Program NO-1

Aim: write a python program to find the square of the numbers entered by the user?

program

```
num = float (input ( " please enter your value. "))
square = num * num
print (" the square of the given value is {0} = {1} ".
        format (num, square))
```

Result :

The program has been executed and output is verified.

output

please enter any value 5

The square of the given value is 5.0 = 25.0

Program NO: 2

AIM: write a program to return area of a circle using a function?

Program

```
def find Area (r);
    PI = 3.142
    return PI * (r*r)
num = float (input ("enter a value :"))
print ("Area is %.6f" % find Area (num));
```

Result :

The program has been executed and output is verified.

Output

enter a value : 5

Area is 78.5000

Program NO - 3

Aim : write a python program to find the biggest of the 3 numbers entered by the user?

program

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

Result :

The program has been executed and output is verified.

Output

Enter first number : 10

Enter second number : 15

Enter third number : 25

The largest number is 25.0

Program NO: 4

Aim: List comprehensions . square of N numbers

```
for x in range(100):
    print(x**2)
```

Result:

The program has been executed and output is verified.

output

0
1
4
16
25
36
49
64
81
100
121
144
...

Program No: 5

Aim: From a list of vowels selected from a given word?

Program

```
def check_vow(string, vowels):
    final = [each for each in string if each in vowels]
    print(len(final))
    print(final)

string = "jeenamathew thekkidagil"
vowels = "AaEeIiOoUu".
check_vow(string, vowels):
```

Result: The program has been executed output is verified.

Output

9

['a', 'à', 'i', 'ì', 'e', 'è', 'e', 'é', 'á']

program NO-6

Aim: count the occurance of each word in a line of text ?

program

```
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("hello Seena mathew how are you"))
```

Result:

The program has been executed output is verified.

output

{'hellow': 1, 'geeks': 1, 'noathew': 1, 'are': 2, 'how': 1, 'are': 2, 'fine': 1}

program NO : 7

Aim : store a list of first numbers . count the occurence of 'a' with in the list ?

program

```
test_str = " Jeena mathew thekkidayil house "
count = 0 :
    for i in test_str :
        if i == "a":
            count = count + 1
    print ( " count of a in jeena mathew thekkidayil house "
            + str (count))
```

Result :

The program has been executed & output is verified.

output

count of a in Jeena Mathew Thekkidayil house : 3

Program No -8

Aim: Enter 2 lists of integers. check a) whether list are of same length b) whether list sums to same value c) whether any value occur in both?

program

```
def lists():
    List 1 = [ ]
    List 2 = [ ]
    List 3 = [ ]
    n1 = int (input ("total number of elements in list 1:"))
    for i in range (n1):
        val = int (input ("Enter a number:"))
        List 1 . append (val)
    n2 = int (input ("total number of element is list 2:"))
    for i in range (n2):
        val = int (input ("Enter a number:"))
        List 2 . append (val)
    if (n1 == n2):
        print ("List are of same length")
    else:
        print ("List are not same length:")
    if (sum (list 1) == sum (list 2)):
        print ("sum value is same")
    else:
        print ("sum value is not same")
        List 3 = [each for each in list 1 if each in
        list 2]
        print ("value in the both lists are :", list 3)
            List 3
```

Result : The program has been executed output is verified.

output

total number of elements in list 1:4

enter a number : 5

enter a number : 6

enter a number : 2

enter a number : 4

total number of elements in the list 2: 5

enter a number : 2

enter a number : 5

enter a number : 7

enter a number : 8

enter a number : 9

List are not same length

Same values is not same

Values in the both lists are : [5, 2]

Program No: 9

Aim: Get a string from an input string where all occurrence of first character replace with '$', except first character?

Program

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

Result:

The program has been executed output is verified.

output

2 4 $ 8 t a # t

Program NO : 10

Aim: create a string from given string where the first and last characters exchanged.

program

```
str = input ("Enter a string :")
new_str = str [-1 :] + str [1 : -1] + str [: 1]
print (new_str)
```

Result:

The program has been executed and output is verified.

output

enter a string : jeeraj

ajeer

program NO : 11

Aim : Accept the radious from user and find area of circle

program

```
import math:
    r = float (input ("enter the radius of the circle:"))
    area = math.pi *r* r
    print ("%.2f" %area)
```

Result :

The program has been executed and output verified.

Output

Enter the radious of circle 5

78.54

program No : 12

Aim : Accept an integer n and compare n+nn+nnn

program

```
n = int(input("Enter a number n: "))
    temp = str(n)
    t1 = temp + temp
    t2 = temp + temp + temp
    comp = n + int(t1) + int(t2)
print("the value is", comp)
```

Result :

The program has been executed and output is verified.

output

Enter a number n: 5

the value is : 615

program NO : 13

Aim : sort dictionary in asending and desending order

program

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print ("dictionary :", d)
S= sorted (d.items (), key = operator. item getter(1))
print ("asending order :", S)
S1 = dict (sorted (d item (), key = operator. Itemgetter
(1), revese = True))
print ("decending order :, 'S1)
```

Result :

The program has been executed and output is verified.

output

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

ascending order [(0,0), (2,1), (1,2), (4,3), (3,4)]

descending order : {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}

Program No - 14

Aim : merge two dictionaries :

Program

```
x = {'a' : 1, 'b' : 2}
y = {'b' : 10, 'c' : 11}
z = x.update(y)
print(z)
print(x)
```

Result

The program has been executed and output verified

Output

None

{'a': 1, 'b': 10, 'c': 11}

program No : 15

Aim : Find gcd of two numbers

Program

```
num1 = int (input ("Enter 1st number :"))
num2 = int (input ("Enter 2nd number :"))
i = 1
while (i <= num1 and i <= num2):
    if (num1 % i == 0 and num 2 % i == 0):
        gcd = i
        i = i+1
        print ("GCD is", gcd).
```

Result :

The program has been executed and output is verified.

output

Enter 1st number : 80

Enter 2nd number : 20

GCD is 20

Program NO : 16

Aim : Program to find the factorial of a number

Program

```
num = int(input("Enter a number:"))
factorial = 1
if num<0:
    print("sorry, factorial dot found")
elif num == 0:
        print("The factorial of", num" is", factorial
```

Result :

The program has been executed and output verified.

Output

Enter a number : 9

The factorial of 9 is 362380

program No: 17

Aim: Generate fibanacci series of N terms program:

program

```
nterms = int (input ("How many terms?"))
    n1, n2 = 0, 1
    count = 0
if nterms <= 0:
    print ("please enter positive integer")
    elif nterms == 1:
print ("fibanacci sequence upto", nterms ":")
print (n1)
else:
print ("fibanacci sequence:")
while count < nterms:
print (n1)
nth = n1 + n2
    n1 = n2
    n2 = nth
    count += 1
```

Result:

The program has been executed and output is verified.

Output

How many terms ? 4

fibonacci Sequence :

0
1
1
2

Program NO : 18

Aim: Find the sum of all items in a list.

```
lst = [ ]
num = int (input ("How many numbers: "))
    for n in range (num):
        numbers = int (input ("Enter numbers "))
        lst . append (numbers)
print ("sum of elements in given list is : ", sum
                                        (lst))
```

Result:

The program has been executed and output is verified.

Output

How many numbers : 7

Enter number 2

Enter number 3

Enter number 4

Enter number 5

Enter number 6

Enter number 7

Enter number 8

Sum of elements in given list is : 35

program NO : 19

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

program

```
def prefect square (l, b):
    for i in range (l, ut+1):
        if (i**(.5) == int(i**(.5))):
            print (i, end = " ")

l = 2
u = 24
prefect squre (l, r)
```

Result:

The program has been executed and output is verified.

output

4, 9, 16

program NO : 20

Aim: Display the given pyramid with step number accep
from user eg: n = 4

1
2 4
3 6 9
4 8 12 16

program

```
def pyr():
    n= int(input("Enter the number:"))
    i=1
    for i in range(1,n+1):
        j=1
        for j in range(1,i+1):
            temp = i*j
            print(temp, end = " ")
        print(" ")
```

Result:

The program has been executed and output is verified

Output

Enter the number : 4

1

2 4

3 6 9

4 8 12 16

program NO: 21

Aim: count the number of characters (character frequency) in a string.

program

```
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('google.com'))
```

Result:

The program has been executed and output is verified.

output

{'g': 2, 'ò': 3, 'i': 1, 'ë': 1, '.': 1, ',': 1

'm': 1}

program NO: 22

Aim: Add 'ing' at the end of a given string. If it already ends with 'ing' then add 'ly'.

```
def add_string(str1):
    length = len(str1)
    if length > 2:
        if str1[-3] == 'ing':
            str1 += 'ly'
        else:
            str1 += 'ing'
    return str1
print(add_string('jce'))
print(add_string('jeena'))
print(add_string('string'))
```

Result:

The program has been executed and output is verified.

output

seeing

seeing

stringly.

program NO: 23

Aim: Accept a list of coords an retarn length of logest coord.

Aprogram

```
a=[ ]
n = int (input ("Enter the number of elements in list:"))
for oc in range (0,n):
    element = input ("Enter element"+ str (oc+1) +":")
    a . append (element)
max1 = len (a[0])
temp = a[0]
for i in a:
    if (len (i)>max1):
        max1 =len(i)
        temp = i
print ("the coord with the lagest length is")
print (temp
```

Result:

The program has been exccecated and oatput is verified.

Output

Enter the number of elements in list : 3

Enter element 1 : banana

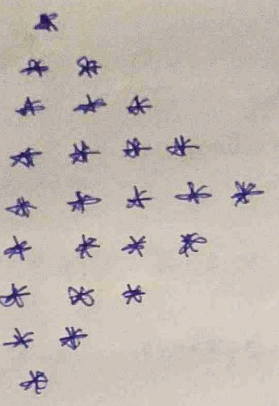
Enter element 2 : Apple

Enter element 3 : watermelon

The word with the logest length is : watermelon

Program No - 24

Aim: Construct following using nested loop:



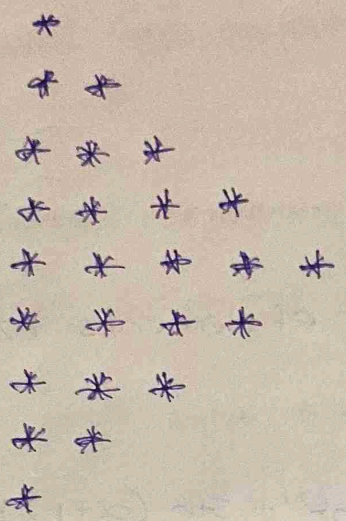
program

```
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 ('')
```

Result:

The program has been executed, output is verified.

Output



program NO : 25

Aim : Create all factors of numbers of a number?

program

```
def print_factors (x):
print ("The factor of ", x, are :")
  for i in  range (1, x+1):
    if x% i == 0:
      print (i)

  num = 200

print_factors (num)
```

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

Output

The factors of 320 are

1
2
4
5
8
10
16
20
32
40
64
80
160
320

program NO: 26

Aim: cosite lambda function to find area of square rectangle and triangle?

program

```
import math
t = peri = lambda p,q,s : p+q+s
s_ area = lambda . len . br : len*bt
c-peri = lambda rad : 2* math. pi * rad.
c_area = lambda rad : math pi * rad *rad
    print ("perimeter of triangle (10,20,15) is:"
        t_ peri (10,25,15))
    print ("Area of rectangle (30,20) is : "r_ area
    (25,30))
```

Result:

The program has been excecated and output is verified.

Output

perimeter of triangle (10, 20, 15) is : 50

Area of rectangle (30, 20) is : 750.

program NO : 27

Aim : from a list of integers, creates a list removing even numbers

program

```
list = [11, 22, 33, 44, 55, 66]
print ("original list")
print (list)
for i in list:
    if (i%.2 ==0):
        list.remove (i)
        print ("list after removing an even
                number : ")
        Print (list)
```

Result :

The program has been executed and output verified

Output

original list

[11, 22, 33, 44, 55, 66]

List after removing an even numbers:

[11, 33, 44, 55, 66]

List after removing an even numbers:

[11, 33, 55, 66]

List after removing an even numbers:

[11, 33, 55]

Program NO : 28

Aim : Displaf future leap geares from current gear to a final geare entered by user?

program

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

Resalt:

The program bas been executed and oatpat is verified.

Output

Enter final year : 2080

leap years:

2024
2028.

Program NO: 29

Aim: generate positive list of numbers from a given list of integers.

Program

```
list1 = [56, 25, 85, -44, -620, -95, 55-50, 66]
post = 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 output is verified.

Output

original list : [56, 25, 85, -44, -715, 5, 66]

positive integers list : [56, 25, 85, 5, 66]

program NO: 30

Aim: find biggest of 3 numbers entered.

program

```
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')
```

Result:

The program has been executed and output is verified.

Output

Enter 1st no: 17

Enter 2nd no: 15

Enter 3rd no: 20

20 is the biggest number.

program No: 31

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

program

```
colors = (input ("Enter colors seperated by commas:"))
          .split (',')
print ("First color:", color [0])
print ('Last color:', colors [len (colors)-1])
```

Result:

The program has been exceceted and output is verified.

Output

Enter colors separated by commas : red, yellow, green

first color : red

last color : green

program NO : 32

Aim: printout all colors from color-list1 not contained in color-list2

program

```
colors 1 =set ([input ('Enter colors separated by commas
        ')).split(',')]
colors 2 = set ([input ('Enter colors separated by commas:'))
        .split(',')]
print ('colors in color_list1 not contained in color_list2 are:
    ', list (colors1.difference (colors 2)))
```

Result:

The program has been excecuted and output is verified.

Output

Enter colors separated by commas : black, green, yellow

Enter colors seperated by commas: blue, red, white

color in color-list not contained in color-list 2 are :

['black', 'green', 'yellow']

program No: 33

Aim: create a single separated with space from two strings by swapping the string.

program

```
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, output is verified

Output

Enter a string : Jeena mathew

Enter another string : rbekkidagil

Jeena mathew jbekkidagil

program No: 34

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

program

circle.py / find area.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 length of the cuboid :'))
b = float (input ('Enter breadth of the cuboid :'))
b = float (input ('Enter bright of the cuboid:'))
    cuboid.area (l,b,b)
r = float (input ('Enter the radius of the sphere:'))
   sphere.are(r)
```

findperimeter.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 length 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:"))
b = float (input ("Enter height of the cuboid:"))
cuboid . perimeter (l, b, b)
r = float (input ("Enter the radius of the spere:"))
spere . perimeter (r)

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 ("perimete of rectangle with sides", a, "and" b, b,
        "%.2f" %(2*(a+b)), "units")

3D - graphics
cuboid . py
def are (l,b,b):
```

```
print ('Total surface area of cubord with dimensions',
    l,',',b,',',b,'is','%.2f'%(2*((l*b)+(b*b)+(l*b))),'sq.
    units')
def perimeter (l,b,b):
print ("perimeter of cubord with dimensions',l,',',b,',',b,'is
    ','%.2f'%(4*(l+b+b)),'units') sphere
    sphere.pg
    def area (r):
    print ('Area of sphere with radius',r,'is',':','%.2f'%(4*
    (3.14*r*r)),'sq.units')
    def perimeter (r):
    print ('perimeter of (greate circle of) sphere with radius',
    r,'is','%.2f'%(2*3.14*r),'units')
```

Result:

The program has been executed and output is verified

Output

perimeter of a circle with radius 10 is 62.831853071795865

Area of a circle with radius 10 is : 314.159265358979793

Area of a Rectangle with length and coridth 10 is : 100

peremeter of a Rectangle with length and coidth 10 is : 40

Area of a cuboid with length, coridth, beight 10 is : 600

Perimeter of a cuboid with length, coridth, beight 10 is : 120

Area of a sphere with radius 10 is : 1256.637061435917 3

perimeter of a sphere with radius 10 is : 62.831853071795865

Program NO : 35

Aim: create Rectangle class with attributes length and breadth and methods to find area and perimeter, compare two rectangle objects by their are

Program

```
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 comp(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(7,5)
r2 = Rectangle(8,4)
r1.comp(r2)
```

Result: The program has been executed and output is verified

oatpat

Rectangle with length = 20 and breadth = 4 has the greater area.

Program No: 36

Aim: create a Bank account with members account no: name, type of account and balance, write constructor and methods to deposit at the bank and withdraw an amount from the bank.

program

```
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, 'withdrawal! current balance
            is: Rs.', self.bal)

        else:
            print("insufficient balance to take this
            transaction!")
a = int(input('Enter account numbe:'))
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 is verified.

Output

Enter account number : 1526306 9 98 756

Enter name of the Account holder : Jeena mathew

Enter account type : joined ac

Enter your balance : 630

Enter amount to deposit : 800

Rs : 800.0 deposited ! current balance is : Rs. 1430.0

Enter amount to withdraw : 500

Rs. 500.0 withdrawn ! current balance is : Rs. 930.0

Program NO : 37

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

Program

```
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:'))
b = 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 output is verified

Output

Enter length of 1st rectangle : 5

Enter cordth of 1st rectangle : 3

Enter length of 2nd rectangle : 9

Enter cordth of 2nd rectangle : 6

Rectangle with length = 5.0 and cordth = 3.0 has the lesser area.

program NO : 38

Aim : create a class time with private attributes hour, minute and second, overload '+' operator to find sum of 2 time

program

```
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(2,25,45)
T2 = Time(16,50,45)
T1+T2
```

Result : the program has been executed and output is verified

output

Time[hh:mm:ss] 7:16:30

program No: 39

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. program to display information about a python book, use base class constructor invocation and method overriding.

program

```
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 (423.50, 302, 'An Idealist View of Life', 'Dr.
    S. Radhakrishnan', 'Andesite press')
P1.show()
```

Result :

The program has been executed and output verified.

Output

Book title : Making of new india

Author : Dr. Bibek Debroy

Publisher : ABC Books

Price : Rs. 552.9

No of pages : 260

program NO: 40

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

program

```
def file_read(fname):
    with open(fname) as f:
    # x contents_List is the list that contain the read lines.
    c = f.readlines()
    print(c)

file_read("file11.txt")
```

Result: the program has been executed output is verified

Output

['oop concept is a programming methodology of a object oriented']

program No: 11

Aim: python program to copy odd lines of one file to other?

program

```
a = open ("file12.txt", 'r')
b = open ('bellow2.txt', 'w')
for i in range (0, len(c)):
    if (i%2 !=0)
        b.write (c[i])
    else :
        pass
b.close()
b = open ("file12.txt", 'r')
d = b.read()
print (d)
a.close()
b.close()
```

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

Output

1. I'm not going to go deep into history matters. This introduction was just to make it a bit fun, and show how the phrases given as an indication of the importance of pictures.

2. It is thus no doubt that picture play an important part to our communication.

Program NO : 42

Aim: write a program to read each row from a given csv files and print a list of strings?

program

```
import csv
with open('jj.csv', newline='') as csvfile:
    d= csv.reader(csvfile, delimeter=',', quotechar='|')
    for r in d
        print(',', join(r)
```

Result:

The program has been executed and output verified.

Output

player_name, rade_rating

magnus, carlsen, 2790

fabiano, caruana, 2929

ding, liren, 2801

Program No : 43

Aim: write a pyfiles pgro to read specific columns of a given csv files and print the colums

program

```
import csv
with open ('col.csv', newline '') as csvfile
    d = csv, DictReader (csv file)
    print ("author original-title")
    for r in d:
      print (r ['authors'], r ['original_title'])
```

Result

The program has been executed and output is verified.

oatpat

oathers orginal_title

Suzanne collins . The Hanger games

J.K Rowling, marg Grandprt Harry. potter and pbiiosopbers store

stepbenie megar Tcollight.

Program No :44:

Aim : write a pgm to write a python directory to a csv file. after writing the csv file read the csv file and display the content

Program

```
import csv
field_names = ['best_books_id', 'authors', 'original_title']
books = [ { 'best_books_id': 1, 'authors': 'suzanne collins', 'original_
    title': 'the hunger games' }, { 'best_book_id : 2, 'authors'
    : 'JK, Rowling, mary GrandPre', 'original_title': '
    Harry potter and the philosophers stone' },
    { 'best_book_id': 3, 'authors': 'stephenie mayer',
      'original_title': 'Twilight' }, ]

with open ('c1.csv', 'w') as csvfile:
    writer = csv.DictWriter (csvfile, fieldnames = field_
    names)
    writer.writeheader()
    writer.writerows (books)

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

Result: the program has been executed and output verified.

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, Stephenie Meyer, Twilight