

Program ::

Aim: Python Program to find square
of N numbers

```
def squaresum(n):  
    sm = 0  
    for i in range(1, n + 1):  
        sm = sm + (i * i)  
    return sm
```

n = 4

```
print("The square of N(4) numbers are")  
print('-----')  
print(squaresum(n))
```

Result: The program has been executed
and output is verified

Out Put

The Square of N(4) numbers are

30

Program : 2

Aim : Python Program to form a list
of vowels selected from a given
word

```
Sentence = input('Enter your sentence: ')
for letter in sentence:
    if letter in 'aeiou':
        print (letter)
```

Result : The program has been
executed and output was
verified

Output

Enter your sentence: aleena

a
e
e
n

Program : 3

Aim: Python Program for count the occurrences of each words in a line of text.

```
def word_count(str):  
    counts = dict()  
    words = str.split()  
  
    for word in words:  
        if word in counts:  
            counts[word] += 1  
        else:  
            counts[word] = 1
```

return counts

```
string = input('Enter your string: ')  
print(word_count(string))
```

Result: The Program has been executed and output was verified

Output

Enter your string : Python is a Programming
language , i love Python

```
{'Python': 2, 'is': 1, 'a': 1, 'Programming': 1,  
'language': 1, 'i': 1, 'love': 1}
```

Program : 4

Aim: Python Program to store a list of first names. Count the occurrence of 'a' within the list

```
def countx(lst, x):  
    return lst.count(x)
```

```
lst = ["a", "b", "l", "u", "a", "a", "P"]
```

```
print(lst)
```

```
x = 'a'
```

```
print('{} has occurred {} times'.format(x, countx  
                                         (lst, x)))
```

Result: The program has been
executed and output was
verified

Output

['a', 'b', 'l', 'u', 'a', 'a', 'p']

a has occurred 3 times

Program : 5

Aim: Python Program to get a string from an input string where all occurrences of first character replaced with '\$', except first character

```
def change_char(str1):  
    char = str1[0]  
    str1 = str1.replace(char, '$')  
    str1 = char + str1[1:]  
    return str1
```

```
string = input('enter your string: ')  
print(change_char(string))
```

Result: The program has been executed and output was verified

~~output~~

Enter your string : Steing

String\$

Program : 6

Aim: Python Program to create a string
from given string where first and
last character exchanged.

```
def change_string(str1):  
    return str1[-1] + str1[1:-1] + str1[:1]
```

```
String = input('Enter your string: ')
```

```
Print(change_string(String))
```

Result : The Program has been executed
and output was verified

Output

Enter your string : Python

Python

Program : 7

Aim : Python Program to create accept
the radius from user and find
area of circle.

```
import math
print('CIRCLE AREA')
print("*****")
r = float(input('Enter the radius of
the circle: '))
area = math.pi * r * r
print("Area of the circle is : %.2f" % area)
```

Result : The Program has been executed
and output was verified

Output

CIRCLE AREA

* * * * *

Enter the radius of the circle : 5

Area of the circle is : 78.54

Program : 8

Aim : Python program for find gcd
of 2 numbers

import math

a = int(input('First number: '))

b = int(input('Second number: '))

Print ("the gcd of a and b is : ", end="")

Print (math.gcd(int(a), int(b)))

Result : The Program has been executed
and output was verified

Output

First number : 20

Second number : 33

The gcd of a and b is : 1

Program : 8

Aim : Python program for find gcd
of 2 numbers

import math

a = int(input('First number: '))

b = int(input('Second number: '))

Print ("the gcd of a and b is : ", end="")

Print (math.gcd(int(a), int(b)))

Result : The Program has been executed
and output was verified

Output

First number : 20

Second number : 33

The gcd of a and b is : 1

Program : 9

Aim : Python Program to, From a list of integers, create a list removing even numbers

```
list = [11, 20, 3, 7, 8]
```

```
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 output was verified

output

Original list:

[11, 20, 3, 7, 8]

List after removing EVEN numbers:

[11, 3, 7]

Program : 10

Aim: Python Program to sort dictionary
in ascending and descending order

```
y = {'car': 40, 'alan': 2, 'bob': 1, 'danny': 3}
```

```
l = list(y.items())
```

```
l.sort()
```

```
Print('Ascending order is:', l)
```

```
l = list(y.items())
```

```
l.sort(reverse=True)
```

```
Print('Descending order is:', l)
```

```
dict = dict(l)
```

```
Print("Dictionary", dict)
```

Result: The program has been
executed and output was
verified.

Output

Ascending order is: [('alan', 2), ('bob', 1), ('car', 40),
('danny', 3)]

Descending order is [('danny', 3), ('car', 40),
('bob', 1), ('alan', 2)]

Dictionary { 'danny': 3, 'car': 40, 'bob': 1, 'alan': 2 }

Program : //

Aim: Python Program for merge
two dictionaries

```
def merge(dict1, dict2):  
    return (dict2.update(dict1))
```

```
dict1 = {'a': 10, 'b': 8}
```

```
dict2 = {'d': 6, 'c': 4}
```

```
Print("Merging of Dictionaries")
```

```
Print("-----")
```

```
Print(dict1)
```

```
Print(dict2)
```

```
Print("merged dictionary is : ")
```

```
Print(merge(dict1, dict2))
```

```
Print(dict2)
```

Result : The program has been executed
and output was verified

Output :

Merging of Dictionaries

{'a': 10, 'b': 8}

{'d': 6, 'c': 4}

Merged dictionary is :

None

{'d': 6, 'c': 4, 'a': 10, 'b': 8}

Program : 12

Aim - Python Program to find biggest
of 3 numbers entered.

```
Print ("finding Largest Number")
Print ("-----")
number1 = input('Enter The 1st number: ')
number2 = input('Enter The 2nd number: ')
number3 = input('Enter the 3rd number: ')
if (number1 >= number2) and (number1 >=
    number3):
    lnumber = number1
elif (number2 >= number1) and (number2 >=
    number3):
    lnumber = number2
else:
    lnumber = number3
Print ("The largest number is:", {lnumber})
```

Result : The program has been executed
and output was Verified

Output

- Finding Largest number -

Enter the 1st number : 3

Enter the 2nd number : 1

Enter the 3rd number : 4

The largest number is : 4

Program : 13

Aim: Python Program to find the Factorial of a number.

```
num = int(input('Enter your value: '))
factorial = 1
if num < 0:
    print('Sorry, Factorial does not exist
          for negative numbers')
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1, num + 1):
        factorial = factorial * i
    print("The factorial of", num, "is",
          factorial)
```

Result : The Program has been executed
and output was verified.

Output

Enter your value : 7

The factorial of 7 is 5040

Program - 14

Aim - Python Program to generate all factors of a number.

```
def Print_factors(x):
    Print("The factors of", x, "are:")
    for i in range(1, x + 1):
        if x % i == 0:
            Print(x)
num = int(input('Enter a number: '))
Print_factors(num)
```

Result: The Program has been executed
and output was verified

Output

Enter a number : 50

The Factorial of 50 are:

1

2

5

10

25

50

Program - 15

Aim - Python Program to add 'ing' at the end of a given string.

If it already ends with 'ing', then add 'ly'.

```
def add_string(stri):
    length = len(stri)
    if length > 2:
        if stri[-3:] == 'ing':
            stri += 'ly'
        else:
            stri += 'ing'
    return stri
```

```
print(add_string('add'))
```

```
print(add_string('string'))
```

Result : The program has been executed and output was verified

Output

adding
steingly

Programs : 16

Aim : Python Program to write lambda function to find area of square, rectangle, and triangle.

import math

s-area = lambda side : side * side

r-area = lambda len, ht : len * ht

t-peri = lambda a, b, c : (a+b+c)/2

t-area = lambda t-peri, a, b, c : (t-peri * (t-peri - a) * (t-peri - b) * (t-peri - c)) ** 0.5

print("Area of square (5) is : ", sarea(5))

print("Area of Rectangle (30, 20) is : ", r-area(30, 20))

print("Perimeter of triangle (5, 6, 7) is : ", t-peri(5, 6, 7))

print("Area of triangle (5, 6, 7) is : ", t-area(9, 5, 6, 7))

Result: The Program has been executed
and output was Verified

Output

Area of Square (15) is : 225

Area of Rectangle (30,20) is : 600

Perimeter of triangle (5,6,7) is : 22

Area of triangle (5,6,7) is : 14.69693845

Program : 17

Ques: Python Program to Construct following Pattern using nested Loop.



```
rows = int(input("Enter the number of rows: "))

for i in range(0, rows):
    for j in range(0, i + 1):
        print("*", end=' ')
    print()

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

Result : The program has been executed
and output was verified

Output

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

Program: 18

Aim: Python Program to accept a list of words and return length of longest word

```
def longest_word(words_list):
    word = []
    for n in words_list:
        word.append([len(n), n])
    word.sort()
    return word[-1][0], word[-1][1]
result = longest_word(["apple", "blue", "at"])
print("In Longest word: ", result[1])
print("Length of the longest word: ", result[0])
```

Result: The Program has been executed and output was verified.

Output

Longest word : apple

Length of the longest word : 5

Program : 19

BIN : Python Program to find the sum
of all Items in a List

```
Print("Sum of list items")
Print(' + * + + + + + + + + ')
Sum = 0
list = [1, 5, 7, 8, 2, 3]
For item in range(0, len(list)):
    Sum = Sum + list[item]
Print("The list is: ")
Print(list)
Print("The sum of all elements in given
list : ", Sum)
```

Result : The program has been executed
and output was verified

Output

Scans of list items

The list is :

[1, 5, 7, 8, 2, 3]

The sum of all elements in given list : 26

Program : 20

Aim : Python Program to generate a list of four digit number in a given range with all their digits even and the number is a perfect square.

```
min = int(input("Enter minimum range: "))
max = int(input("Enter maximum range: "))
a = []
a = [x for x in range(min, max+1) if
    (int(x**0.5))**2 == x and
    sum(list(map(int, str(x)))) < 10]
print(a)
```

Result : The program has been
executed and output was
verified

output

input

Enter minimum Range : 1

Enter maximum range : 50

[1, 4, 9, 16, 25, 36]

Program : 21

Aim : Python Program to generate Fibonacci Series of N terms.

```
Print("FIBONACCI SERIES")
Print(".....")
Items = int(input("Enter number of terms: "))
num1, num2 = 0, 1
Count = 0

if items <= 0:
    Print("Please enter a positive integer")
elif items == 1:
    Print("Fibonacci sequence upto", items, ":")
    Print(num1)
else:
    Print("Fibonacci sequence:")
    while Count < items:
        Print(num1)
        Series = num1 + num2
        num1 = num2
        num2 = Series
        Count += 1
```

Result: The program has been executed
and the output was verified

output

Output

FIBONACCI SERIES ,

0 1 1 2 3 5 8 13 21 34

Enter number of terms: 5

Fibonacci sequence:

0

1

1

2

3

Programs : 22

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

Eg: N=4

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

def a(lines):

```
i = 1  
l = 1  
while i <= lines:  
    i = 1  
    while l <= i:  
        temp = i + l  
        print(temp, end = " ", flush = True)  
        print(" ", end = " ", flush = True)  
    l = l + 1;  
    print()  
    i = i + 1;  
line = int(input("Enter a number: "))  
a(line);
```

Result : The Program has been executed
and output was verified

output

Enter a number : 6

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

Program : 23

Aim: Count the number of characters
(Character frequency) in a string.

test = input('Enter your string: ')

freq = []

for i in test:

 if i in freq:

 freq[i] += 1

 else:

 freq[i] = 1

print("Count of all character in given
string is: " + str(freq))

Result: The program has been executed
and output was verified

~~Output~~

Enter your string: amal jyothi

Count of all characters in given string is:

{'a': 2, 'm': 1, 'l': 1, ' ': 1, 'j': 1, 'y': 1, 'o': 1, 'h': 1, 'b': 1}

Program: 24

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

~~def leap (year):~~

~~c = 2021~~

~~while c <= year:~~

~~if c % 4 == 0 and c % 400 == 0:~~

~~ct += 1~~

~~elif c % 100 != 0:~~

~~print(c)~~

~~ct += 1~~

~~n = int(input("Enter the year :"))~~

~~leap (n)~~

Result : The program
and output was

has been executed
verified

Program: 24

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

~~def leap (year):~~

~~c = 2021~~

~~while c <= year:~~

~~if c % 4 == 0 and c % 400 == 0:~~

~~ct += 1~~

~~elif c % 100 != 0:~~

~~print(c)~~

~~ct += 1~~

~~n = int(input("Enter the year :"))~~

~~leap (n)~~

Result : The program
and output was

has been executed
verified

output

enter the year: 2032

2024

2028

2032

Program : 25

Bim: Create Positive list of numbers
from a given list of integers.

List1 = [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -67]

Pos = (list())

for i in List1:

if P > 0;

Pos.append(i)

Print("Original list : ", list1)

Print("Positive integer list: ", Pos)

Result : Program has been executed
and output was verified

Output:

Original List : [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 21, 61]
Positive Integer List : [1, 2, 9, 87, 24]

Program : 26

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

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 output was verified.

output

Enter colors separated by commas: blue, green,
Red

First color: blue

Last color: Red

output

Enter colors separated by commas: blue, green,
Red

First color: blue

Last color: Red

Program: 27

Aim: Print out all colors from color-list
not contained in color-list 2

Colors1 = set(input('Enter colors separated
by commas: ')).split(',')

Colors2: Set (input('Enter colors separated
by commas: ')).split(',')

Print ('colors in color-list1 not contained in
color-list2 are: ', list(Colors1.difference
(Colors2)))

Result: The Program has been
executed and output was verified.

output

Enter colors separated by commas: pink, blue

Enter colors separated by commas: black, blue

colors in color-list, not contained in color-lists are: ['pink']

Program : 28

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

```
str1 = input('Enter a string: ')
str2 = input('Enter another string: ')
str3 = str2[0] + str1[1:] + ' ' + str1[0] + str2[1:]
print(str3)
```

Result: The program has been executed and output was verified

For Transport

Output

Enter a string : aleena

Enter Another string : anu

aleena anu

Program : 30

Ques: Create rectangle class with attributes length and breadth and methods to find area and Perimeter. Compare two Rectangle object by their area.

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 __cmp__(self, obj):

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

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

elif 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(9, 3)

R2 = Rectangle(4, 3)

R1 cmp R2

Output:

-they have equal area.

Result: The program has been executed
and output was verified.

Program : 31

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

class BankAccount:

def __init__(self, a, n, t, b):

self.aeno=a

self.name=n

self.type=t

self.bal=b

def deposit(self, a):

self.bal+=a

print('Rs.', a, 'deposited! Current balance is :
Rs.', self.bal)

def withdraw(self, a):

if self.bal >= a:

self.bal -= a

print('Rs.', a, 'withdrawn! Current
balance is : Rs.', self.bal)

else:

print('Insufficient balance to make
this transaction!')

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

Result: The Program has been executed
and output was Verified

Output :

Enter account number : 1200198

Enter name of the account holder : sneha

Enter account type : current

Enter your balance : 10000

Enter amount to deposit : 20000

Rs. 20000.0 deposited! Current balance is : Rs. 30000

Enter amount to withdraw : 15000

Rs. 15000.0 withdrawn! Current balance is : Rs. 15000

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

t = input('Enter Account type : ')

b = float(input('Enter your balance : '))

aci = BankAccount(acid, b)

aci.deposit(float(input('Enter amount to deposit : ')))

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

Program: 32

Aim: Create a class Rectangle with private attributes length and width. overload ' $<$ ' operator to compare the area of 2 rectangle.

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 1st rectangle: '))

w = float(input('Enter width of 1st Rectangle: '))

R1 = Rectangle(l, w)

l = float(input('Enter length of 2nd rectangle: '))

w = float(input('Enter width of 2nd rectangle: '))

R2 = Rectangle(l, w)

R1 < R2

Output

Enter length of 1st rectangle: 4

Enter width of 1st Rectangle: 3

Enter length of 2nd rectangle: 5

Enter ~~length~~^{width} of 2nd rectangle: 2

Rectangle with length = 5.0 and width = 2.0
has the lesser area!

Result : The Program has been executed
and output was verified

Program : 33

Aim: Create class Time with private attributes hour, minute and second. Overload '+' operator to sum of 2 time.

class Time :

def __init__(self, hh=0, mm=0, ss=0):

self.__hour = hh

self.__minute = mm

self.__second = ss

def __add__(self, other):

second = int((self.__second + other.__second) / 60)

minute = int((self.__minute + other.__minute) / 60 + ((self.__second + other.__second) / 60))

hour = int((self.__hour + other.__hour) / 24 + self.__minute + other.__minute / 60)

Print('Time [hh:mm:ss] ', hour, ':', minute, ':', second)

T1 = Time(12, 23, 45)

T2 = Time(16, 45, 26)

T1 + T2

Result: The program has been executed and output was verified

~~output~~

Time [hh:mm:ss] 5 : 11 : 41

Program: 34

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 Pages. Write a Program that displays information about a Python Book. Use base class constructor invocation and method overriding.

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, p_no, title, author, name):

self.price = p

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: Rs.', self.price)
    print('No of Pages:', self.no.of.pages)
```

```
P1 = Python(55.90, 250, 'Programming', 'MV. Rassum',
             'ABC Books')
```

```
P1.show()
```

Result: The Program has been executed
and output was verified

content

Book title: Programming

Author : UV Rossum

Publisher : ABC Books

Price. Rs. 565.9

No. of Pages : 250

Programs : 35

Aim: Python Program to read a file line by line and store it into a list.

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

Result: The program has been executed
and output was verified

Output:

['hello!', 'aleena', 'where are you?', 'hello world']

Program : 36

Aim : Python Program to read each row from a given CSV file and Print a list of strings.

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

Result: The Program has been executed and output was verified.

Outputs:

Greens

Hai! hellos

Very good

Where are you?

Programs : 37

Aim: write a Python program to write a Python dictionary to a CSV file.
After writing the CSV file read the CSV file and display the content

import csv

field_names = ['best_book_id', 'authors', 'original_title']

books = [

{

'best_book_id': 1, 'authors': 'Suzanne Collins',
'original_title': 'The hunger games'

}

{

'best_book_id': 2, 'authors': 'J K Rowling',
'original_title': 'Harry Potter and the Philosophers'

},

{'best_book_id': 3, 'authors': 'Stephenie Meyer',
'original_title': 'Twilight'}

}

]

with open('c1.csv', 'w') as csvfile:

writer = csv.DictWriter(csvfile, fieldnames=
field_names)

```
writer.writerow(header)
writer.writerow(book)
with open('c1.csv', newline='') as csvfile:
    d = csv.DictReader(csvfile, delimiter=',')
    for i in range(10):
        print('>', d[i])
```

Result: The program has been
executed and output was
verified.

Output

1. Suzanne Collins, The hunger Games
2. J.K. Rowling , many grand Pee , Harry Potter and the Philosophers.
3. Stephenie meyer , Twilight.

Program: 38

Aim: write a Python Program to read specific columns of a given CSV file and print the content of the columns

```
import csv  
with open('c1.csv', newline = '') as csvfile:  
    d = csv.DictReader(csvfile)  
    print("Department Name")  
    for x in d:  
        print(x['branch'], x['dept'])
```

Result: The program has been executed and output was verified.

outPut

ID Department Name

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Program: 39

Aim: Python Program to copy odd lines
of file to other.

```
a = open('write.txt', 'r')
```

```
b = open('r.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('r.txt', 'r')
```

```
d = b.read()
```

```
print(d)
```

```
a.close()
```

```
b.close()
```

Result: The program has been executed
and output was verified

out put

1. hai
2. aleena
3. fgckilmngi
4. helloworld
5. Python

Program : 40

Bim : Program to find square of a number.

digit = int(input("Enter number: "))

Square = digit * digit

Print ("Square of {digit} is {square}")

Result : The program has been executed
and output was verified.

no conversion

Output

Program output is:
Enter number : 5

Square of 5 is 25

Program : 41

Aim: Python Programs to create a package graphic with modules Rectangle, Circle, and sub package 3D-graphic with modules cuboid and sphere. Include methods to find area and Perimeters of respective figures in each area module with Programs that find area and Perimeter of figure by different importing statements.

Circle.py

def area(r):

Print('Area of Circle with radius', r, 'is:',
'pi * r^2' % (3.14 * r * r), 'sq units')

def circumference(r):

Print('Circumference of Circle with radius',
r, 'is:', '2 * pi * r' % (3.14 * 2 * r), 'units')

Rectangle.py

def area(a,b):

Print('Area of rectangle with sides', a, 'and',
b, 'is:', '%.2f' % (a * b), 'sq units')

def Perimeter(a,b):

Print('Perimeter of rectangle with sides', a,
b, 'is:', '%.2f' % (2 * (a + b)), 'units')

sphere.py

def area(r):

Print('Area of sphere with radius', r ,
'is:', ' $\pi \cdot r^2 = \pi \cdot (4 * (3.14 * r * r))$ ', 'sq. units')

def Perimeter(r):

Print('Perimeter of (great circle of) sphere
with radius', r , 'is:', ' $2\pi r = 2 * 3.14 * r$ ',
'units')

cuboid.py

def area(l, b, h):

Print('Total surface area of cuboid with
dimensions', ' l ', ' b ', ' h ', 'is:', ' $2(l * b + b * h + l * h)$ ', 'sq. units')

def Perimeter(l, b, h):

Print('Perimeter of cuboid with
dimension', ' l ', ' b ', ' h ', 'is:', ' $4(l + b + h)$ ', 'units')

Find Perimeter.py

import circle

from rectangle import *

from graphics import cuboid, sphere.

from graphics import graphics

```
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.Area(r)
```

Findarea.py

```
import circle
from rectangle import *
from graphics import 3D-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 length of the cuboid: '))
b = float(input('Enter breadth of the cuboid: '))
h = float(input('Enter height of the cuboid: '))
Cuboid.area(l,b,h)

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

Output

Enter length of the rectangle : 4

Enter breadth of the rectangle : 3

Perimeter of rectangle with sides 4.0 and
3.0 is : 14.00 units.

Enter the radius of the circle : 2

Circumference of circle with radius 2.0 is :
12.56 units

Enter length of the cuboid : 5

Enter the breadth of the cuboid : 4

Enter the height of the cuboid : 3

Perimeter of the cuboid with dimensions so,

4.0, 3.0 is 48.00
cm's

Enter the radius of the sphere : 2

Perimeter of (great of circle of) sphere with
radius 2.0 is 12.56 units.

Enter length of the rectangle : 2

Enter breadth of the rectangle : 3

Area of rectangle with side 2.0 and 3.0 is :
6.00 sq. units

Enter the radius of the circle : 4

Area of circle with radius 4.0 is 50.24 sq.

Enter length of the cuboid : 4

Enter breadth of the cuboid : 7

Enter height of the cuboid : 2

Total surface area of cuboid with dimension
4.0, 7.0, 2.0 is 100.00 sq. units.

Enter the radius of the sphere : 1

Area of sphere with radius 1.0 is 12.5632 units

Program :- 42

Aim: Python Program to copy odd line
of one file to another.

```
a = open('demo.txt', 'r')
b = open('t.txt', 'w')
c = a.read().split()
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 output was verified

Output

They are distinguished from dump trucks by configuration: a dumper is usually an open A-shuttered vehicle with the load slip in front of the driver.

Program: 43

Aim: Program to check the length of list

list1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]

list2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]

len1 = len(list1)

len2 = len(list2)

if len1 == len2:

Print ("both lists have equal length")

else:

Print ("both lists do not have equal length")

Result: The program has been executed
and output was verified

Output

both lists have equal length

Program : 44

Bim: Program to check the ~~common~~ ^{Common} of list element in list.

list1 = [10, 10, 11, 12, 13, 14, 16, 15, 16, 12]

list2 = [10, 10, 11, 12, 12, 16, 14, 15, 14, 12]

for value in list1

 if value in list2

 common = 1

 if common == 1:

 Print ("There are common elements")

else:

 Print ("no common elements")

Result: The Program has been
executed and output was
verified.

an example

Output

There are common elements