Program No: 1

AIM: Python program to find area

```
def area(r):
    Pi = 3.14
    return Pi a (r*r);
num = float (input ("Enter the value for:"))
print ("Area is %.6f" % area(num));
```

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

Output

```
Enter the value for : 3
Area is 28.26000
```

Program No: 2

AIM: Python program to find largest among three numbers.

```
num1= float(input("Enter the first
                         number:"))
num2= float(input("Enter the second number:"))
num3= float(input("Enter the third number:"))
if(~~n~~num1>num2)and(num1>num3):
    largest=num1
elif(num2>num1)and(num2>num3):
    largest=num2
else
    largest=num3
print("The largest number is", largest)
```

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

Output

```
Enter the first number: 3
Enter the second number: 8
Enter the third number: 5
The largest number is 8
```

Program No:3

AIM: Python program to find square of a number.

```
digit = int(input("Enter an integer
                   number:"))
square = digit * digit
print("square of {digit} is {square}"
```

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

Output

```
Enter an integer number: 3
square of 3 is 9.
```

Program No:4

AIM: Python program to find area of circle.

```
from math import pi
r=float(input the radius of the circle:
print("The area of the circle with radius
    +str(r)+"is:"+str(pi*r**2))
```

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

Output

```
Input the radius of the circle:4
The area of the circle with radius
4.0 is: 50.2654
```

Program No:5

AIM: Python program to find square of n.

```
list1 = [14, 20, 13, 8, 6, 2]
for n in list1:
    square = n**n
    print(n, squared is", square)
```

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

Output

```
14 squared is 196
20 squared is 400
13 squared is 169
8 squared is 64
6 squared is 36
2 squared is 4
```

Program No: 6

AIM: Python program to find vowels in a string.

```
string A = ".Hello... how are you"
print("Given string:\n", stringA)
vowels = "AaEeIiOoUu.
res = set([each for each in stringA
        if each in vowels])
print("The vowels present in the
        string:\n" res)
```

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

Output

```
Given string:
    Hello... how are you
The vowels present in the string
    {'u', 'a', 'e', 'o'}
```

Program No: 7

AIM: Python program to count words in a sentence.

```
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
print(word_count("when you change the
  quality of your thinking, you change
  the quality of your life sometimes
  instantly"))
```

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

Output

```
{'when':1, 'you':2, 'change':2, 'the':2,
'quality':2, 'of':2, 'your':2, 'thinking':1,
'life':1, 'sometimes':1, 'instanly':1}
```

Program No: 8

AIM: Python program to count a in a list

```
a=['arya', 'arun', 'keerthi', 'mari']
str1 = (''.join(a))
count=0
for i in str1:
    if i ='a':
      count = count+1
print("count of a in the list is:"
                 + str(count))
```

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

Output.

```
count of a is the list is:3
```

AIM: Python program to check the length of lists

```
list1 = [10,10.11, 12, 12, 13, 14, 16, 15, 16, 12]
list 2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]
len 1 = len(list1)
len 2 = len(list 2)
if len1 == len2:
    print('both list have equal length')
else:
    print('both list doesn't have equal
        length')
```

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

Output

```
both list have equal length
```

Program No:10

AIM: Python program to check the sum of lists

```
list 1=[10,10,11,12,12,13,14,16,15,16,12]
list 2=[6,12,13,14,15,16,10,11,12,10,12]
total 1= sum(list1)
total 2=sum(list2)
if (total 1== total 2:
    print('both list have equal sum')
else:
    print('both list doesn't have equal
             sum')
```

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

Output

```
both list have equal sum.
```

Program No: 11

AIM: Python program to check the commo
elements in the list.

```
list1=[10,10,11,12,12,16,22,33,44,22]
list2=[10,10,11,12,12,16,22,33,44,22]
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 execute
and the output was verified.

Output.

```
there are common elements
```

Program No: 12

AIM: Python program to replace a character.

```
def change-char(str1):
char= str1[0]
str1 = str1.replace(char,'$')
str1 = char + str1[1:]
print(change-char('refresh'))
```

Result: The program has been execu
and the output was verified.

Output

```
refresh
```

Program No:13

AIM: Python program to exchange the first and last letter in a string

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

Result: The program has been execut
and the output was verified.

Output

```
esneappl.p.
```

Program No: 14

AIM: Python program to merge 2 dictionaries.

```
def merge(dict1, dict2)
    return(dict2.update(dict1))
dict1 = {'a':10, 'b':8}
dict2 = {'d':5, 'c':2}
print(merge(dict1, dict2))
print(dict2)
```

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

Output

```
None
{'d': 5, 'c': 2, 'a': 10, 'b': 8}
```

Program No: 15

AIM: Python program to ascend and decent dictionary

```
import operator
d={1:2,3:4,4:3,2:1,0:0}
print('original dictionary:'d)
sorted_d = sorted(d.items(), key=operator.
                          itemgetter(1))
print('Dictionary is ascending order by
                   value:', sorted_d)
sorted_d = dict (sorted(d.items(),key=operator
      itemgetter(1), reverse = True))
print('Dictionary in descending order by
                value:'sorted_d)
```

Result: The program has been execiated and the output was verified.

Output

```
Original dictionary: {1:2, 3:4, 4:3,
                      2:1, 0:0}
Dictionary in ascending order
          value: [(0,0),(2,1),(1,2),(4,3),
Dictionary in descending order by
          {3:4, 4:3, 1:2, 2:1, 0:0}.
```

Program No:16

AIM: Python program to remove even numbers from the list.

```
list = [11,22, 33,44, 55,66,77,88,99]
print(list)
for i in list:
    if(i % 2==0)
        list.remove(i)
print("list after removing:", list)
```

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

Output

```
[11,22,33,44,55,66,77,88,99]
list after removing:[11,33,55,77,9
```

Program No:17

AIM: python program to find gcd of number

```
def gcd(a,b):
  if(b==0):
     return a
  return gcd(b,a%b)

a=45
b=65
if (gcd(a,b)):
   print('GCD of', a,'and', b,'is',
                      gcd(a,b))
else:
   print('not found')
```

Result: The program has been execu
and the output was verified.

Output

```
GCD of 45 and 65 is 5
```

Program No:18

AIM: Python program to find factorial of a number.

```
num=int(input("Enter a number:"))
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 the output was verified.

Output

```
Enter a number:5
The factorial of 5 is 120.
```

Program No:19

AIM: Python program to find fibonacci sequence.

```
def reccur_fibo(n):
  if n<=1:
  return n
  else:
    return(reccur_fibo(n-1)+reccur_fibo(n-2))
returns=int(input("How many terms?"))
if return<=0:
  print("Please enter a positive integer")
else:
  print("Fibonacci sequence:")
    for i in range(nterms):
      print(return_fibo(i))
```

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

Output

```
How many terms? 4
Fibonacci sequence:
0
1
1
2.
```

Program No: 20

AIM: Python program to perform string function.

```
def add_string(str1):
   length = len(str1)
  if length>1:
    if str1[-3:]="ing":
    str1+ = 'ly'
  else:
    str1+ ='ing'
  return str1
print (add_string('do'))
print (add_string('according'))
```

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

Output

doing
accordingly

Program No:21

AIM: python program to perform the sum of given items

```
numbers=[1,2,3,4,5,2,5]
sum= sum(numbers)
print(sum)
```

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

Output

```
22.
```

Program No: 22

Aim: Python program to find perfect even square numbers in a range

```
num1= int(input ("Enter a number: "))
num2= int(input ("Enter a number: "))
for ir in range (num1, num2+1):
    for i in range (32,100+1)
        if i =j*j:
        string = str[i]
        if int(string[0])%2==0) and
           int(string[1]%2==0) and
           int (string[2] %2==0) and
           int (string[3]%2==0):
            print(i)
```

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

Output

```
Enter a number: 4444
Enter a number: 9999

4624
6084
6400
8464
```

Program No: 23

Aim: Python program to display the given pyramid with step number accepted from user.

```
lines = int(input("Enter a number:"))
i = 1
j=1
while i<= lines:
j=  j=1
    while j<=i:
        temp = i*j
        print(temp, end='', flush=True)
        print(" ", end='', flush=True)
        j=j+1
        print(" ")
    i = i+1
```

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

Output

```
Enter a number:4

1

2  4

3   6   9

4   8   12   16
```

Program No: 24

Aim: Python program to count the number of characters in a string

```
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('hellow how are you
```

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

Output

```
{'h': 2, 'e': 2, 'l': 2, 'o': 3, ' ': 3, 'w':
'a': 1, 'r': 1, 'y': 1, 'v': 1}
```

Program No: 25

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

```
def find(word):
wl =[ ]
for n in word:
    wl.append(len(n),n)
     wd
wl.sort()
result = wl[-1][0], wl[-1][1]
print ("longest word:", result[1])
print("length of the longest word:",
                    result[0])
find (["hello","morning", "hi"])
```

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

Output.

```
longest word : morning
length of the longest word:7
```

Program No: 26

Aim: Python program to construct pattern using nested loop.

```
def star():
    n=5
    for i in range(n):
        for j in range(i):
            print("*", end=" ")
        print(" ")
    for is in range(n,0,-1):
        for j in range(i):
            print("a", end=" ")
        print(" ")

star()
```

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

Output

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

Program No: 27

Aim: Python program to print 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(i)
print_factors(232)
```

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

Output

```
The factors of 232 are:

1
2
4
8
29
58
116
232.
```

Program No: 28

Aim: Python program to create lambda functions to find area of square, rectange and tragngle.

```
print("Enter the length of a side of
        square:")
s= int(input("Enter your value:"))
print("Enter the length and breadth
        of rectangle:")
l= int(input("Enter your value:"))
b= int(input("Enter your value:"))
print("Enter the base and height of
            triangle")
h= int(input("Enter your value:"))
d= int(input("Enter your value:"))
```

```
x= lambda s: s*s
y= lambda l, b: l*b
t = 0.5
z= lambda h,d,t : h*d*t
print("Area of square is": x(5))
print("Area of rectangle is :"y(l,b))
print("Area of triangle is :", z(h,d,t))
```

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

Output

```
Enter the length of a side of
                       square

Enter your value:2
Enter the length and breadth of rectang
Enter your value:4
Enter your value:2
Enter the base and height of triang
Enter your value:3
Enter your value:2
Area of square:4
Area of rectangle:8
Area of triangle:3.0
```

Program No: 29

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

```
import datetime
a=int(a.year)
b=int(input("Enter final year:"))
print("In leap years:")
for i in range(a,b+1):
    if(i%4==0):
        print(i)
```

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

Output

```
Enter final year : 2040
Leap years :
        2024
        2028
        2032
        2036
        2040
```

Program No: 30

Aim: Pytho program to generate 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 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, -5, 9, -2, -54, 87, -33, -7
                 24, -67]

positive integer list : [1, 2, 9, 87, 24]
```

Program No: 31

Aim: Python program to find biggest of 3 numbers entered

```
a=int(input('Enter 1st no:'))
b=int(input('Enter 2nd no:'))
c=int(input('Enter 3rd no:'))
if a>b and b>c:
    print(a,'is biggest number')
elif b>a and b>c:
    print(b,'is biggest number')
else:
    print(c,'is the biggest number')
```

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

Output

```
enter 1st no : 5
enter 2nd no : 6
enter 3rd no : 8
8 is the biggest number
```

Program No: 32

Aim: Python program to create a list of colours from comma-seperated colour names entered by user. Display first and last colours.

```
colors=(input('Enter colors seperated
        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

```
enter colors separated by comm
        red, black, yellow.

First color: red
Last color: yellow
```

Program No: 33

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

```
colors1 = set((input('Enter colors
    seperated by commas:')).split(','))
colors2 = set((input('Enter colors
    seperated 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 the output was verified.

Output

```
Enter colors seperated by comma:
    red, yellow, brown
Enter colors seperated by comma
    black, white
colors in color-list1 not contained"
  color-list2 are ['brown', 'red', 'yello
```

Program No: 34

Aim: Python program 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 modules. Write programs that find area and perimeter of figures by different importing statements.

circle.py

```
def area(r):
    print('Area of circle with radius',
        r,'is:','%.2f'%(3.14*r*r),'sq.units')
```

```
def circumference(r):
    print('Circumference of circle with
          radius:', r, 'is:', '%.2f'%(3.14*2*r),
                 'units')
```

<u>rectangle.py</u>

```
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", 'and', b, 'is:', '%.2f'
        %(2*(a+b)), 'units')
```

<u>Sphere.py</u>

```
def area(r):
    print('Area of sphere with radius',
    r, 'is:', '%.2f' '% .2f' %(4*(3.14*r*r)),
    'sq.units')

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


cuboid.py

def area(l,b,h):
    print('Total surface area of cuboid
        with diamentions :', l, ',', b, ',', h,
        'is:'. '%.2f', %(2*((l*b)+(b*h)+
        (l*h))), 'sq units')

def perimeter(l,b,h):
    print('perimeter of cuboid with
        diamentions :', l, ',', b, ',', h, 'is',
'%.2f'%(4*(l+b+h)), 'units')
```

FindPerimeter.py.

```
import circle
from rectangle import a
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 a
from Graphics_3D_graphics import
              cuboid,sphere
a=float(input('Enter length of the
         rectange:'))
```

```
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
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 radiu
    2.0 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 dimen
    5.0, 4.0, 3.0, is 48.00 units.
Enter the radius of the sphere:2
```

```
Perimeter of (great 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 sides 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. units.

Enter length of the cuboid: 4
Enter breadth of the cuboid: 7
Enter height of the cuboid: 2
```

```
Total surface area of cuboid with
 dimensions 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
12.56 sq. units.
```

```
def comp(self.obj):
    if self.area()>obj.area():
        print('Rectangle with length=',
            self.length, 'and', 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 (3,4)
r1.comp(r2)
```

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

Output (35)

Rectangle with length = 9 and breadth = 3 has the greater area

Program No : 36

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.

```
class BankAccount:
    def_init_(self, a,n,t,b):
    self.acno=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:'))
n=input('Enter name of the account
              holder:')
t=input('Enter account type:')
```

```
b=float(input('Enter your balance:'))
ac1=BankAccount(a,n,t,b)
ac1.deposit(float(input('Enter amount
            to deposit:'))
ac1.withdraw(float(input('Enter amount
                to withdraw:')))
```

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

Output

```
Enter account number: 00900923213
enter name of the account holder:
                              Karthik.

enter account type : savings
enter your balance : 100000
Enter amount to deposit : 300000
Rs. 300000.0 deposited! current balan
is Rs. 400000.0

enter amount to withdraw: 5000
Rs. 5000.0 withdrawn! Current
balance is: Rs. 395000.0
```

Program No: 37.

Aim: Python program to create rectangle with attributes length and breadth and methods to find area and perimeter. Compare 2 rectangle objects by their area.

```
class Rectangle:
  def_init_(self, l,b):
     self.length=1
     self.breadth=b
  def are(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)
r3 = Rectangle (3,4)
r1. cmp (r2)
```

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

Output.

```
Rectangle with length=9 and
breadth=3 has the greater area.
```

Program No: 38

Aim: Python program to create a class rectangle with private attribute length and width. Overload '<' operator to compare the area of 2 rectangles

```
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
R1 = Rectangle (l,w)
l = float(input('Enter length of 2nd
              rectangle:'))
w= float(input('Enter width of 2nd
              rectangle :'))
R2 = Rectangle (l,w)
```

$R_1 < R_2$.

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

Output

```
Enter length of the 1st rectangle:7
Enter width of the 1st rectangle:8
Enter length of the 2nd rectangle:8
Enter width of the 2nd rectangle:7
They have equal area!
```

Program No: 39

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 constructor invocation and method overriding.

```
class publisher:
def-init-(self,name1):
self.name=name1
def show(self):
pass.
```

```
class Book(publisher):
  def_init_(self, title1, author1, name1):
    self.title = title1
    self.author = author1
  Publisher._init_(self, name1)
  def show(self):
    pass.
 Class python (self
              Book):
   def_init_(self, P,no, title1, author1,
                    names):
    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(565.90, 250, 'Programming
    with Python', 'GV Rossum', 'ABC Books')
P1.show()
```

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

Output

```
Book title : Programming with
        python.
Author : GV Rossum
Publisher : ABC books.
Price : 565.9
No of Pages : 250
```

Program No: 40

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:
    # content_list is the list that contains
       the read lines.
     c = f.readlines()
     print(c)
    # print (len(c))
file_read ("demo.text")
```

Result: The program has be executed and the output was verified.

Output

```
['A trailer is a vehicle design
for carrying bulk material. often in
building states in', 'They are
distinguished from dump trucks
by configuration: a dumped']
```

Program No: 41

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

```
a= open('demo.txt','r')
b= open('1.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('1.txt','r')
d= b.read()
print(d)
```

a.close()

b.close()

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

Output.

They are distinguished from dump trucks by configuration: a dumper is usually an open 4-wheeled vehicles with the loa skip in front of the driver.

Program No: 42.

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

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

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

Output

```
"['1', '2', '3']", "[33,, 28,, 56]", "[35,, 30,, 30
"[21,, 40,, 55]", "[71,, 25,, 55]", "[10,, 10,, 40]"
"['1',, '2',, '3']", "[33,, 25,, 56]", "[35,, 30,, 30]"
"[21,, 40,, 50]", "[71, 25, 55]", "[10,, 10,, 40]
```

Program No: 43

Aim: Python program to read specific columns of a given csv file and print the content of the columns

```
import csv
with open('dep.csv', newline='') as
                                    csvfile:
d=csv.DictReader(csv file)
print ("ID Department Name")
print ("_ _ _ _ _ _ _ _ _ _ _")
for r in d:
    print [r['value'], r['data'])
```

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

Output

```
I   Depaument   Name
- - - - - - - - - - - - - -

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

Program No: 44

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

```
import csv
field_names = ['best_book_id', 'authors':
               'original_title':
               'The hunger Games'3,
{
 best_book_id': 1, 'authors': 'JK Rowling
   Mary Grand Pre', 'Original_title':
   'Harry Potter and the Philosophers
   stone' 3,
'best_book_id': 3, 'authors': 'stephanie
   meyer', 'original_title': 'twilight'}

with open('c1.csv', 'w') as csvfile:
  writer = csv.DictWriter(csvfile,
      fieldnames= field_names)
  writer = writeheader()
  writer = writerows(book)

with open('c1.csv', newline = '') as csv
                                        file:
 d= csv. reader (csvfile, delimiter = ',')
for r in d:
      print (','.join(r))
```

Output

```
best_book_id, authors, original_l
1, Suzanne Collins, The hunger Gam
2, "J.K Rowling, Mary GrandPre", Harr
        Potter and the Philosopher
3. Stephanie Meyer, Twilight.
```