers and sum of two add one by one

by Rishabh

Program No:- 30

Aim:- Python program to find biggest of 3 numbers entered.

Program 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 is the biggest number')  
  
elif b > a and b > c:  
    print('b is the biggest number')  
  
else:  
    print('c is the biggest number')
```

Result:-

The program has been executed and the output was verified.

Output:-

enter first number: 5

enter second number: 10

enter third number: 15

[P, A, O - E, G - F, F - G - S, I - J]

C is the biggest number.

(i) banggap-eog

(I+2I': t2I: loaijiro)

(eog 't2I: vagtoj avitizagi)

Program No:- 31

Aim:- python programs to Create a list of colors from comma Separated color names entered by user. Display first and last colors.

Program Code:-

```
colors = (input ('enter colors separated by commas:')).  
          split(',')  
  
print ('first color:' colors [0])  
print ['Last color:' colors [len(colors)-1]]
```

Result:-

The program has been executed and the output was verified.

Output:-

Entered colors Separated by commas: orange, green, blue.

first color: orange

Last color: blue. ((radon) tuqai) to

((radon) tuqai) to

((radon) bright) tuqai) to

((radon) tuqai) to

((radon) tuqai) to

((radon) tuqai) to

(((radon) tuqai) to) between and and carpet  
position

Program No:- 32

Aim:- Python program to print out all colors from color-list1 not contained in color-list2.

Program code:-

```
color-list-1 = set(["white", "Black", "Red", "Orange", "Blue",
                    "Yellow", "Brown"])
```

```
color-list-2 = set(["Red", "Green", "White"])
```

```
print(color-list-1.difference(color-list-2))
```

Result:-

The program has been executed and the output was verified.

Output:-

[Orange, 'Brown', 'Black', 'Blue', 'Yellow', 'White']

(: scanner pd int 1000 10100 int 071) tuqni  
(1, 1) tuqni

([0] erolo, color, torif)

[1- (erolo) 00] erolo, 'color' 10100 1021]

tuqni with two buttons need 2nd margin  
button

Program No:- 33

Aim:- python programs to 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 find area and perimeter of figures by different importing statements.

Program code:-

circle.py

```
def area(r):
    print(f'area of circle with radius {r}, is {3.14 * r * r} sq units')
```

def circumference(r):
 print(f'circumference of circle with radius {r}, is {2 \* 3.14 \* r} units')

rectangle.py

```
def area(a, b):
    print(f'area of rectangle with sides {a} and {b}, is {a * b} sq units')
```

def perimeter(a,b):  
print('perimeter of rectangle with sides',  
'a, and b, is:', '1..2f' '1. (2\*(a+b)),  
'units')

sphere.py  
def area(r):  
print('area of sphere with radius', 'r, is:',  
'1..2f', '1. (4 \* (3.14 \* r \* r)), 'sq. units')

def perimeter(r):

print('perimeter of (great circle of) sphere  
with radius', 'r, is:', '1..2f' '1. (2 \* 3.14 \* r),  
'units')

cuboid.py

def area(l,b,h):

print('total surface area of cuboid with  
dimensions', 'l, ', 'b, ', 'h, is:', '1..2f' '1. (2 \* (l \* b) \*  
(b \* h) + (l \* h)), 'sq. units')

def perimeted(l,b,h):

print('perimeted of cuboid with dimensions',  
'l, ', 'b, ', 'h, is:', '1..2f' '1. (4 \* (l + b + h)), 'units')

## Find Perimeter.py

```
import circle
from rectangle import *
from graphics.3D-graphics import Cuboid, Sphere

a = float(input('Enter length of the rectangle:'))
b = float(input('Enter breadth of the rectangle:'))
perimeter(a, b)

r = float(input('Enter the radius of the circle:'))
circle.circumference(r)

l = float(input('Enter length of the cuboid:'))
b = float(input('Enter breadth of the cuboid:'))
h = float(input('Enter height of the cuboid:'))

Cuboid.perimeter(l, b, h)

r = float(input('Enter the radius of the sphere:'))
sphere.perimeter(r)
```

## Find Area.py

```
import circle
from rectangle import *
```

```
From graphics import cuboid, sphere  
a = float (Input ('enter length of the rectangle:'))  
b = float (Input ('enter breadth of the rectangle:'))  
area (a,b)  
  
r = float (Input ('enter the radius of the circle:'))  
circle. area (r)  
  
l = float (Input ('enter the length of the cuboid:'))  
b = float (Input ('enter the breadth of the cuboid:'))  
h = float (Input ('enter the height of the cuboid:'))  
cuboid. area (l,b,h)  
  
r = float (Input ('enter the radius of the sphere:'))  
sphere. area (r)
```

Result:-

The program has been executed and the output was verified.

Output:-

enter length of the rectangle: 4

enter breadth of the rectangle: 3

enter breadth of the rectangle: 3

perimeter of rectangle with sides 4.0 and  
3.0 is 14 units.

enter the radius of the circle: 2

circumference of circle with radius 2 is  
12.56 units.

enter length of the cuboid: 5

enter breadth of the cuboid: 4

enter height of the cuboid: 3

perimeter of cuboid with dimensions 5.0,  
4.0, 3.0 is 48. units.

enter the radius of the sphere: 2

perimeter of sphere with radius 2.0 is 12.56  
units.

enter length of rectangle: 2

enter breadth of rectangle: 3

Area of rectangle with radius 4.0 is 50  
sq. units.

enter length of cuboid: 4

enter breadth of cuboid: 7

enter height of cuboid: 2

Total surface area of cuboid with dimensions 4.0, 7.0, 2.0 is 100. sq. units.

enter the radius of the sphere: 1

Area of square with radius 1 is 10.56  
sq. units.

: (r) returning  
edge (to start more) to returning) from  
depths 40. 1. '70. 1. 121. & ceiling  
(2nd)

B7 b

other sides to next surface input) will  
(dist) 40. 1. '70. 1. 121. & ceiling  
(ceiling - 40. ((dist) + 0.1)

other sides to next surface input) will  
(ceiling - ((dist + 1) \* 0.1) to

Date:- 17.2.21

Program No:- 34

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

Program code:-

```
class Rectangle:  
    def __init__(self, l, b):  
        self.length = l  
        self.breadth = b  
  
    def area(self):  
        return self.length * self.breadth  
  
    def perimeter(self):  
        return 2 * (self.length + self.breadth)  
  
    def compare(self, obj):  
        if self.area() > obj.area():  
            print('Rectangle with length =', self.length,  
                  'and breadth =', self.breadth, 'has  
                  the greater area')
```

if self.area < obj.area:  
    print ("rectangle with length =", obj.length,  
          "and breadth =", obj.breadth, "has the  
          the greater area")

else:

    print ("They have equal area!")

y1. Rectangle (8,4)

y2. Rectangle (1,3)

y1. compare(y2)

Result:-

The program has been executed and the output was verified.

Output:-

Rectangle with length = 8 and breadth = 3 has  
the greater area.

$$:(d, l, H, z) - \text{val} - 7102$$

$d = \text{atboard}, 7102$

$d = \text{atboard}, 7102$

$$:(7102) \rightarrow \text{val} - 701$$

$\text{atboard}, 7102 * \text{atboard}, 7102$  or vice

$$:(7102) \rightarrow \text{val} - 701$$

$(\text{atboard}, 7102 * \text{atboard}, 7102) * 6$  or vice

$$:(100, 7102) \rightarrow \text{val} - 701$$

$\text{and}, 7102 ! * \text{atboard}, 100 & (\text{atboard}, 7102, 701)$

$\text{and}(\text{atboard}, 7102, 701) \rightarrow \text{val}$

$(\text{atboard}, 'pos', 701)$

$\text{and}(\text{atboard}, 701)$

**Aim:-** python program to create a Bank account with members, account number, name, type of account and balance. write constructor and methods, to deposit at the bank and withdraw an amount from the bank.

**Program Code:**

```

class BankAccount:
    def __init__(self, acc, n, t, bal):
        self.account_no = acc
        self.name = n
        self.type = t
        self.balance = bal

    def deposit(self, acc):
        self.balance = acc
        print('Rs.', acc, 'deposited. Current balance is Rs.', self.balance)

    def withdrawal(self, acc):
        if self.balance >= acc:
            self.balance -= acc
            print('Rs.', acc, 'withdrawn. Current balance is Rs.', self.balance)

```

else:  
    print ('Insufficient balance to make this transaction')

a = int (input ('Enter account number:'))

b = input ('Enter name of the account holder:')

t = input ('Enter account type:')

Rs. bal = float (input ('Enter your balance:'))

a1 = Bank Account (acc, n, t, bal)

a1.deposit (float (input ('Enter amount to deposit'))))

a1.withdraw (float (input ('Enter amount to withdraw:'))))

Result:-

The program has been executed and the output was verified.

output:

enter account number: 00101432012311

enter name of the account holder: Adm

enter account type: savings

enter your balance: 200000

enter amount to deposit:- 300000

Rs. 300000 deposited current balance is Rs.  
500000.

entered amount to withdraw: 100000

Rs. 100000 withdrawn current balance is Rs.  
400000.

(200, 7152) transaction 7st

200 = acc. 7152

(200, 7152) transaction 7th  
(acc. 7152 deb. 200 in balance)

(200, 7152) transaction 7th

200 = acc. 7152

200 = posted 7152

over withdrawal (acc. 7152)

Date: - 17. 2. 01

Program No:- 36

Aim:- Python program to create rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two rectangle object by their area.

Program code:-

class Rectangle:

def \_\_init\_\_(self, l, b):

self.length = l

self.breadth = b

def area(self):

return self.length \* self.breadth

def perimeter(self):

return 2 \* (self.length + self.breadth)

def compare(self, obj):

if self.area() > obj.area():

print('rectangle with length= ', self.length,  
'and breadth= ', self.breadth,  
'has the greater area!')

if self. area < obj. area:  
    print ('rectangle with length =', obj. length,  
          'and breadth =', obj. breadth, 'has  
          the greater area')

else:

    print ('They have equal area!')

r1 = Rectangle (10, 5)

r2 = Rectangle (2, 4)

rs. compare (r2)

Result:-

The program has been executed and the output was verified.

Output:-

rectangle with length = 10 and breadth = 5  
area = 50

length > breadth

length < breadth

length = breadth

length <= breadth

length >= breadth

length < breadth

length >= breadth

length <= breadth

length > breadth

length <= breadth

length >= breadth

Date:- 17.2.21

Program No:- 37

Aim:- python programs to create a class Rectangle with private attribute length and width. Overload ' $<$ ' operator to compare the area of 2 rectangles.

Program code:-

class Rectangle:

def \_\_init\_\_(self, l, w):

self.length = l

self.width = w

self.area = self.width \* self.length

def \_\_lt\_\_(self, other):

if self.area < other.area:

print('rectangle with length = ', self.length, 'and width = ', self.width, 'has the lesser area!')

elif other.area < self.area:

print('rectangle with length = ', other.length, 'and width = ', other.width, 'has the lesser area!')

else: print ('they have equal area!')

l = float (input ('enter length of first rectangle'))

w = float (input ('enter width of first rectangle'))

R1 = Rectangle (l, w)

l = float (input ('enter length of second rectangle'))

w = float (input ('enter width of second rectangle'))

R2 = Rectangle (l, w)

R1 < R2.

Result:-

The programs has been executed and the output was verified.

Output:-  
enter length of first rectangle: 10  
enter width of first rectangle: 5  
enter length of second rectangle: 5  
enter width of second rectangle: 10

They have equal area.

$$(l_1 \times w_1) = l_2 \times w_2$$

$$10 \times 5 = 5 \times 10$$

$$50 = 50$$

$$d_{\text{ratio}}(l_1, w_1) > d_{\text{ratio}}(l_2, w_2)$$

$$(l_1 \times w_1) / l_1 = w_1$$

$$w_1 > l_2$$

$d_{\text{ratio}}(l_1, w_1) > d_{\text{ratio}}(l_2, w_2)$  so ratio of areas is more than 1

(more than 1 so it is not equal)

$d_{\text{ratio}}(l_1, w_1) > d_{\text{ratio}}(l_2, w_2)$  so ratio of areas is more than 1

and  $d_{\text{ratio}}(l_1, w_1) > d_{\text{ratio}}(l_2, w_2)$  so ratio of areas is more than 1

(more than 1 so it is not equal)

Program No:- 38

Aim:- python program to 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 construction invocation and method overriding to do.

Program Code:-

class publisher:

def \_\_init\_\_(self, name):

self.name = name

def show(self):

pass

class Book(publisher):

def \_\_init\_\_(self, title, author, name):

self.title = title

self.author = author

publisher.\_\_init\_\_(self, name)

```
def show(self):
    pass
class python(Book):
    def __init__(self, pno, title, author, name):
        self.price = pno
        self.no_of_pages = no
    book __init__(self, title, author, name)
```

```
def show(self):
    print('Book title:', self.title)
    print('Author:', self.author)
    print('publisher:', self.name)
    print('price:', self.price)
    print('no of pages:', self.no_of_pages)
```

```
p1 = python(500, 200, 'programming with python',
            'Guido Rossum', 'ABC Books')
```

```
p1.show()
```

Result:-

The program has been executed and the output was verified.

Acyclic sort

Output:-

Books title: programming with python  
Author: Guido van Rossum

Author: Guido van Rossum  
Publisher: ABC Books

Publisher: ABC Books  
Price: 600/-

No of pages: 200.

-books

: reading

:(books, 7102) -> i1 -> f1b

books : books . 7102

:(7102) word 70b

322d,

:(readit) 2100d

(books, word 70b, 1st bit, 7102) -> i1 -> f1b

1st bit = 0 & bit 7102

!readit -> readit . 7102

(books, 7102) -> i1 -> readit, o1

Program No: 39

Program :- python program to read a file line by line  
and store into a list.

Program code:-

```
def file_read (fname):  
    with open (fname) as f:  
        c=f.readlines ()  
        print (c)
```

```
file.read ("demo.txt")
```

Result:-

The program has been executed and the output  
was verified.

Output :-

Output:-

[python is a powerful general purpose programming language. it is used in web development, data science, creating software prototypes]

Date: 21.2.21

Program No:- 40

Aim:- Python program to copy old lines of one file to other.

[41.01.01] :- [22.02.17] :- [22.02.10]

Program code:-

[22.02.22] :- [22.02.22] :- [22.02.11]

a = open('demotext', 'r')

b = open('t.txt', 'w')

c = a.readlines()

for i in range(0, len(c)):

if (i % 2 == 0):

b.write(c[i])

else:

pass

b.close()

b = open('t.txt', 'r')

d = b.read()

print(d)

a.close()

b.close()

Result:-

The program has been executed and the output was verified.

Output:-

Python is powerful general purpose programming language.  
it is used in web development, data science  
creating software prototypes]

Program No:- 41

Date:- 21.2.21

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

Program code:-

```
import csv
with open ('temp.csv', newline = '') as csvfile:
    d = csv.reader (csvfile, delimiter = ',', quotechar
                    = "'")
    for r in d:
        print (',', join (r))
```

Result:-

The program has been executed and the output was verified.

Output:

"[1, 2, 3]", "[3, 25, 56]", "[35, 30, 30]"

"[21, 40, 55]", "[71, 25, 55]", "[10, 10, 40]"

"[1, 2, 3]", "[3, 25, 56]", "[35, 30, 30]"

"[21, 40, 55]", "[71, 25, 55]", "[10, 10, 40]"

Oscillation

:((0) 0, 0) appear in

: (0 = 16 + i) 71

([i] 0) stirred

2229

2201

(1, 1, 0, 0, 0, 0)

6260

(1)

Program No:- 42

Date: 17.2.21

Aim:- Python programs to find area

Program code:

```
def area(r):
```

```
    pi = 3.14
```

```
    return pi * (r * r);
```

```
num = float (input ("Enter the value:"))
```

```
print ("Area is %.6f" % area (num));
```

Result:-

The program has been executed and the output was verified.

Output:-

entered the value:- 5

Area is 78.550000.

area =  $\pi r^2$  (cm<sup>2</sup>) or  
=  $\pi (5)^2$  cm<sup>2</sup> = 78.5 cm<sup>2</sup>

(r) ratio of 1 : 1 fair

so all four ratios are good and perfect  
because 1 : 1