OUTPUT

```
Enter the value for : 3
Area is 28.260000
```

OUTPUT

```
Enter the first number : 2
Enter the second number : 4
Enter the third number : 5
The largest number is 5
```

14/01/2021

PROGRAM NO.1

AIM : Python program to find the area of a circle

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

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

16/01/2021

PROGRAM NO.2

AIM : Python program to find largest among 3 numbers

```
number1 = float(input("Enter the first number :"))
number2 = float(input("Enter the second number :"))
number3 = float(input("Enter the third number :"))
if (number1 > number2) and (number1 > number3):
    largest = number1
elif (number2 > number1) and (number2 > number3):
    largest = number2
else:
    largest = number3
print("The largest number is ", largest)
```

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

OUTPUT

```
Enter an integer number : 4
Square of 4 is 16
```

OUTPUT

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

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
```

16/01/2021

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

26/01/2021

PROGRAM NO.4

AIM : Python program to find area of circle using math module

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

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

26/01/2021

PROGRAM NO.5

AIM : Python program to find the 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 was executed and the output was verified

OUTPUT

Given string :

Hello ... how are you

The vowels present in the string :
{'u','a','e','o'}

OUTPUT

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

26/01/2021

PROGRAM NO.6

AIM : Python program to find vowels in a string

```
stringA = "Hello ... how are you"
print (" Given string : \n", stringA)
vowels = "AaEeIoiOoUu"
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

26/01/2021

PROGRAM NO.7

AIM : Python program to count ~~vowels~~ words in a sentence

```
def word_count (str):
    counts = dict()
    words = str.split()
    for words 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

Count of a in the list is : 3

OUTPUT

Both lists have equal length

26/01/2021

PROGRAM NO.8

AIM: Python program to count 'a' in a list.

```
a = ['abhi', 'seban', 'roshan', 'joseph']
str1 = (''.join(a))
count = 0
for i in str1:
    if i == 'a':
        count += 1
print("Count of a in the list is :" + str(count))
```

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

26/01/2021

PROGRAM NO.9

AIM: Python program to check the length of lists

```
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 are of unequal length')
```

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

OUTPUT

Both lists have equal sums

OUTPUT

There are common elements

26/01/2021

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 = [16,12,13,14,15,16,10,11,12,10,12]
total 1 = sum(list 1)
total 2 = sum(list 2)
if total 1 == total 2 :
        print ('Both lists have equal sum')
else:
        print ('Both lists have unequal sum')
```

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

27/01/2021

PROGRAM NO.11

AIM: Python program to check the common elements in the lists

```
list 1 = [10,10,11,12,12,13,14,16,15,16,12]
list 2 = [10,10,11,12,12,16,14,16,15,19,12]
for value in list 1 :
        if value in list 2 :
                common = 1

if common == 1 :
        print ('There are common elements')
else :
        print ('No common elements')
```

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

OUTPUT

ref$esh

OUTPUT

eineapplp

OUTPUT

None

{'d':5, 'c':2, 'a':10, 'b':8}

27/01/2021

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 executed and the output was verified

27/01/2021

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 executed and the output was verified

27/01/2021

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 executed and the output was verified

OUTPUT

Original dictionary : {1:2, 3:4, 4:3, 2:1, 0:0}

Dictionary in ascending order by value : [(0,0), (2,1), (1,2), (4,3), (3,4)]

Dictionary in descending order by value : {3:4, 4:3, 1:2, 2:1, 0:0}

OUTPUT

[11, 22, 33, 44, 55, 66, 77, 88, 99]

List after removing : [11, 33, 55, 77, 99]

27/01/2021

PROGRAM NO.15

AIM : Python program to ascend and descend 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 in 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 executed and the output was verified

27/01/2021

PROGRAM NO.16

AIM : Python program to remove even numbers from a list

```
list = [11, 12, 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

GCD of 45 and 65 is 5

OUTPUT

Enter a number : 5
The factorial of 5 is 120

27/01/2021

PROGRAM NO.17

AIM : Python program to find gcd of 2 numbers

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

a=45
b=65
if(g(a,b)):
    print('GCD of',a,'and',b,'is',g(a,b))

else:
    print('Not found')
```

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

03/02/2021

PROGRAM NO.18

AIM : Python program to find factorial of a number.

```
num = int(input("Enter a number:"))
factorial = 1
if num < 0:
    print(" Factorial doesn't exist ")
elif num <= 1:
    print(" The factorial of {num} 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

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

OUTPUT

```
doing
accordingly
```

03/02/2021

PROGRAM NO.19

AIM: Python program to find fibonacci sequence

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

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

03/02/2021

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

15

03/02/2021

PROGRAM NO. 21

AIM: Python program to perform the sum of given items

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

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

OUTPUT

15

OUTPUT

Enter a number : 4444
Enter a number : 9999
4624
6084
6400
8464

03/02/2021

PROGRAM NO. 21

AIM: Python program to perform the sum of given items

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

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

03/02/2021

PROGRAM NO. 22

AIM: Python program do find perfect even square numbers in a Range

```
num1 = int(input("Enter a number :"))
num2 = int(input("Enter a number :"))
for i 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 :4
1
2 4
3 6 9
4 8 12 16
```

OUTPUT

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

03/02/2021

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=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

03/02/2021

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('hello how are you'))
```

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

OUTPUT

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

OUTPUT

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

03/02/2021

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)
    wl.sort()
    result = wl[-1][0], wl[-1][1]
    print("longest word :", result[1])
    print("length of the longest work :", result[0])

find(["hello", "morning", "hi"])
```

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

03/02/2021

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 i in range(n, 0, -1):
            for j in range(i):
                print("*", end = " ")
                print(" ")


star()
```

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
```

OUTPUT

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

03/02/2021

PROGRAM NO. 27

AIM: Python program to print factors of a number.

```
def print_factors(x):
    print("The factor 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

03/02/2021

PROGRAM NO. 28

AIM: Python program to write lambda functions to find area of square, rectangle and triangle

```
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(s))
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 final year : 2040
Leap years:
2024
2028
2032
2036
2040
```

OUTPUT

```
Original list : [1,-1,2,-5,9,-2,-54,87,-33,-76,24,-67]
Positive integer list : [1,2,9,87,24]
```

17/02/2021

PROGRAM NO.29

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

```
import datetime
a = datetime.datetime.now()
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

17/02/2021

PROGRAM NO.30

AIM : Python 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

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

17/02/2021

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 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 colors separated by commas : red, black, yellow
First color : red
Last color : yellow

17/02/2021

PROGRAM NO. 32

AIM : Python program to create a list of colors from comma-separated color names entered by users. 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 the output was verified

OUTPUT

```
Enter colors separated by commas : red, yellow, brown
Enter colors separated by commas: black, white
Colors in color-list1 not contained in color-list2 are:
['brown', 'red', 'yellow']
```

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 ~~length~~ breadth of the cuboid : 4
Enter height of the cuboid : 3
Perimeter of cuboid with dimensions : 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
```

17/02/2021

PROGRAM NO. 33

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

```
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 ~~by~~ in color-list2
      are :', list(colors1.difference(colors2)))
```

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

17/02/2021

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 module. 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')
```

```
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
```

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

sphere.py

```
def area(r):
    print('Area of sphere with radius', r, [redacted] 'is:', '%.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 dimensions :', l, ',', b,
     ',', h, 'is :', [redacted] '%.2f'%(2*((l+b)+(b*h)+(l*h))),
        'sq. units')


def perimeter(l,b,h):
    print('Perimeter of cuboid with dimensions :', l, ',', b, ',', h,
     'is :', '%.2f'%(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)
```

FindArea.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 :'))
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

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

17/02/2021

PROGRAM NO.35

AIM: Python program to create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects 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:')


x1 = Rectangle (9,3)
x2 = Rectangle (3,4)
x1.cmp(x2)
```

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

OUTPUT

```
Enter account number : 00900909090909
Enter name of the account holder : John
Enter account type : Savings
Enter your balance : 100000
Enter amount to deposit : 300000
Rs. 300000.0 deposited ! Current balance is Rs. 400000.0
Enter amount to withdraw : 5000
Rs. 5000.0 withdrawn ! Current balance is : Rs. 395000.0
```

17/02/2021

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 :')))
```

OUTPUT

```
Enter length of 1st Rectangle : 2
Enter width of 1st Rectangle : 3
Enter length of 2nd Rectangle : 5
Enter width of 2nd Rectangle : 2
Rectangle with length = 2.0 and width = 3.0 has the lesser area!
```

```
ac1.withdraw(float(input('Enter amount to withdraw : ')))
```

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

17/02/2021

PROGRAM NO. 37

AIM: Python program to 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
```

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

OUTPUT

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

17/02/2021

PROGRAM NO. 38

AIM : Python program to create a class time with private attribute hour, minute and second. Overload '+' operator to find 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, 25, 45)
T2 = Time (16, 45, 56)
T1 + T2
```

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

OUTPUT

Book title : Programming with Python
Author : GV Rossum
Publisher: ABC Books
Price : Rs. 565.9
No. of pages : 250

17/02/2021

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 display 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(Book):
    def __init__(self, p, no, title1, author1, name1):
        self.price = p
        self.no_of_pages = no
        Book.__init__(self, title1, author1, name1)
    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(565.90, 250, 'Programming with Python', 'GV Rossum',
        'ABC Books')
```

OUTPUT

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

OUTPUT

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

P1.show()

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

21/02/2021

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.txt")
```

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

21/02/2021

PROGRAM NO. 41

AIM : Python program to copy odd lines of one file to other

```
a=open ('demo.txt', '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

21/02/2021

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 csvfile:
        d = csv.reader(csvfile, delimiter=' ', quotechar='|')
        for r in d:
            print(', '.join(r))
```

OUTPUT

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

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

21/02/2021

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(csvfile)
    print("ID  Department  Name")
    print("- - - - - - - - - - - - - - -")
    for r in d:
        print(r['value'], r['data'])
```

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
```

OUTPUT

best_book_id, authors, original_title

1, Suzanne Collins, The Hunger Games

2, "J.K. Rowling, Mary GrandPré", Harry Potter and the Philosophers Stone

3, Stephanie Meyer, Twilight

21/02/2021

PROGRAM NO. 44

AIM: 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']
book = [
   {
'best_book_id': 1,'authors': 'Suzanne Collins','original_title': '
   The Hunger Games' },
 {
'best_book_id': 2, 'authors': 'J.K. Rowling, Mary GrandPré', 'original_
      title' : ' Harry Potter and the Philosophers Stone'
       },

'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 csvfile :
      d = csv.reader (csvfile , delimiter = '|')
       for r in d :
           print (',' .join (r))
```