SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-23

20 MCA 132-PROGRAMMING LAB LABORATORY RECORD

Submitted by

ATHIRA M

REG NO: SNG21MCA-2010

in partial fulfilment for the award of degree in

MASTER OF COMPUTER APPLICATION

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

DEPARTMENT OF COMPUTER APPLICATIONS

20 MCA 132-PROGRAMMING LAB LABORATORY RECORD

Certified that this is a Bonafide record of practical work done by ***ATHIRA M*** *to the APJ Abdul Kalam Technological University in partial fulfilment of the requirement for the award of the degree in the Master of Computer Applications of Sree Narayana Gurukulam College of Engineering done during the Academic year 2021-23.*

Kadayiruppu

Course Instructor
Date: ………………..

Head of the Department

Prof. Dr. SANDHYA R

Submitted for University Practical Examination

Reg No:

SNG21MCA-2010 On……………….

EXTERNAL EXAMINER
INTERNAL EXAMINER

SL No:	Date	Name of the program	Page No	Remark
		Co1		
1	24/11/21	Familiarizing Text Editor ,IDE,Code,Analysis Tools etc		
2	24/11/21	Leap Year		
3	24/11/21	List Comprehensions		
4	24/11/21	Occurrences of each word		
5	24/11/21	Prompt the use for a list integers		
6	24/11/21	Store a list of first names		
7	24/11/21	Checking list are of same length ,sums, to same value occur in both		
8	24/11/21	Get a string from an input string and replacing a characters		
9	24/11/21	Create a string from given string where first and last characters exchanged		
10	24/11/21	Accept the radius from user and find area of circle		

\

11	29/11/21	Find biggest of 3 numbers entered		
12	29/11/21	Accept a file name from user and print extension of that		
13	29/11/21	Create a list of colors, Display first and last colors		
14	29/11/21	Accept an integer n and compute n+nn+nnn		
15	29/11/21	Print out all colours from color-list1 not contained in color-list2		
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1		
17	29/11/21	Sort dictionary in ascending and descending order		
18	29/11/21	Merge two dictionaries		
19	29/11/21	Find gcd of 2 numbers		
20	29/11/21	From a list of integers , create a list removing even numbers		
		C02		
1	01/12/21	Program to find the factorial of a number		
2	01/12/21	Generate Fibonacci series of N terms		
3	01/12/21	Find the sum of all items in a list		
4	01/12/21	Generate a list of all four digit numbers in a given range with all their digits even and the number is a perfect square		
5	01/12/21	Display the given pyramid with step number accepted from user		
6	01/12/21	Count the number of characters [character frequency]in a string		
7	08/12/21	Add ‘ing’ at the end of given string .If it already ends with ‘ing’,then add ‘ly’		
8	08/12/21	Accept a list of words and return length of longest word		
9	08/12/21	Construct pattern using nested loop		

10	08/12/21	Generate all factors of a number .def print -factors (x):		
11	08/12/21	Write lambda functions to find area of square , rectangle and triangle		

	Co3			
1	15/12/21	Work with built -in packages		
2	15/12/21	Creation of packages		
	C04			
1	09/01/22	Compare two rectangle objects by their area		
2	09/01/22	Create a Bank account with members account number, name, type of account and balance		
3	09/01/22	Overload ‘<’operator to compare the area if 2 rectangles		
4	09/01/22	Overload ‘+’operator o find sum of 2 time		
5	09/01/22	Use base class constructor invocation and method overriding		
	Co5			
1	30/01/22	Write a python program to real a file line by line and store it into a list		
2	30/01/22	Python program to copy odd lines of one file to other		
3	30/01/22	Write a python program to read each row from a given csv file and print a list of strings		
4	30/01/22	Write a python program to read specific columns of a given csv file		
5	30/01/22	Write a python program to write a python dictionary to csv file		

I.COURSE OUTCOME 1 (CO1)

PROGRAM NO :1

DATE :24/01/21

AIM: : Familiarizing Text Editor , IDE ,Code Analysis Tools etc //Use any IDE

Pycharm

-specialized project views allowing quick switching between files.

-facilitates web development along with Django flask and web 2py.

Eric

-build-in support for Django.

-code fielding.

-Format syntax highlighting,

Thonny

-separate windows are provided to execute function calls.

-statement stepping without break points.

PROGRAM NO :2

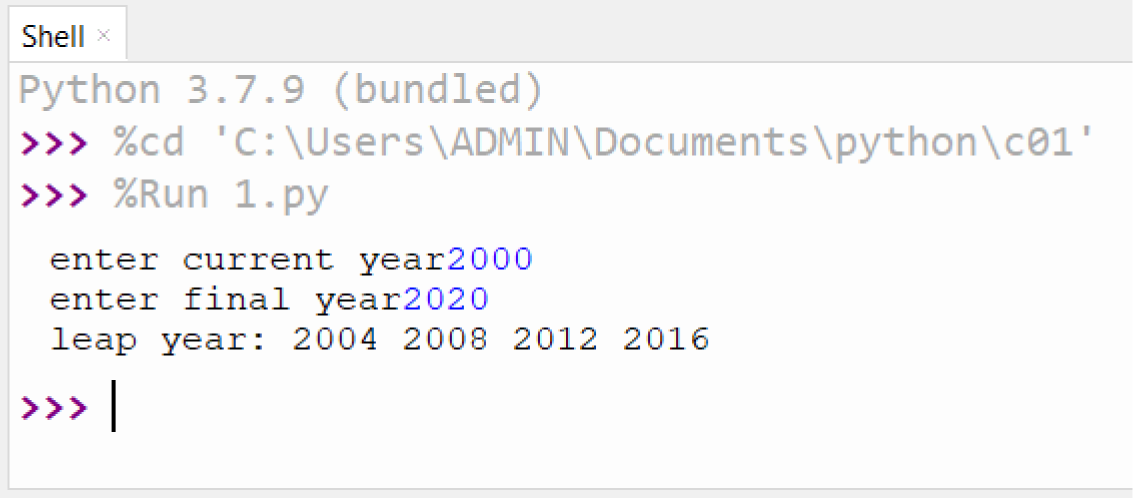
DATE :24/01/2021

AIM : Display future leap year from current year to a find year entered by user.

Program:

```
  c=int(input("enter current year"))
  f=int(input("enter final year"))
   if(c<f):
   print("leap year:",end=" ")
 for i in range(c,f):
   if(i%4==0 and i%100!=0):
  print(i,end=" ")
```

output:



```
Python 3.7.9 (bundled)
>>> %cd 'C:\Users\ADMIN\Documents\python\c01'
>>> %Run 1.py

 enter current year2000
 enter final year2020
 leap year: 2004 2008 2012 2016

>>> 
```

PROGRAM NO:3

DATE :24/01/2021

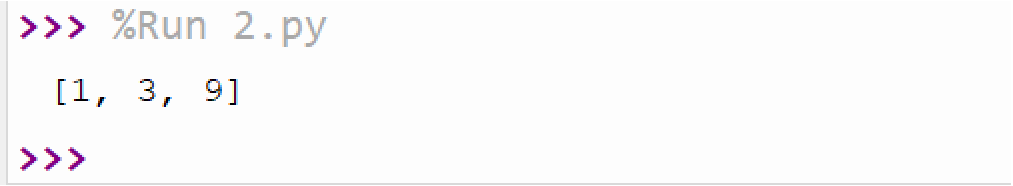
AIM : List Comprehension

a) Generate positive list of numbers from a given list of integers

Program:

```
list=[1,3,-4,-8,9]
p=[n for n in list if n>=0]
print(p)
```

output:



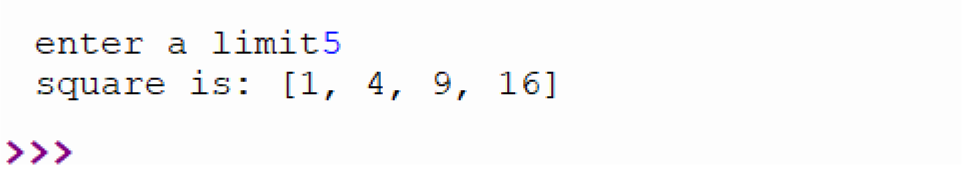
```
>>> %Run 2.py
 [1, 3, 9]
>>>
```

b) Square of N number

Program:

```
n=int(input("enter a limit"))
s=[i**2 for i in range(1,n)]
print("square is:",s,end=" ")
```

output:



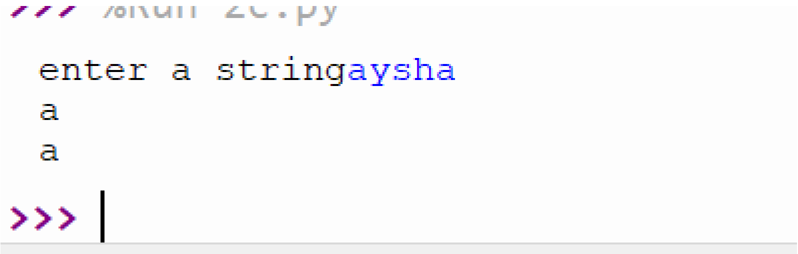
```
 enter a limit5
 square is: [1, 4, 9, 16]
>>>
```

c) Form a list of vowels selected from a given word

Program:

```
n=str(input("enter a string"))
for i in n:
  if i in"aeiouAEIOU":
    print(i)
```

output:



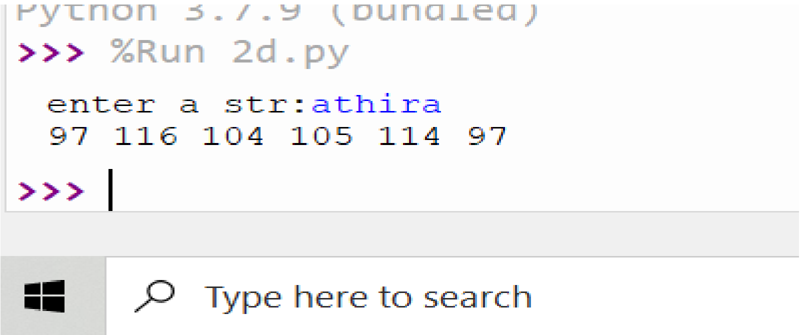
```
>>> %Run 2c.py

 enter a stringaysha
 a
 a

>>> 
```

d) **List ordinal value of each element of a word (Hint: use ord() to get ordinal values)**

Program:

```
n=str(input("enter a str"))
for i in n:
 print(ord(i),end=" ")
```

output:



```
Python 3.7.9 (bundled)
>>> %Run 2d.py

 enter a str:athira
 97 116 104 105 114 97

>>> 
```

PROGRAM NO :4

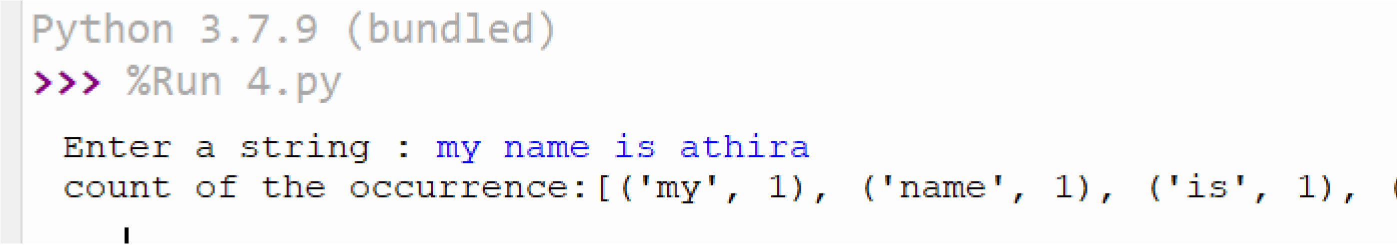
DATE :24/01/2021

Aim: Count the occurrences of each word in a line of text.

Program:

```
str1 = input("Enter a string : ")
wordlist = str1.split()
count= []
for w in wordlist: count.append(wordlist.count(w))
print("count of the occurrence:" + str(list(zip(wordlist, count))))
```

output:



```
Python 3.7.9 (bundled)
>>> %Run 4.py

 Enter a string : my name is athira
 count of the occurrence:[('my', 1), ('name', 1), ('is', 1), (
```

PROGRAM NO :5

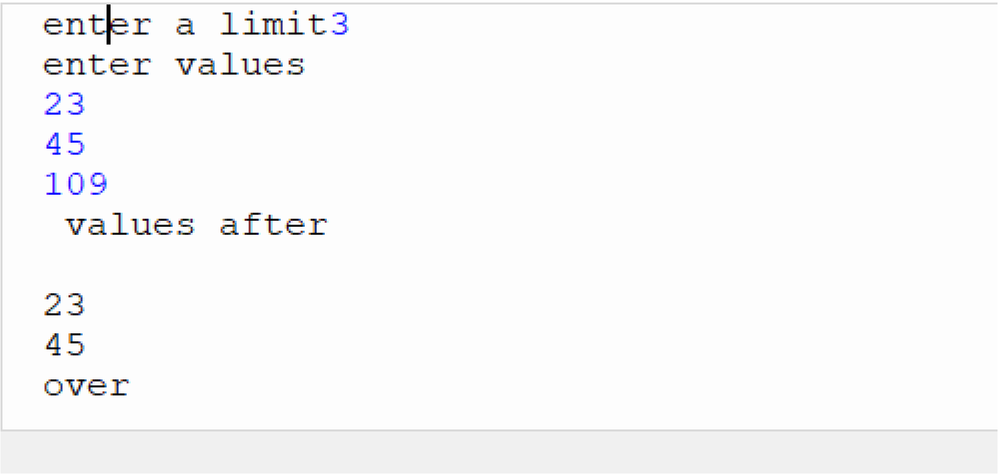
DATE :24/01/2021

Aim:Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead

Program:

```
n=[]
s=int(input("enter a limit"))
print("enter values")
for i in range(0,s):
   n.append(int(input()))
print(" values after\n")
for i in range(0,len(n)):
        if n[i]>=100:
           print("over")
        else:
            print(n[i])
```

output:



```
enter a limit3
enter values
23
45
109
 values after

23
45
over
```

PROGRAM NO:6

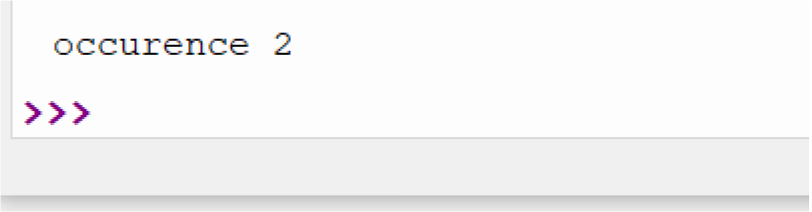
DATE :24/01/2021

Aim: Store a list of first names. Count the occurrences of ‘a’ within the list

Program:

```
list=['athira','frd','abc','a','a']
l1=list.count("a")
print("occurence",l1)
```

output:



```
 occurence 2
>>>
```

PROGRAM NO:7

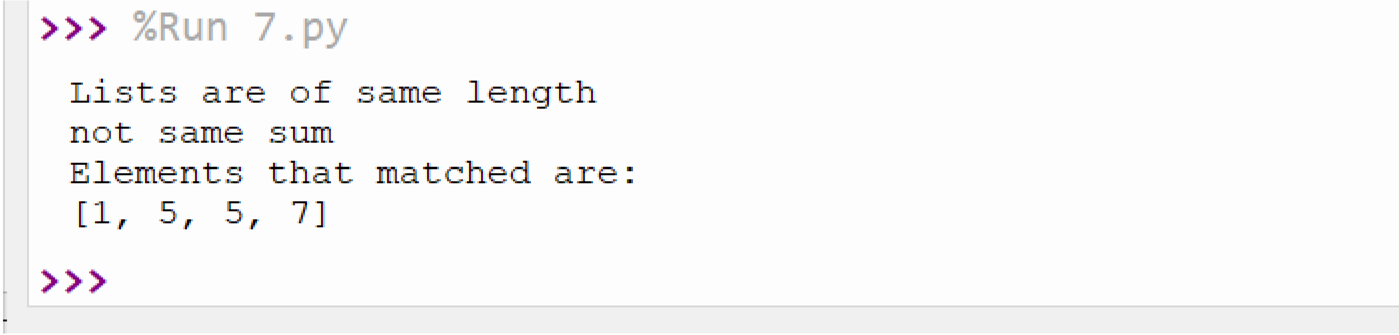
DATE :24/01/2021

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:

```
lst=[1,3,5,7,9,11,4]
lst1=[5,13,5,7,0,6,1]
s=int(0)
c=int(0)
if(len(lst)==len(lst1)):
  print("Lists are of same length")
else:
  print("Lists have different length")
for i in range(0,len(lst) and len(lst1)):
 s=s+lst[i]
 c=c+lst1[i]
if(s==c):
print("equal sum")
else:
 print("not same sum")
print("Elements that matched are:")
l=[]
for i in range(0,len(lst)):
 for j in range(0,len(lst1)):
  if lst[i]==lst1[j]:
    l.append(lst[i] and lst1[j])
  else:
   continue
print(l)
```

output:



```
>>> %Run 7.py

  Lists are of same length
  not same sum
  Elements that matched are:
  [1, 5, 5, 7]

>>>
```

PROGRAM NO:8

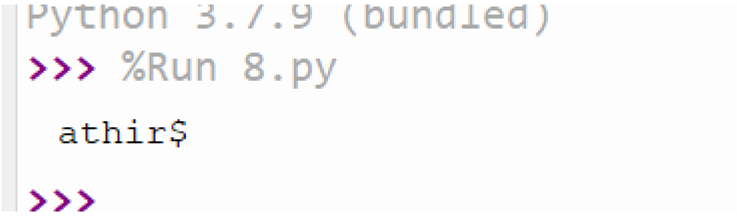
DATE :24/01/2021

Aim: a string from an input string where all occurrences of first character replaced with ‘$’, except first character. [eg: onion -> oni$n]

Program:

```
str="athira"
char=str[0]
str=str.replace(char,'$')
str=char + str[1:]
print(str)
```

output:



```
Python 3.7.9 (bundled)
>>> %Run 8.py
 athir$
>>>
```

PROGRAM NO:9

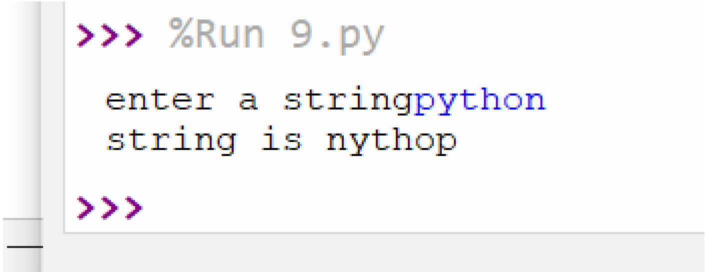
DATE :24/01/2021

Aim: Create a string from given string where first and last characters exchanged. [eg: python -> nythop]

Program:

```
str=input("enter a string")
str1=str[-1:]+str[1:-1]+str[:1]
print("string is",str1)
```

output:



```
>>> %Run 9.py

 enter a stringpython
 string is nythop

>>>
```

PROGRAM NO:10

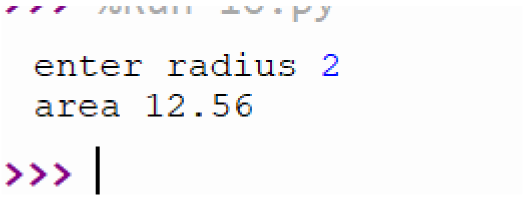
DATE :24/01/2021

Aim: Accept the radius from user and find area of circle.

Program:

```
pi=3.14
r=int(input("enter radius"))
a=pi*r*r
print("area",a)
```

output:



```
>>> %Run 10.py

 enter radius 2
 area 12.56

>>> 
```

PROGRAM NO:11

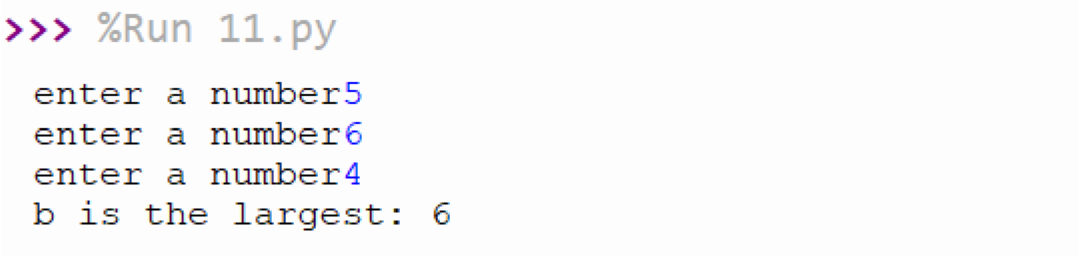
DATE :29/01/2021

Aim: Find biggest of 3 numbers entered

Program:

```
a=int(input("enter a number"))
b=int(input("enter a number"))
c=int(input("enter a number"))
if(a>b and a>c):
  print("a is the largest:",a)
elif(b>c):
    print("b is the largest:",b)
else:
    print("c is the largest:",c)
```

output:



```
>>> %Run 11.py

 enter a number5
 enter a number6
 enter a number4
 b is the largest: 6
```

PROGRAM NO:12

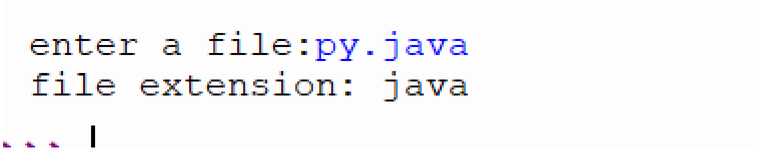
DATE :29/01/2021

Aim: Accept a file name from user and print extension of that

Program:

```
x,y=(input("enter a file:").split('.'))
print("file extension:",y)
```

output:



```
enter a file:py.java
file extension: java
```

PROGRAM NO:13

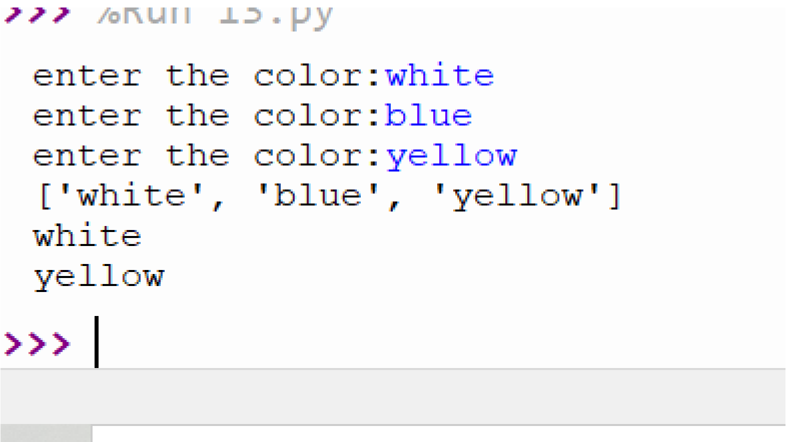
DATE :29/01/2021

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

Program:

```
a=[]
for i in range(3):
  b=input("enter the color:")
  a.append(b)
print(a)
print(a[0])
print(a[2])
```

output:



```
>>> %Run 13.py

 enter the color:white
 enter the color:blue
 enter the color:yellow
 ['white', 'blue', 'yellow']
 white
 yellow

>>> 
```

PROGRAM NO:14

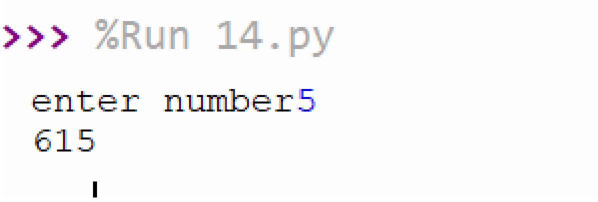
DATE :29/01/2021

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

Program:

```
n=input("enter number")
x=int(n+n+n)
y=int(n+n)
z=int(n)
print(x+y+z)
```

ouput:



```
>>> %Run 14.py

 enter number5
 615
```

PROGRAM NO:15

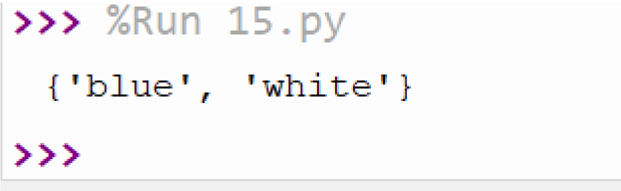
DATE :29/01/2021

Aim: Print out all colors from color-list1 not contained in color-list2.

Program:

```
list1=set(['red','white','blue'])
list2=set(['green','black','red'])
print(list1.difference(list2))
```

output:



```
>>> %Run 15.py
  {'blue', 'white'}
>>>
```

PROGRAM NO:16

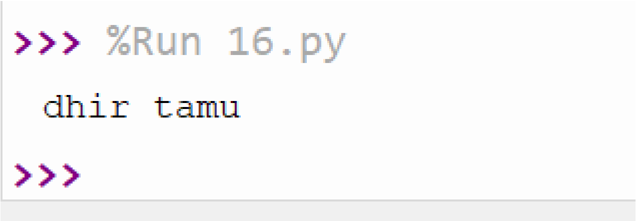
DATE :29/01/2021

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

Program:

```
a="thir"
b="damu"
p1=a[0]
p2=b[0]
c=b[0]+a[1:]+" "+ a[-4]+b[1:]
print(c)
```

output:



```
>>> %Run 16.py
  dhir tamu
>>>
```

PROGRAM NO:17

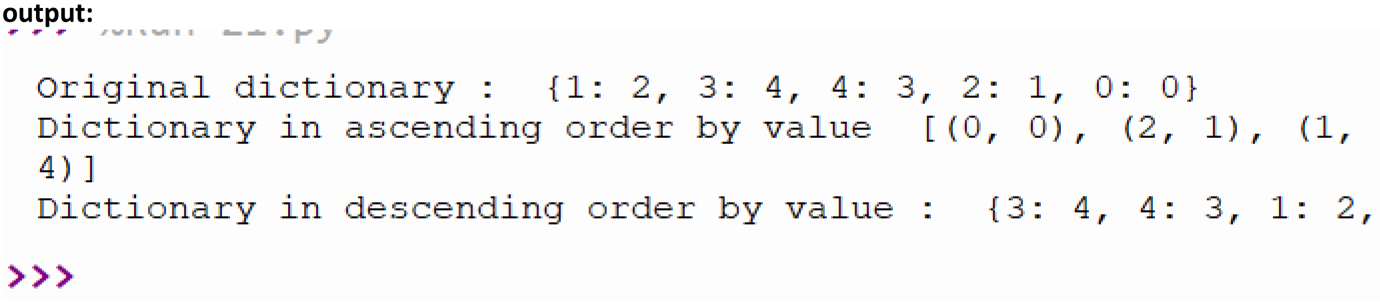
DATE :29/01/2021

Aim : Sort dictionary in ascending and descending order

Program:

```
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)
```

output:



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

>>>
```

PROGRAM NO:18

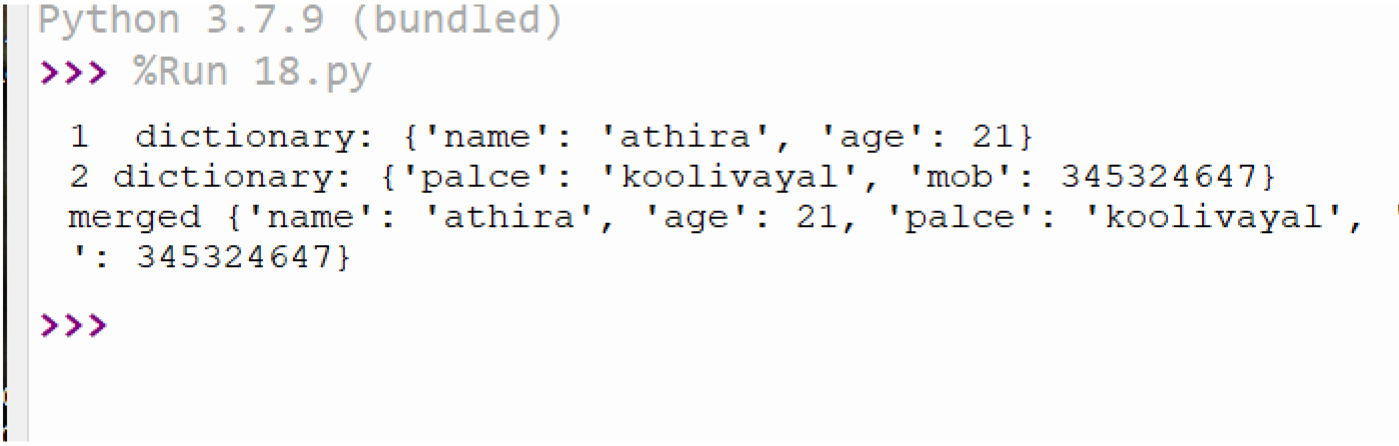
DATE :29/01/2021

Aim : Merge two dictionaries

Program:

```
a={'name':'athira','age':21}
b={'palce':'koolivayal',"mob":345324647}
print("1  dictionary:",a)
print("2 dictionary:",b)
c=a.copy()
c.update(b)
print("merged",c)
```

output:



```
Python 3.7.9 (bundled)
>>> %Run 18.py
 1  dictionary: {'name': 'athira', 'age': 21}
 2 dictionary: {'palce': 'koolivayal', 'mob': 345324647}
 merged {'name': 'athira', 'age': 21, 'palce': 'koolivayal', '
 ': 345324647}

>>>
```

PROGRAM NO:19

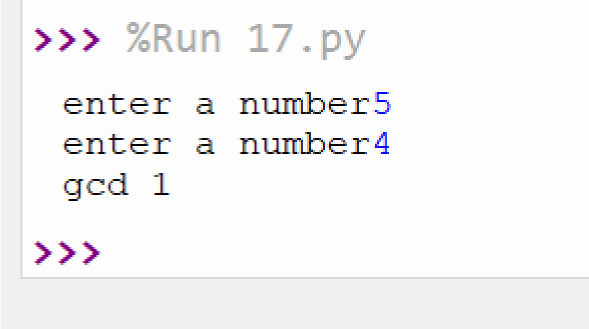
DATE :29/01/2021

Aim: Find gcd of 2 numbers.

Program:

```
a=int(input("enter a number"))
b=int(input("enter a number"))
i=1
while(i<=a and i<=b):
   if(a%i==0 and b%i==0):
      gcd=i
   i=i+1
print("gcd",gcd)
```

output:



```
>>> %Run 17.py

  enter a number5
  enter a number4
  gcd 1

>>>
```

PROGRAM NO:20

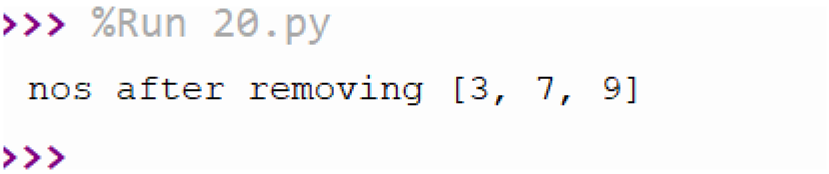
DATE :29/01/2021

Aim: From a list of integers, create a list removing even numbers.

Program:

```
n=[2,3,6,7,9,4,8]
n=[x for x in n if x%2!=0]
print("nos after removing:",n)
```

output:



```
>>> %Run 20.py
 nos after removing [3, 7, 9]
>>>
```

II.COURSE OUTCOME 2(CO2)

PROGRAM NO:1

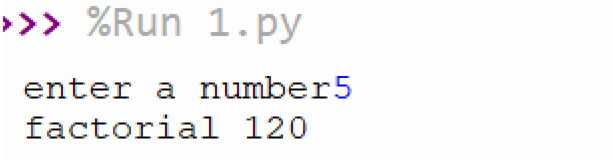
DATE :01/012/2021

Aim: Program to find the factorial of a number

Program:

```
n=int(input("enter a number"))

f=1

for i in range(1,n+1):

   f=f*i

print("factorial",f)
```

output:



```
>>> %Run 1.py

 enter a number5
 factorial 120
```

PROGRAM NO:2

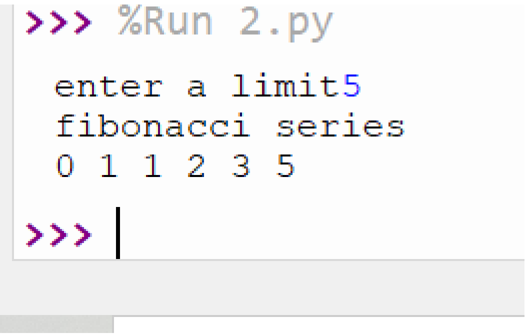
DATE :01/12/2021

Aim: Program to find the Fibonacci of a number

Program:

```
n=int(input("enter a limit"))

a=0

b=1

c=0

i=0

print("fibonacci series")

while(i<=n):

   print(c,end=" ")

   i=i+1

   a=b

   b=c

   c=a+b
```

output:



```
>>> %Run 2.py

 enter a limit5
 fibonacci series
 0 1 1 2 3 5

>>> 
```

PROGRAM NO:3

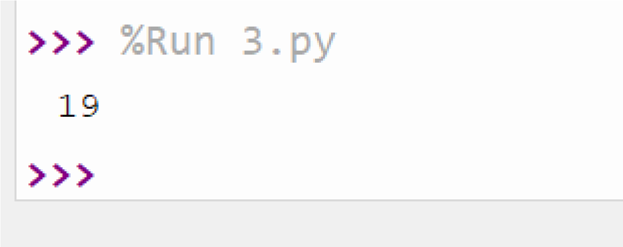
DATE :01/12/2021

Aim: Find the sum of all items in a list

Program:

```
l1=[2,4,5,6,2]

a=sum(l1)

print(a)
```

output:



```
>>> %Run 3.py
 19
>>>
```

PROGRAM NO:4

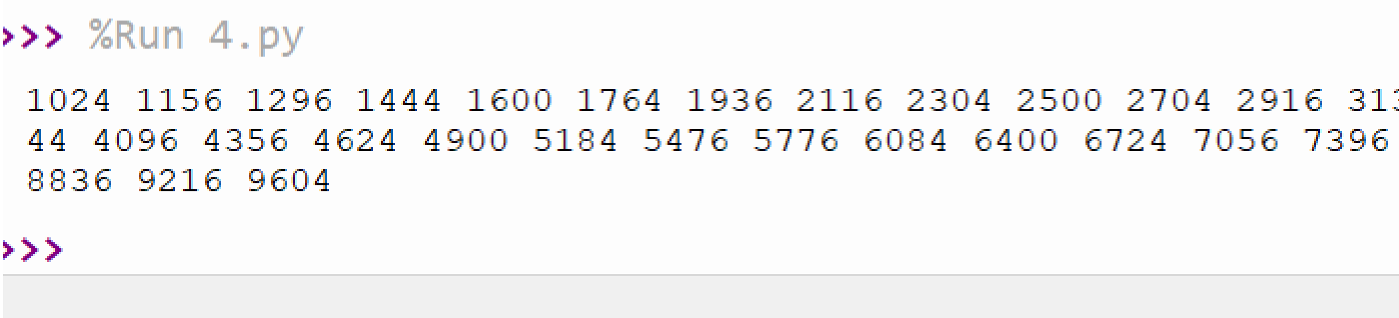
DATE :01/12/2021

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

Program:

```
from    math
import   sqrt
as  s  for  i  in
range(1000,
10000):
if s(i)==int(s(i)) and i%2==0: print(i,end=" ")
```

output:



```
>>> %Run 4.py
 1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 31
 44 4096 4356 4624 4900 5184 5476 5776 6084 6400 6724 7056 7396
 8836 9216 9604

>>>
```

PROGRAM NO:5

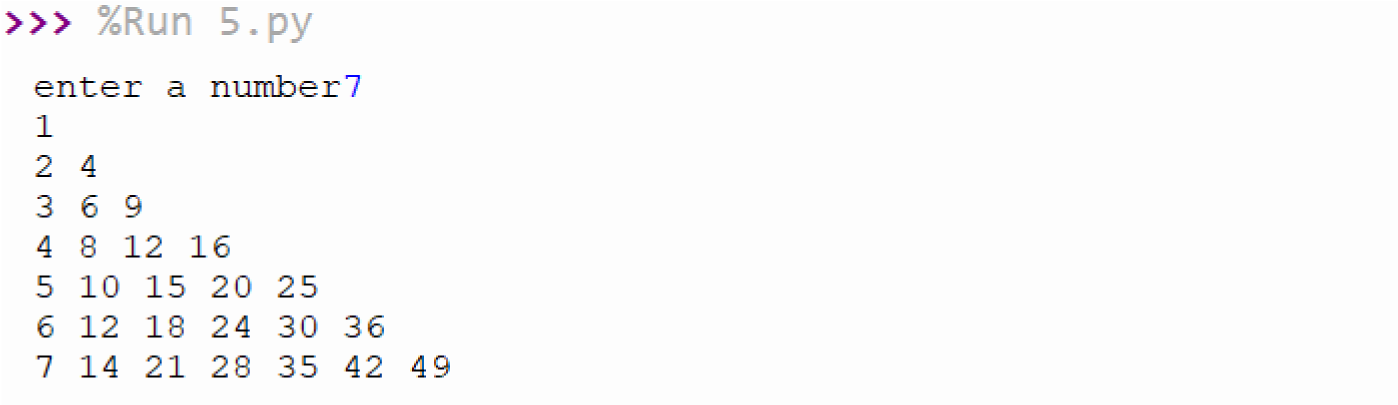
DATE :01/12/2021

Aim: Display the given pyramid with step number accepted from user. Eg: N=4 1 2 4 3 6 9 4 8 12 16

Program:

```
n=int(input("enter a number"))

for i in range(1,n+1):

    for j in range(1,i+1):

        print(i*j,end= " ")

    print()
```

output:



```
>>> %Run 5.py

 enter a number7
 1
 2 4
 3 6 9
 4 8 12 16
 5 10 15 20 25
 6 12 18 24 30 36
 7 14 21 28 35 42 49
```

PROGRAM NO:6

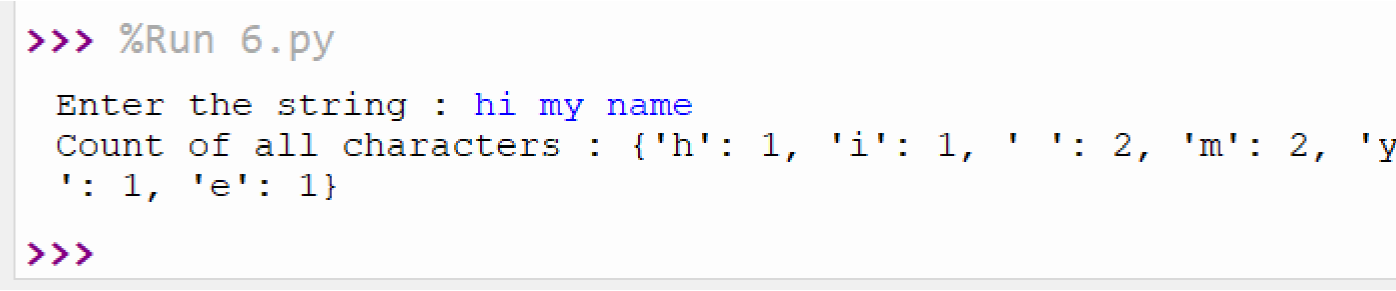
DATE :01/12/2021

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

Program:

```
test_str=str(input("Enter the string : "))

freq = {}

for i in test_str:

        if i in freq:

            freq[i] += 1

        else:

                freq[i] = 1

print ("Count of all characters : "+ str(freq))
```

output



```
>>> %Run 6.py

 Enter the string : hi my name
 Count of all characters : {'h': 1, 'i': 1, ' ': 2, 'm': 2, 'y
 ': 1, 'e': 1}

>>>
```

PROGRAM NO:7

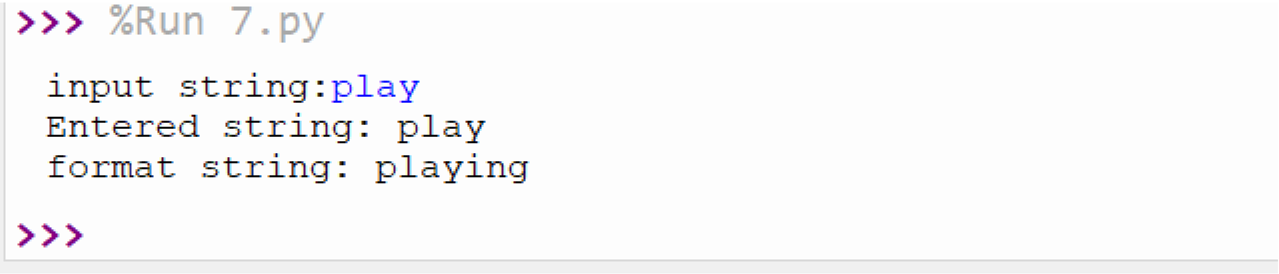
DATE :08/12/2021

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

Program:

```
str=input("input string:")

print("Entered string:",str)

if(str.endswith("ing")):

   str=str+'ly'

else:

   str=str+'ing'

 print("format string:",str)
```

output:



```
>>> %Run 7.py

 input string:play
 Entered string: play
 format string: playing

>>>
```

PROGRAM NO:8

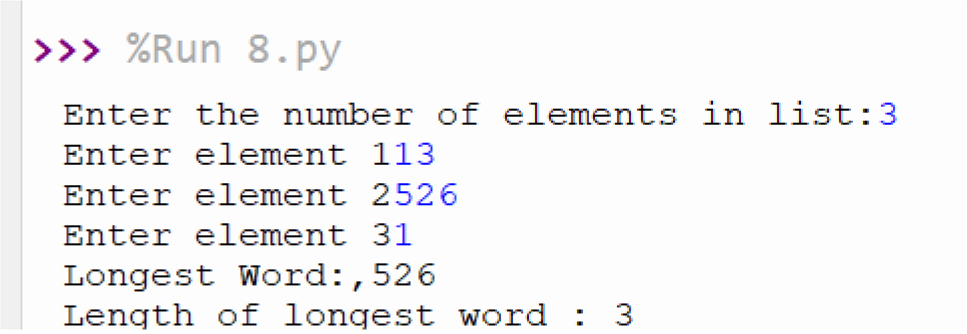
DATE :08/12/2021

Aim: Accept a list of words and return length of longest word

Program:

```
a=[]

n= int(input("Enter the number of elements in list:"))

for x in range(0,n):

   element=input("Enter element "+ str(x+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("Longest Word:",temp,sep=",")

print("Length of longest word :",max1)
```

output:



```
>>> %Run 8.py
 Enter the number of elements in list:3
 Enter element 113
 Enter element 2526
 Enter element 31
 Longest Word:,526
 Length of longest word : 3
```

PROGRAM NO:9

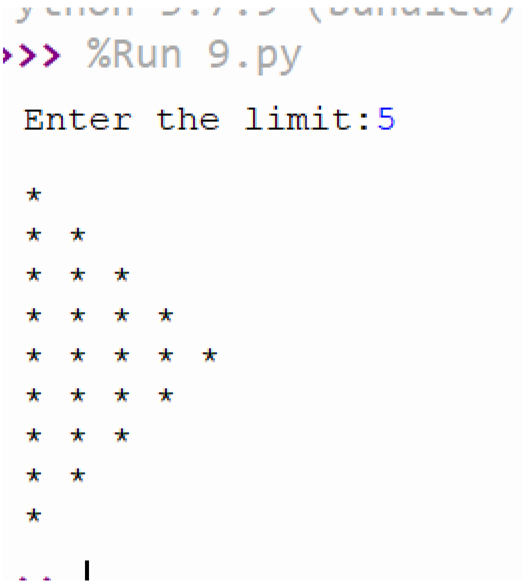
DATE :08/12/2021

Aim: Construct following pattern using nested loop *

Program:

```
n= int(input("Enter the limit:"))
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('')
```

output:



```
ython 3.7.9 (bundled)
>>> %Run 9.py
 Enter the limit:5

 *
 * *
 * * *
 * * * *
 * * * * *
 * * * *
 * * *
 * *
 *

>>> |
```

\

PROGRAM NO:10

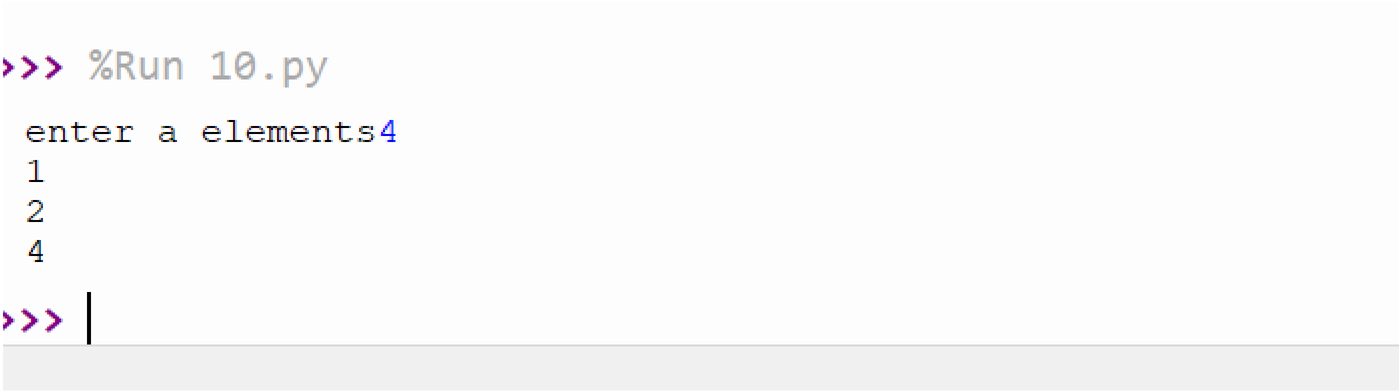
DATE :08/12/2021

Aim: Generate all factors of a number.

Program:

```
def factors(x):
 for i in range(1,x+1):
     if(x%i==0):
         print(i)
n=int(input("enter a elements"))
factors(n)
```

output:



```
>>> %Run 10.py
 enter a elements4
 1
 2
 4
>>> 
```

PROGRAM NO:11

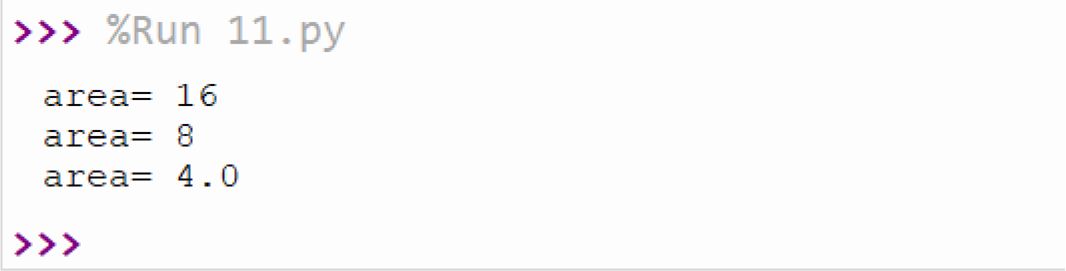
DATE :08/12/2021

Aim: Write lambda functions to find area of square, rectangle and triangle

Program:

```
import math

arsq=lambda a:a*a

arrec=lambda l,b:l*b

artr=lambda l,b:1/2*l*b

print("area=",arsq(4))

print("area=",arrec(4,2))

print("area=",artr(2,4))
```

output:



```
>>> %Run 11.py

  area= 16
  area= 8
  area= 4.0

>>>
```

III.COURSE OUTCOME 3(C03)

PROGRAM NO:1

DATE :15/12/2021

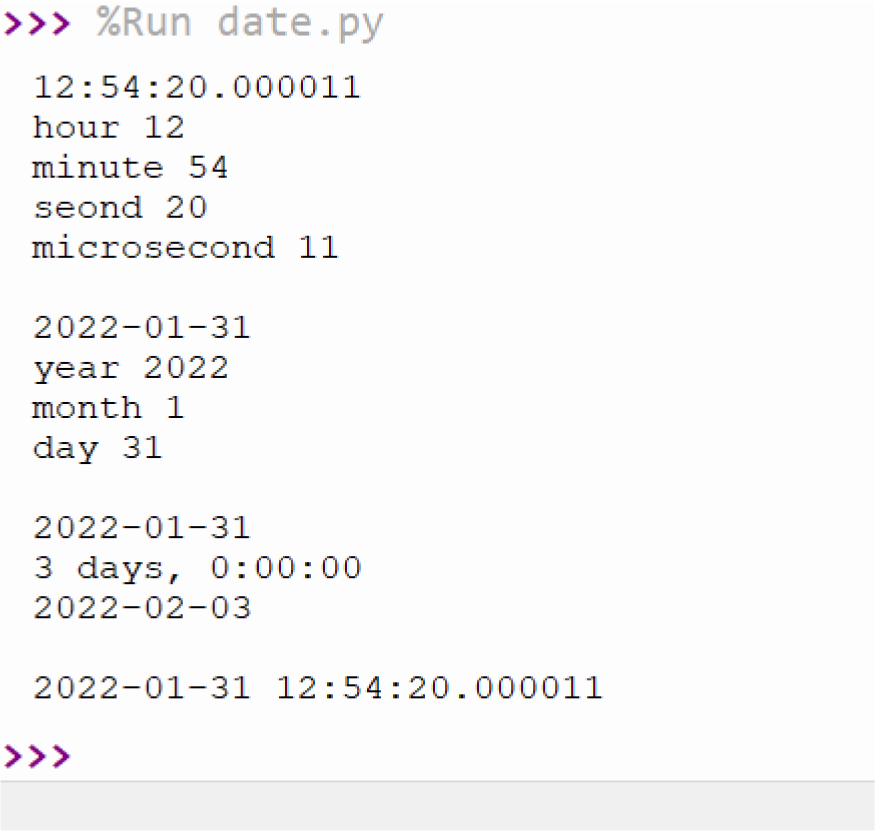
Aim: Work with built-in packages

Program:

A) Date and time

```
Importdatetime
                 t=datetime.time(12,54,20,11)
                 print(t)
                 print("hour",t.hour)
                 print("minute",t.minute)
                 print("seond",t.second)
                 print("microsecond",t.microsecond)
                 print()
                 d=datetime.date.today()
                 print(d)

                 print("year",d.year)
                 print("month",d.month)
                 print("day",d.day)
                 print()
                 d1=datetime.date.today()
                 print(d1)
                 td=datetime.timedelta(days=3)
                 print(td)
                 d2=d1+td
                 print(d2)
                 print()
                 dt=datetime.datetime.combine(d,t)
                 print(dt)
```

output:

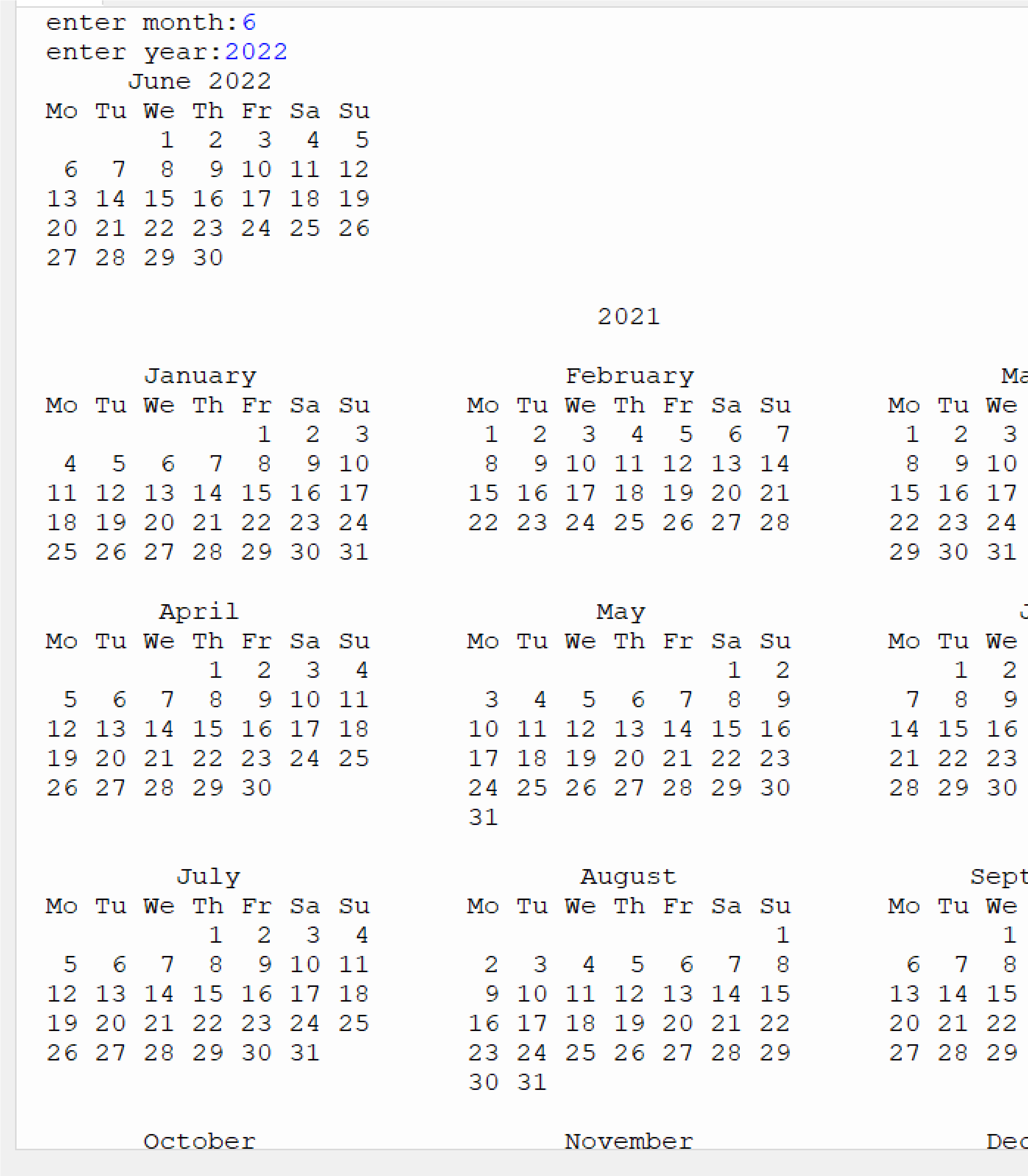


```
>>> %Run date.py
 12:54:20.000011
 hour 12
 minute 54
 seond 20
 microsecond 11

 2022-01-31
 year 2022
 month 1
 day 31

 2022-01-31
 3 days, 0:00:00
 2022-02-03

 2022-01-31 12:54:20.000011

>>>
```

B) Calender

```
import calendar
mm=int(input("enter month:"))
yy=int(input("enter year:"))
print(calendar.month(yy,mm))
print(calendar.calendar(2021))
```

output:



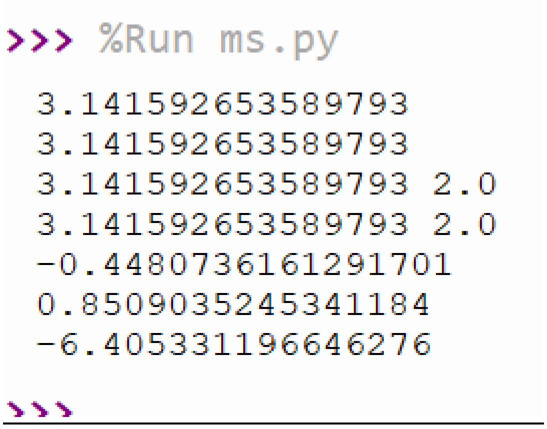
```
enter month:6
enter year:2022
     June 2022
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

                                  2021

      January                   February                   Ma
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We
             1  2  3       1  2  3  4  5  6  7       1  2  3
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       8  9 10
11 12 13 14 15 16 17      15 16 17 18 19 20 21      15 16 17
18 19 20 21 22 23 24      22 23 24 25 26 27 28      22 23 24
25 26 27 28 29 30 31                                29 30 31

       April                      May                       J
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We
          1  2  3  4                      1  2          1  2
 5  6  7  8  9 10 11       3  4  5  6  7  8  9       7  8  9
12 13 14 15 16 17 18      10 11 12 13 14 15 16      14 15 16
19 20 21 22 23 24 25      17 18 19 20 21 22 23      21 22 23
26 27 28 29 30            24 25 26 27 28 29 30      28 29 30
                          31

        July                     August                  Sept
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We
          1  2  3  4                         1             1
 5  6  7  8  9 10 11       2  3  4  5  6  7  8       6  7  8
12 13 14 15 16 17 18       9 10 11 12 13 14 15      13 14 15
19 20 21 22 23 24 25      16 17 18 19 20 21 22      20 21 22
26 27 28 29 30 31         23 24 25 26 27 28 29      27 28 29
                          30 31

      October                   November                  Dec
```

c) Math

```
import math
print(math.pi)
import math as m
print(m.pi)
from math import pi,sqrt
```

```
print(math.pi,math.sqrt(4))
from math import pi,sqrt
print(pi,sqrt(4))
print(math.cos(90))
print(math.sin(45))
print(math.tan(30))
```

output:

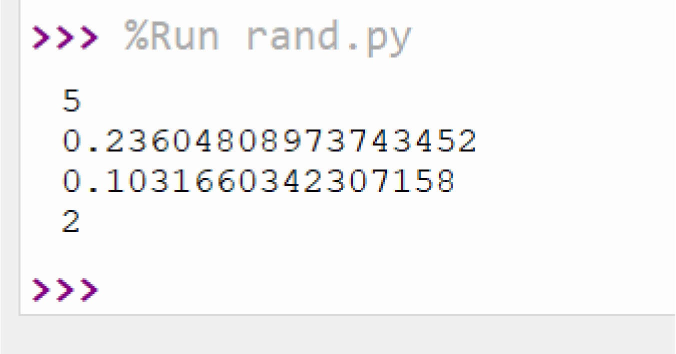


```
>>> %Run ms.py

 3.141592653589793
 3.141592653589793
 3.141592653589793 2.0
 3.141592653589793 2.0
 -0.4480736161291701
 0.8509035245341184
 -6.405331196646276

>>>
```

D)Random

```
 importrandom
                 l1 = [1, 2, 3, 4, 5, 6]
                 print(random.choice(l1))
                 random.seed(4)
                 print(random.random())
                 print(random.random())
                 r1=random.randint(1,2)
                 print(r1)
```

Output



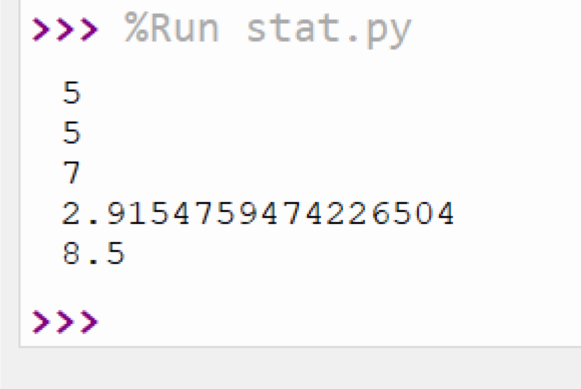
```
>>> %Run rand.py

 5
 0.2360480897374345 2
 0.10316603423071 58
 2

>>>
```

E) Statistic

```
 import
 statistics
               l1=[4,7,8,2,3,4,5,7,8,9,0,7,1]
               a=statistics.mean(l1)
               print(a)
```

```
b=statistics.median(l1)
print(b)
c=statistics.mode(l1)
print(c)
d=statistics.stdev(l1)
print(d)
e=statistics.variance(l1)
print(e)
```

Output:

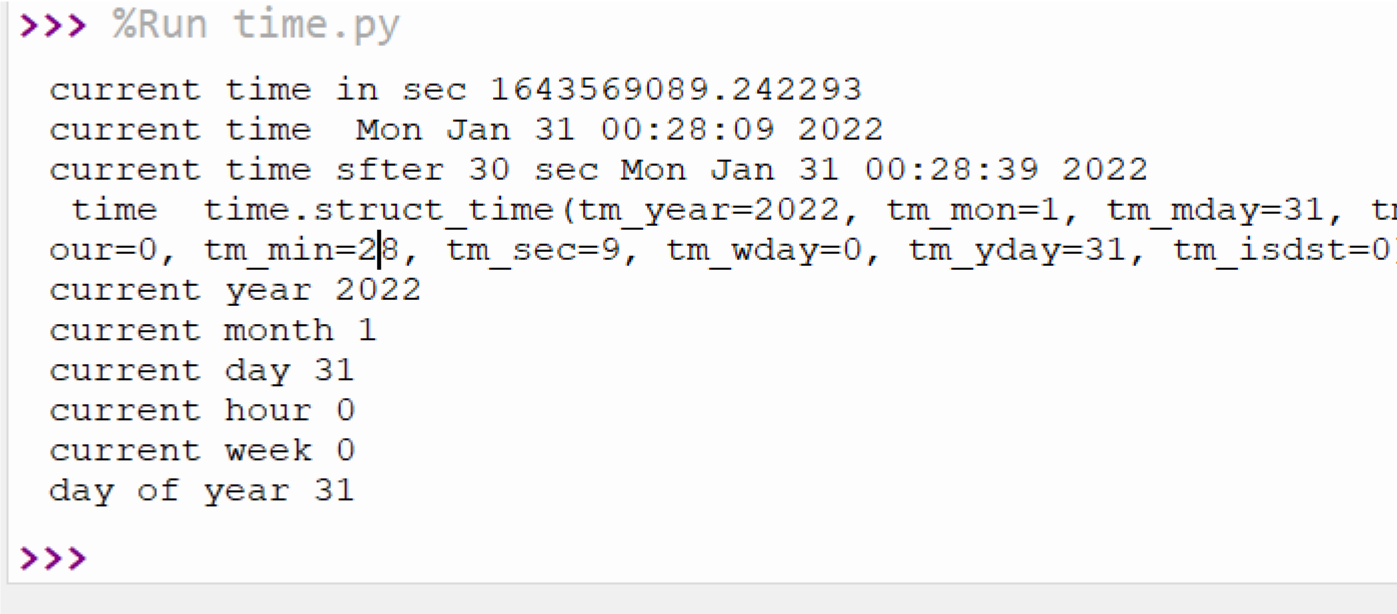


```
>>> %Run stat.py

  5
  5
  7
  2.9154759474226504
  8.5

>>>
```

F) Time

```
import
time
        print("current time in sec",time.time())
        print("current time ",time.ctime())
        print("current          time          sfter          30
        sec",time.ctime(time.time()+30))
        t=time.localtime()
        print(" time ",t)
        print("current year",t.tm_year)
        print("current month",t.tm_mon)
        print("current day",t.tm_mday)
        print("current hour",t.tm_hour)
        print("current week",t.tm_wday)
        print("day of year",t.tm_yday)
```

Output



```
>>> %Run time.py

 current time in sec 1643569089.242293
 current time  Mon Jan 31 00:28:09 2022
 current time sfter 30 sec Mon Jan 31 00:28:39 2022
  time  time.struct_time(tm_year=2022, tm_mon=1, tm_mday=31, t
 our=0, tm_min=28, tm_sec=9, tm_wday=0, tm_yday=31, tm_isdst=0
 current year 2022
 current month 1
 current day 31
 current hour 0
 current week 0
 day of year 31

>>>
```

PROGRAM NO:2

DATE :15/12/2021

Aim: 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 finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

Program:

Graphicuse.py

Package: graphics

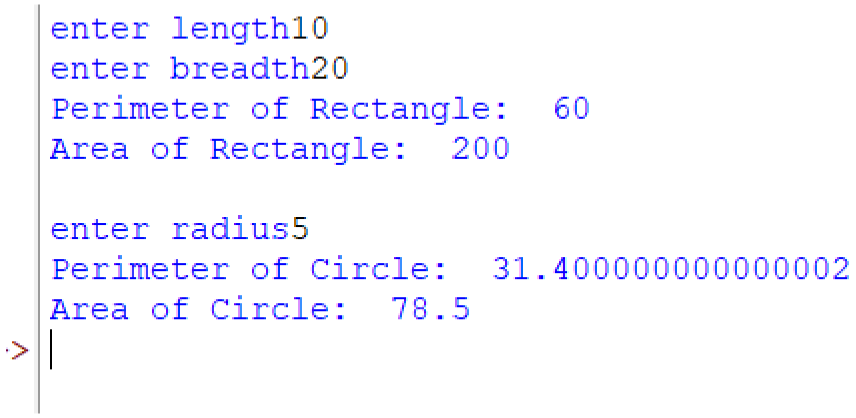
```
from graphics import rectangle
from graphics import circle
l=int(input("enter length"))
b=int(input("enter breadth"))
rectangle.perimeter(l,b)
rectangle.area(l,b)
print()
r=int(input("enter radius"))
circle.perimeter(r)
circle.area(r)
```

Circle.py

```
def
perimeter(r):
          return(2*3.14*r)
        def area(r):
          return(2*3.14*r*r)
```

Rectangle.py

```
def
perimeter(l,b):
           return(l+b)*2
         def area(l,b):
           return(l*b)
```



```
enter length10
enter breadth20
Perimeter of Rectangle:  60
Area of Rectangle:  200

enter radius5
Perimeter of Circle:  31.400000000000002
Area of Circle:  78.5
>
```

IV.COURSE OUTCOME(Co4)

PROGRAM NO:1

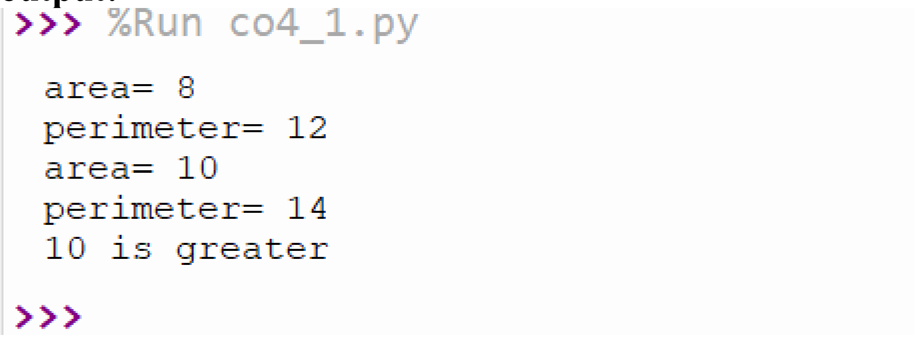
DATE :09/01/2022

Aim: create a rectangle class with attributes length and breadth and methods to find area and perimeter .compare 2 rectangle objects by their area.

Program:

```
class rectangle:
    def __init__(self,length,breadth):
        self.length=length
        self.breadth=breadth
    def area(self):
        a=self.length*self.breadth
        print("area=",a)
        return(a)
    def perimeter(self):
        p=2*(self.length+self.breadth)
        print("perimeter=",p)
        return(p)
p1=rectangle(4,2)
b=p1.area()
p1.perimeter()

p2=rectangle(5,2)
c=p2.area()
p2.perimeter()

if(b>c):
    print(b,"is greater")
else:
    print(c,"is greater")
```

output:



```
>>> %Run co4_1.py

 area= 8
 perimeter= 12
 area= 10
 perimeter= 14
 10 is greater

>>>
```

PROGRAM NO:2

DATE :09/01/2022

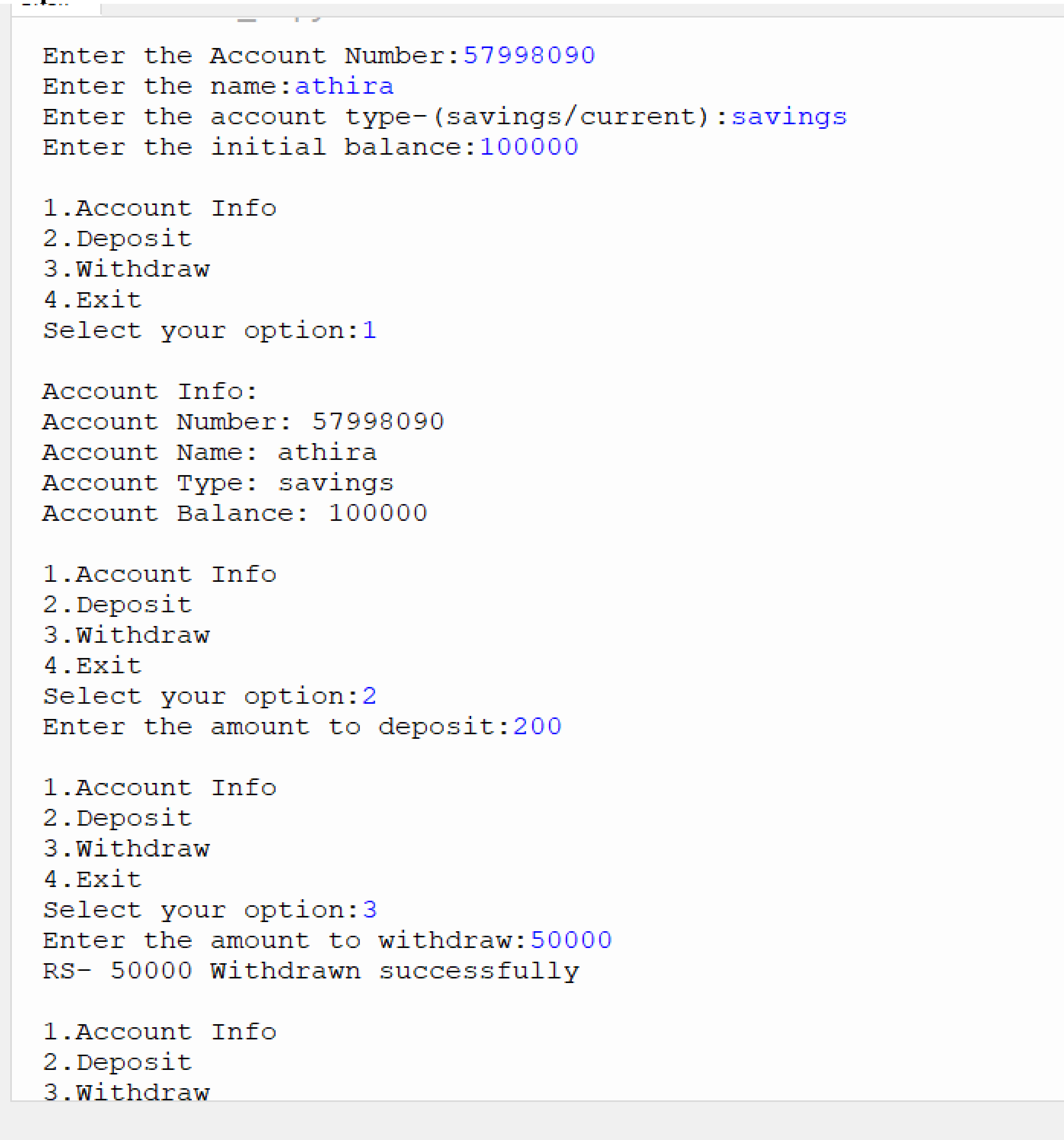
Aim: create a bank account with members account numbers, name, type of account ad balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

Program:

```
class bank:
 bal=0
 def __init__(self,accno,name,ac_type,bal):
  self.accno=accno
  self.name=name
  self.ac_type=ac_type
  self.bal=bal

 def display(self):
  print("\nAccount Info:")
  print("Account Number:",self.accno)
  print("Account Name:",self.name)
  print("Account Type:",self.ac_type)
  print("Account Balance:",self.bal)

 def deposit(self):
  dep=int(input("Enter the amount to deposit:"))
  self.bal=self.bal+dep

 def withdraw(self):
   w=int(input("Enter the amount to withdraw:"))
   if w > self.bal:
     print("Insufficient Balance")
   else:
     self.bal=self.bal-w
     print("RS-",w,"Withdrawn successfully")

acc_no=int(input("Enter the Account Number:"))
acc_name=input("Enter the name:")
acc_type=input("Enter the account type-(savings/current):")
balance=int(input("Enter the initial balance:"))
b1=bank(acc_no,acc_name,acc_type,balance)

while(1):
 print("\n1.Account Info\n2.Deposit\n3.Withdraw\n4.Exit")
 opt=int(input("Select your option:"))
 if opt == 1:
  b1.display()
 elif opt == 2:
  b1.deposit()
 elif opt == 3:
  b1.withdraw()
```

```
elif opt == 4:
 print("Exited")
 break
else:
 print("Invalid Option")
```

output:



```
Enter the Account Number:57998090
Enter the name:athira
Enter the account type-(savings/current):savings
Enter the initial balance:100000

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:1

Account Info:
Account Number: 57998090
Account Name: athira
Account Type: savings
Account Balance: 100000

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:2
Enter the amount to deposit:200

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:3
Enter the amount to withdraw:50000
RS- 50000 Withdrawn successfully

1.Account Info
2.Deposit
3.Withdraw
```

PROGRAM NO:3

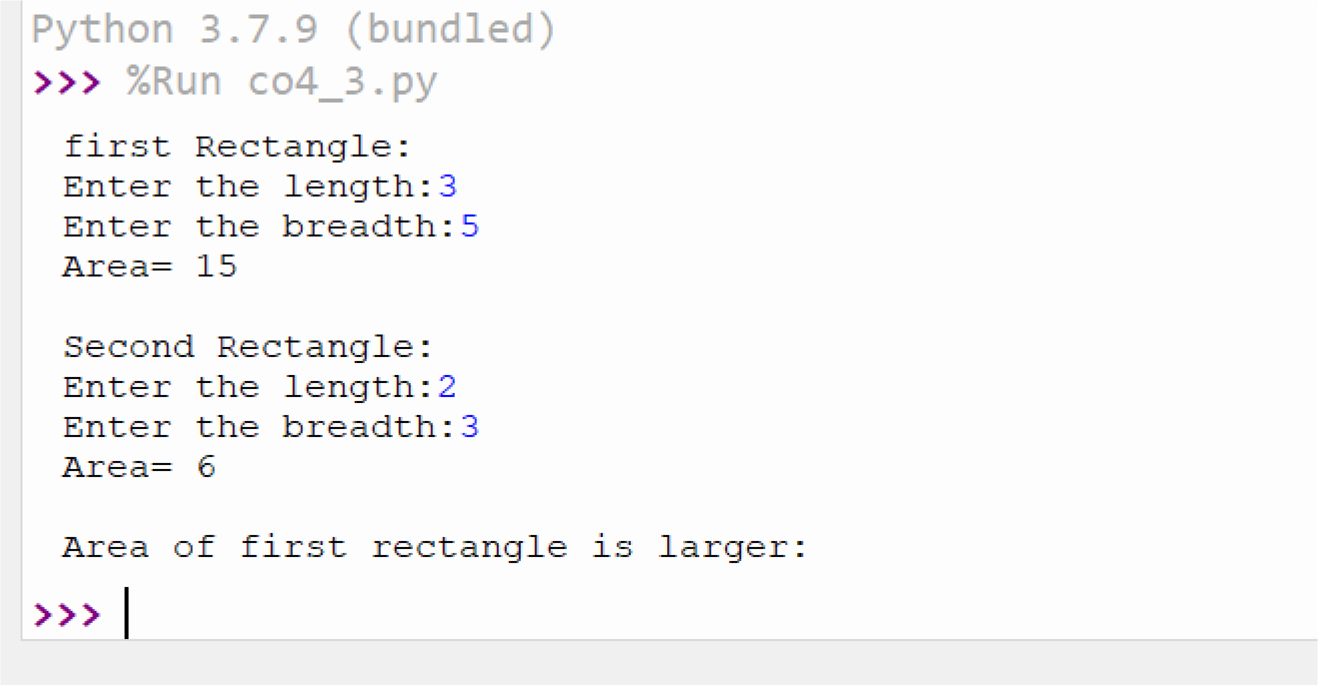
DATE :09/01/2022

Aim: create a class rectangle with private attributes length and width. Overload’<’ operator to compare the area of 2 rectangles.

Program:

```
class rectangle:

  def __init__(self,l,b):
    self.__length=l
    self.__breadth=b

  def area(self):
    self.area=self.__length*self.__breadth
    print("Area=",self.area)

  def __lt__(self,second):
   if self.area < second.area:
    return True
   else:
    return False

print("first Rectangle:")
len1=int(input("Enter the length:"))
bread1=int(input("Enter the breadth:"))
obj1=rectangle(len1,bread1)
obj1.area()

print("\nSecond Rectangle:")
len2=int(input("Enter the length:"))
bread2=int(input("Enter the breadth:"))
obj2=rectangle(len2,bread2)
obj2.area()

if obj1 < obj2 :
 print("\nArea of second rectangle is larger:")
else:
 print("\nArea of first rectangle is larger:")
```

output:



```
Python 3.7.9 (bundled)
>>> %Run co4_3.py

 first Rectangle:
 Enter the length:3
 Enter the breadth:5
 Area= 15

 Second Rectangle:
 Enter the length:2
 Enter the breadth:3
 Area= 6

 Area of first rectangle is larger:

>>> 
```

PROGRAM NO:4

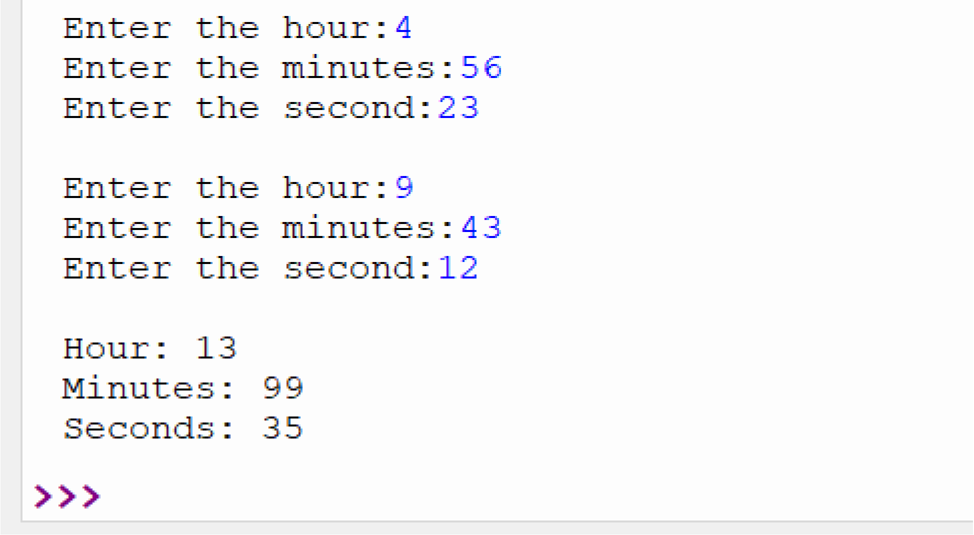
DATE :09/01/2022

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,hour,minute,second):
  self.__hour=hour
  self.__minute=minute
  self.__second=second

 def __add__(self,second):
   print("\nHour:",self.__hour + second.__hour)
   print("Minutes:",self.__minute + second.__minute)
   print("Seconds:",self.__second + second.__second)

hour1=int(input("Enter the hour:"))
minute1=int(input("Enter the minutes:"))
sec1=int(input("Enter the second:"))

obj1=time(hour1,minute1,sec1)

hour2=int(input("\nEnter the hour:"))
minute2=int(input("Enter the minutes:"))
sec2=int(input("Enter the second:"))

obj2=time(hour2,minute2,sec2)

obj1 + obj2
```

output:



```
Enter the hour:4
Enter the minutes:56
Enter the second:23

Enter the hour:9
Enter the minutes:43
Enter the second:12

Hour: 13
Minutes: 99
Seconds: 35

>>>
```

PROGRAM NO:5

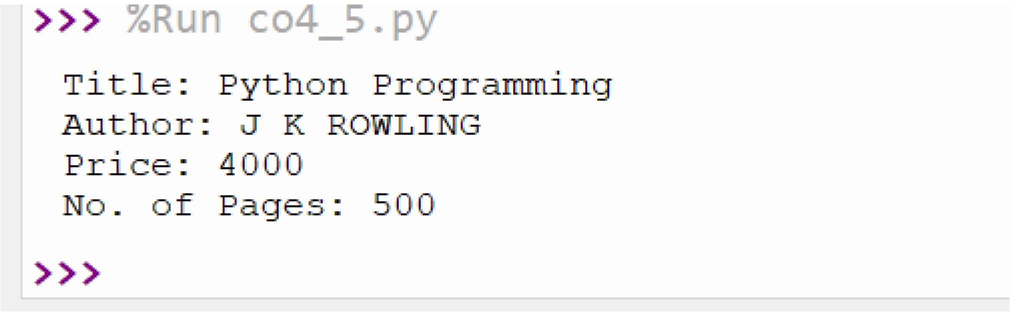
DATE :09/01/2022

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.**

Program:

```
class publisher:
    def __init__(self,title,author):
        self.title=title
        self.author=author
    def display(self):
        print("Title:",self.title)
        print("Author:",self.author)
class book(publisher):
    def __init__(self,price,no_of_page):
        self.price=price
        self.no_of_page=no_of_page
    def display(self):
        print("Price:",self.price)
        print("No. of Pages:",self.no_of_page)
class python(book):
    def __init__(self,title,author,price,no_of_page):
        publisher.__init__(self,title,author)
        book.__init__(self,price,no_of_page)
    def display(self):
        print("Title:",self.title)
        print("Author:",self.author)
        print("Price:",self.price)
        print("No. of Pages:",self.no_of_page)
p=python("Python Programming","J K ROWLING",4000,500)
p.display()
```

output:



```
>>> %Run co4_5.py

 Title: Python Programming
 Author: J K ROWLING
 Price: 4000
 No. of Pages: 500

>>>
```

V. COURSE OUTCOME 5(CO5)

PROGRAM NO:1

DATE :30/01/2022

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

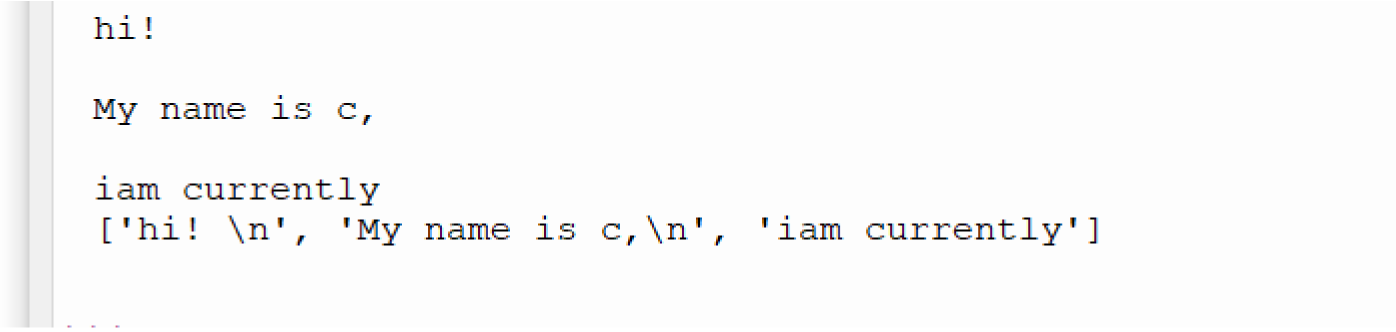
Program:

```
f1=open("file.txt","r")
print(f1.readline())
print(f1.readline())
print(f1.readline())
f1.seek(0,0)
print(f1.readlines())
print()
```

file.txt

hi!
My name is c,
iam currently

output:



```
hi!

My name is c,

iam currently
['hi! \n', 'My name is c,\n', 'iam currently']
```

PROGRAM NO:2

DATE :30/01/2022

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

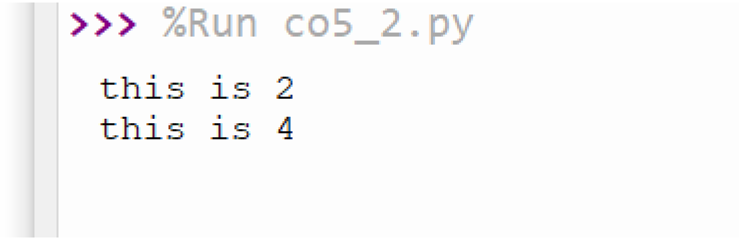
Program:

```
 fn = open('fil1.txt', 'r')
fn1 = open('fil2.txt', 'w')
cont = fn.readlines()
type(cont)
for i in range(0, len(cont)):
    if(i%2!=0):
        fn1.write(cont[i])
    else:
        pass
fn1.close()
fn1 = open('fil2.txt', 'r')
fn=open('fil1.txt','r')
cont1 = fn1.read()
print(cont1)
fn.close()
fn1.close()
```

fil1.txt

```
this is 1
this is 2
this is 3
this is 4
this is 5
```

output:



```
>>> %Run co5_2.py

 this is 2
 this is 4
```

PROGRAM NO:3

DATE :30/01/2022

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

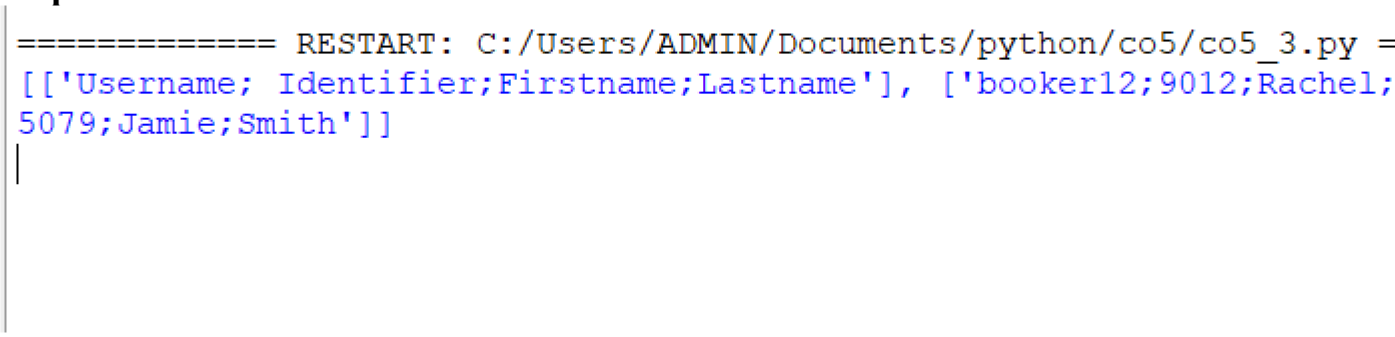
Program:

```
import csv
filename = "username.csv"
rows = []
cf=open(filename, 'r')
csvreader = csv.reader(cf)
for r in csvreader:
  rows.append(r)
print(rows)
cf.close()
```

username. cvs

```
Username; Identifier;Firstname;Lastname
booker12;9012;Rachel;Booker
grey07;2070;Laura;Grey
johnson81;4081;Craig;Johnson
jenkins46;9346;Mary;Jenkins
smith79;5079;Jamie;Smith
```

output:



```
============= RESTART: C:/Users/ADMIN/Documents/python/co5/co5_3.py =
[['Username; Identifier;Firstname;Lastname'], ['booker12;9012;Rachel;
5079;Jamie;Smith']]
```

PROGRAM NO:4

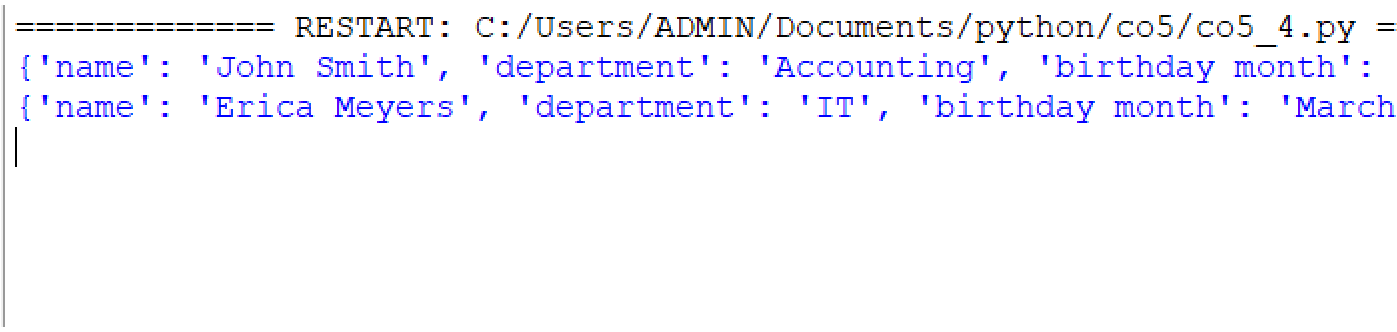
DATE :30/01/2022

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

Program:

```
import csv
filename = "emp.txt"
fields = []
rows = []
cf=open(filename, 'r')
csvreader = csv.DictReader(cf)
for r in csvreader:
  print(dict(r))
name,department,birthday month
John Smith,Accounting,November
Erica Meyers,IT,March
```

output:



```
============= RESTART: C:/Users/ADMIN/Documents/python/co5/co5_4.py =
{'name': 'John Smith', 'department': 'Accounting', 'birthday month':
{'name': 'Erica Meyers', 'department': 'IT', 'birthday month': 'March
```

PROGRAM NO:5

DATE :30/01/2022

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.

Program:

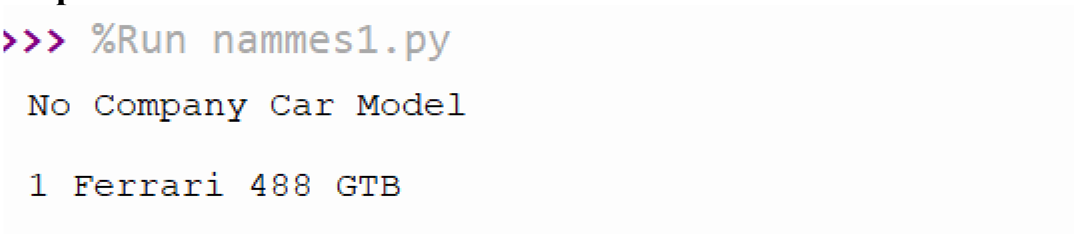
```
import csv

field_names = ['No', 'Company', 'Car Model']

cars = [
{'No': 1, 'Company': 'Ferrari', 'Car Model': '488 GTB'},
{'No': 2, 'Company': 'Porsche', 'Car Model': '918 Spyder'},
{'No': 3, 'Company': 'Bugatti', 'Car Model': 'La Voiture Noire'},
{'No': 4, 'Company': 'Rolls Royce', 'Car Model': 'Phantom'},
{'No': 5, 'Company': 'BMW', 'Car Model': 'BMW X7'},
]

with open('Names1.csv', 'w') as csvfile:
   writer = csv.DictWriter(csvfile, fieldnames = field_names)
   writer.writeheader()
   writer.writerows(cars)

#print(".................")
filename = "names1.csv"


cf=open(filename, 'r')
rows=[]
csvreader = csv.reader(cf)
for r in csvreader:
  rows.append(r)
for r in rows[:3]:
     print(*r)
```

names1.cvs

No	Company	Car Model
1	Ferrari	488 GTB
2	Porsche	918 Spyder
3	Bugatti	La Voiture Noire
4	Rolls Royce	Phantom

5	BMW	BMW X7

output:



```
>>> %Run nammes1.py
 No Company Car Model

 1 Ferrari 488 GTB
```