SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-2022

20 MCA 132 PROGRAMMING LABORATORY RECORD

Submitted by

JERIN JOY

REG NO: SNG21MCA-2019

In partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATIONS

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

20 MCA 132 PROGRAMMING LABORATORY RECORD

Certified that this is a Bonafide record of practical work done by **Mr. JERIN JOY** *to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree in* ***Master of Computer Applications*** *of Sree Narayana Gurukulam College of Engineering done during the Academic year 2021-2023.*

Kadayiruppu   Course Instructor

Date:

Head of the Department

Prof. Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2019   on

External Examiner   Internal Examiner

COURSE OUTCOME 1 (CO1)

PROGRAM NO: 1 **DATE: 03/11/2021**

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

It is a Graphical User Interface (GUI) where programmers write their code and produce the final products. An IDE basically unifies all essential tools required for software development and testing, which in turn helps the programming maximize his output.

- Features of IDE:-

1. Code Editor
2. Syntax Highlighting
3. Auto completion code
4. Debugger
5. Compiler
6. Language Support

IDLE is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- coded in 100% pure Python, using the tkinter GUI toolkit.
- cross-platform: works mostly the same on Windows, Unix, and macOS.
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages.
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features.
- search within any window, replace within editor windows, and search through multiple files (grep).
- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces.
- configuration, browsers, and other dialogs.

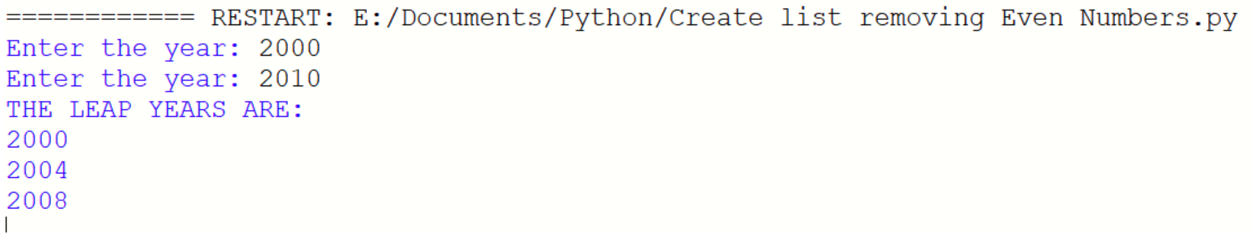
PROGRAM NO: 2 **DATE: 08/11/2021**

AIM : Write a program to Find Leap Year.

PROGRAM

```
years=int(input("Enter the year: "))
yeare=int(input("Enter the year: "))
if(years<yeare):
    print("THE LEAP YEARS ARE: ")
    for i in range(years,yeare):
        if(i%4==0 and i%100!=0 or i%400==0 ):
         print(i)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter the year: 2000
Enter the year: 2010
THE LEAP YEARS ARE:
2000
2004
2008
```

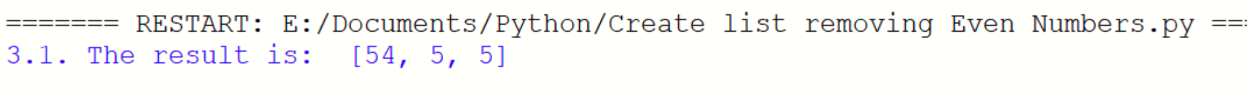
PROGRAM NO: 3 **DATE: 10/11/2021**

AIM : 3.1. Generate positive list of numbers from a given list of integers

PROGRAM

```
list1=[-465,54,-7,5,5,]

result=[number for number in list1 if number>=0]

print("3.1. The result is: ", result)

print()
```

OUTPUT



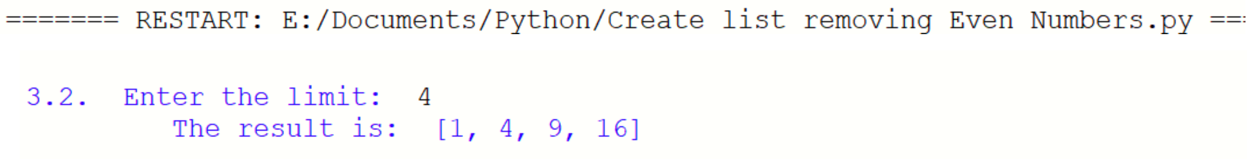
```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==
3.1. The result is:  [54, 5, 5]
```

AIM : 3.2. Write a program to find the Square of N number

PROGRAM

```
n=int(input("3.2.  Enter the limit:  "))

square=[i**2 for i in range(1,n+1)]

print("        The result is: ", square)

print()
```

OUTPUT



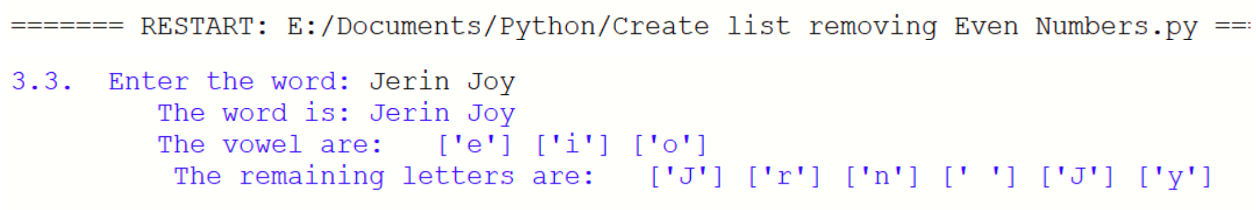
```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==

 3.2.  Enter the limit:  4
          The result is:  [1, 4, 9, 16]
```

AIM : 3.3. Form a list of vowels selected from a given word

PROGRAM

```
n=str(input("3.3.  Enter the word: "))
print("        The word is: "+n)
print("        The vowel are:  " ,end=" ")
for i in n:
   if i in 'aeiouAEIOU':
      print([i],end=" ")
print(" \n        The remaining letters are:  ",end=" ")
for j in n:
 if j not in 'aeiouAEIOU':
      print([j],end=" ")
print()
print()
```

OUTPUT



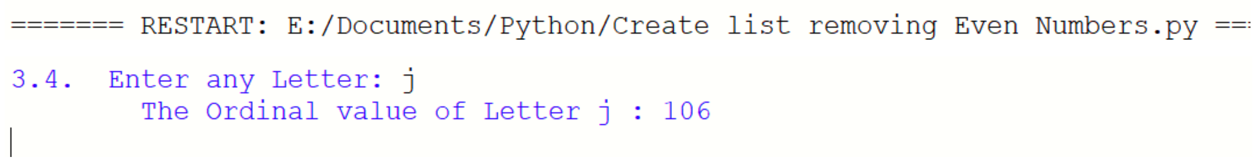
```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==

3.3.  Enter the word: Jerin Joy
         The word is: Jerin Joy
         The vowel are:   ['e'] ['i'] ['o']
          The remaining letters are:   ['J'] ['r'] ['n'] [' '] ['J'] ['y']
```

AIM : 3.4. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

PROGRAM

```
n=str(input("3.4.  Enter any Letter: "))
print("       The Ordinal value of Letter " +n ,":",end=" ")
for i in n:
 print(ord(i),end=" ")
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==

3.4.  Enter any Letter: j
        The Ordinal value of Letter j : 106
```

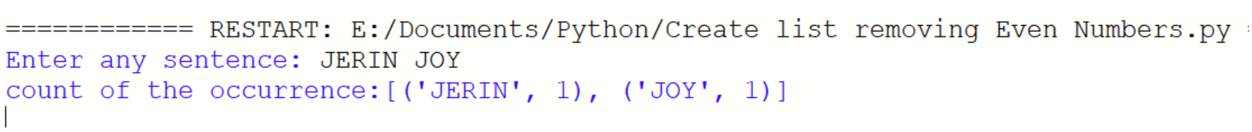
PROGRAM NO: 4 **DATE: 15/11/2021**

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

PROGRAM

```
str1=input("Enter any sentence: ")

wordlist=str1.split()

count= []

for i in wordlist:

    count.append(wordlist.count(i))

print("count of the occurrence:" + str(list(zip(wordlist, count))))
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter any sentence: JERIN JOY
count of the occurrence:[('JERIN', 1), ('JOY', 1)]
|
```

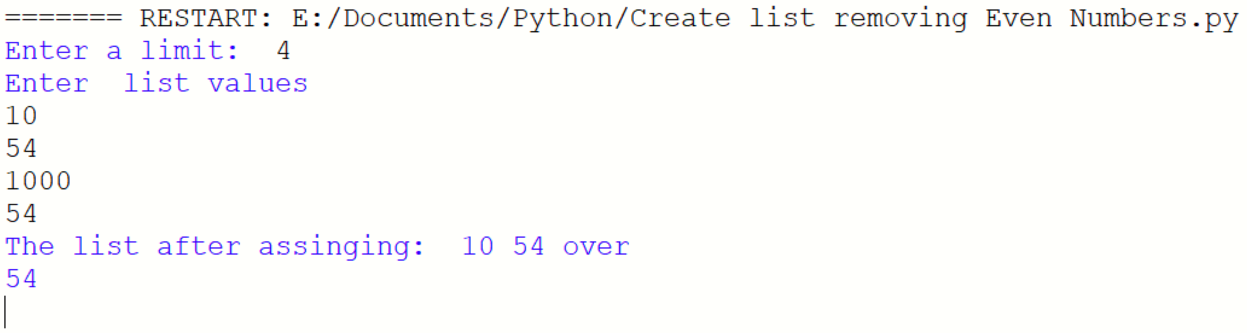
PROGRAM NO: 5 **DATE: 15/11/2021**

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  list values")

for i in range(0,s):

       n.append(int(input()))

print("The list after assinging: ",end=" ")

for i in range(0,len(n)):

  if n[i]<=100:

      print(n[i] ,end=" ")

  else:

      print("over")
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter a limit:  4
Enter  list values
10
54
1000
54
The list after assinging:  10 54 over
54
```

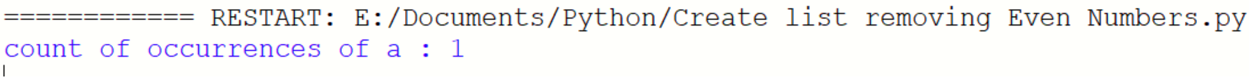
PROGRAM NO: 6 **DATE: 17/11/2021**

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

PROGRAM

```
list1 = ["a", "b", "c","c","b"]

occ = list1.count("a")

print("count of occurrences of a :",occ)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
count of occurrences of a : 1
```

PROGRAM NO: 7 **DATE: 17/11/2021**

AIM : Enter 2 lists of integers. Check

(a) Whether list are of same length.

(b) whether list sums to same value.

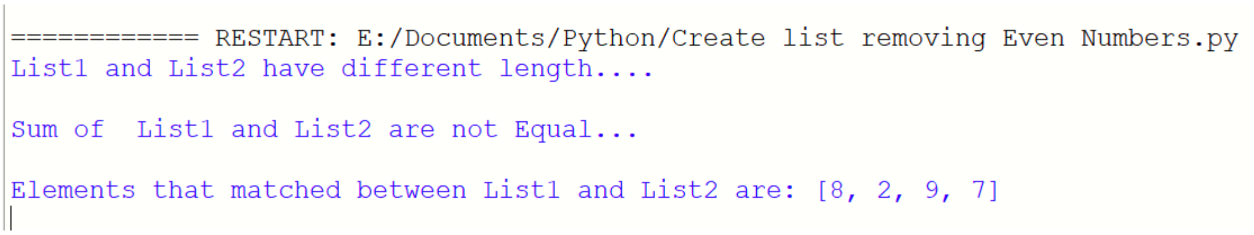
(c) whether any value occur in both.

PROGRAM

```
list1=[4,8,2,4,9,5,7]

list2=[2,1,7,9,8,6]

s=int(0)

c=int(0)


if len(list1)==len(list2):

  print("List1 and List2 are of Same length...")

else:

  print("List1 and List2 have different length....")

print()

for i in range(0,len(list1) and len(list2)):

  s=s+list1[i]

  c=c+list2[i]

if(s==c):

  print("Sum of  List1 and List2 are Equal...")

else:

  print("Sum of  List1 and List2 are not Equal...")

print()

print("Elements that matched between List1 and List2 are:", end=" ")

l=[]
```

```
for i in range(0,len(list1)):
  for j in range(0,len(list2)):
    if list1[i]==list2[j]:
        l.append(list1[i] and list2[j])
    else:
      continue
print(l)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
List1 and List2 have different length....

Sum of  List1 and List2 are not Equal...

Elements that matched between List1 and List2 are: [8, 2, 9, 7]
```

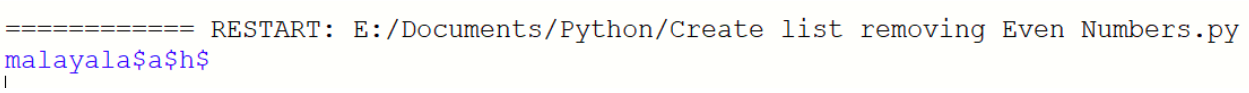
PROGRAM NO: 8 **DATE: 22/11/2021**

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

PROGRAM

```
str1="malayalamamhm"

char = str1[0]

str1 = str1.replace(char, '$')

str1 = char + str1[1:]

print(str1)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
malayala$a$h$
```

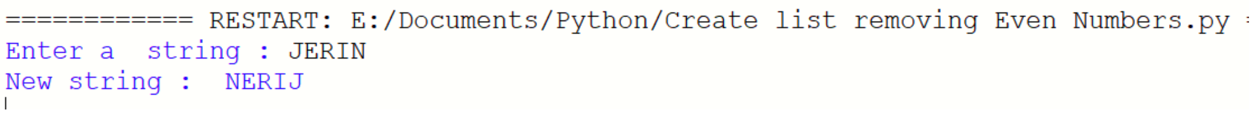
PROGRAM NO: 9 **DATE: 22/11/2021**

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

PROGRAM

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

newstr = str[-1:] +str[1:-1] + str[:1]

print("New string : ",newstr)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter a  string : JERIN
New string :  NERIJ
```

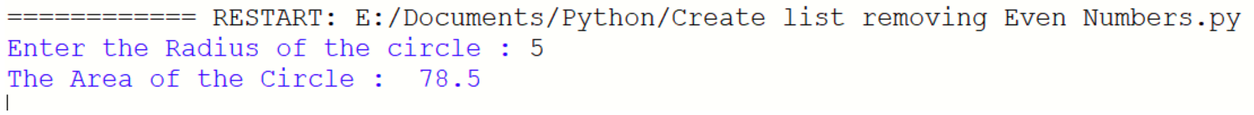
PROGRAM NO: 10 **DATE: 24/11/2021**

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

PROGRAM

```
pi=3.14

r = float(input ("Enter the Radius of the circle : "))

result=3.14 * r**2

print ("The Area of the Circle : ", result)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter the Radius of the circle : 5
The Area of the Circle :  78.5
```

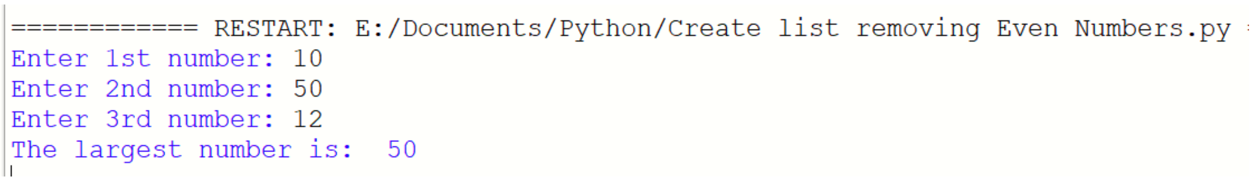
PROGRAM NO: 11 **DATE: 24/11/2021**

AIM : Write a program to find biggest of 3 numbers entered.

PROGRAM

```
x = int(input("Enter 1st number: "))
y = int(input("Enter 2nd number: "))
z = int(input("Enter 3rd number: "))
if (x > y) and (x > z):
    largest = x
elif (y > x) and (y > z):
    largest = y
else:
    largest = z
print("The largest number is: ",largest)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter 1st number: 10
Enter 2nd number: 50
Enter 3rd number: 12
The largest number is:  50
```

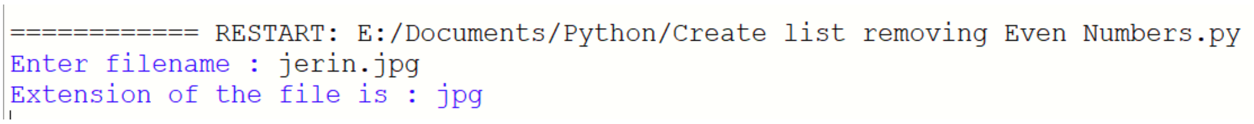
PROGRAM NO: 12 **DATE: 24/11/2021**

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

PROGRAM

```
file= input("Enter filename : ")

f=file.split(".")

print("Extension of the file is : " + f[-1])
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter filename : jerin.jpg
Extension of the file is : jpg
```

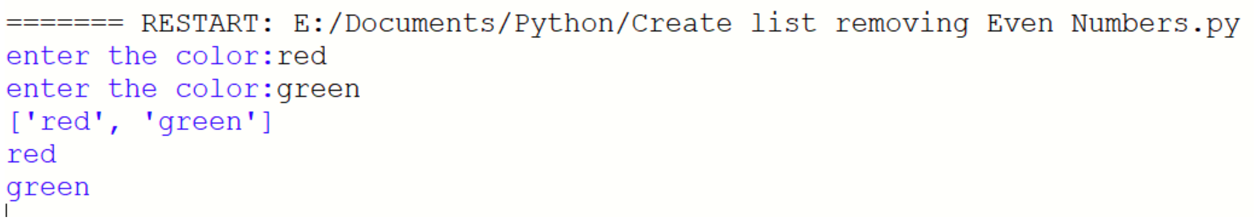
PROGRAM NO: 13 **DATE: 29/11/2021**

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(2):

   b=input("enter the color:")

   a.append(b)

print(a)

print(a[0])

print(a[1])
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
enter the color:red
enter the color:green
['red', 'green']
red
green
```

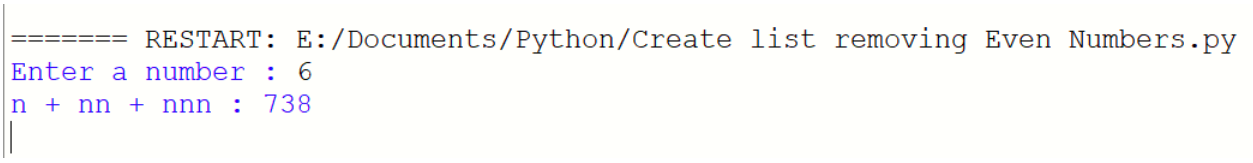
PROGRAM NO: 14 **DATE: 29/11/2021**

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

PROGRAM

```
n=int(input("Enter a number : "))

x=int("%s" % n)

y=int("%s%s" % (n,n))

z=int("%s%s%s" % (n,n,n))

print("n + nn + nnn :",x+y+z)
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter a number : 6
n + nn + nnn : 738
```

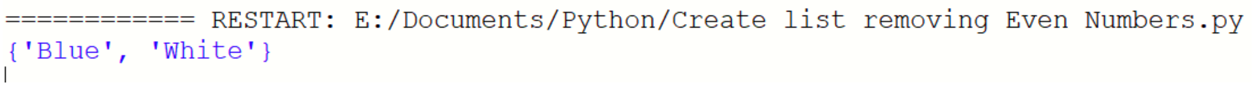
PROGRAM NO: 15 **DATE: 29/11/2021**

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

PROGRAM

```
color_list_1 = set(["White", "pink", "Red","Blue"])

color_list_2 = set(["Red", "Green","pink"])

print(color_list_1.difference(color_list_2))
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
{'Blue', 'White'}
```

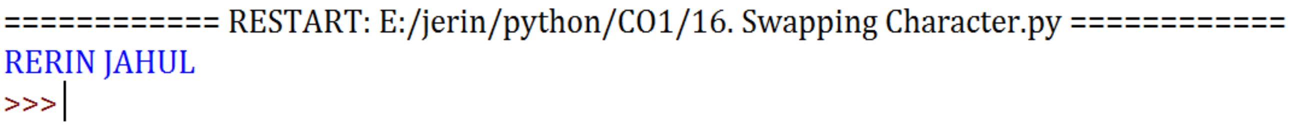
PROGRAM NO: 16 **DATE: 29/11/2021**

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

PROGRAM

```
a="JERIN"

b="RAHUL"

c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]

print(c)
```

OUTPUT



```
============ RESTART: E:/jerin/python/CO1/16. Swapping Character.py ============
RERIN JAHUL
>>>
```

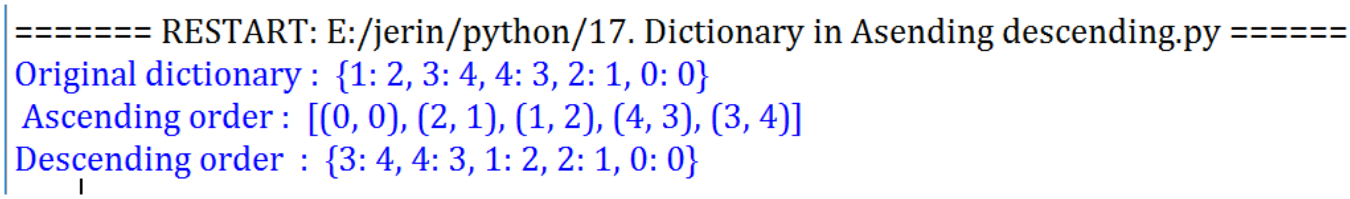
PROGRAM NO: 17 **DATE: 01/12/2021**

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)

sortedic = sorted(d.items(), key=operator.itemgetter(1))

print(' Ascending order : ',sortedic)

sortedic = dict( sorted(d.items(), key=operator.itemgetter(1),reverse=True))

print('Descending order  : ',sortedic)
```

OUTPUT



```
======= RESTART: E:/jerin/python/17. Dictionary in Asending descending.py ======
Original 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}
```

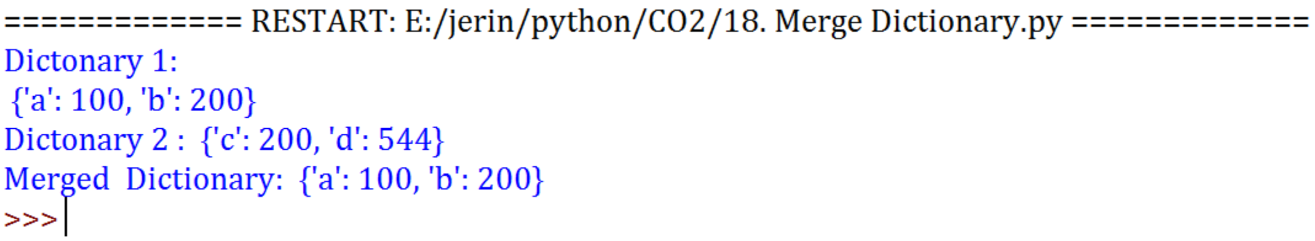
PROGRAM NO: 18 **DATE: 01/12/2021**

AIM : Write a program to merge two dictionaries.

PROGRAM

```
dic1={'a':100,'b':200}

dic2={'c':200,'d':544}

print("Dictonary 1: ", dic1)

print("Dictonary 2 : ", dic2)

dic3=dic1.copy()

dic3.update(dic1)

print("Merged  Dictionary: ", dic3)
```

OUTPUT



```
============= RESTART: E:/jerin/python/CO2/18. Merge Dictionary.py =============
Dictonary 1:
 {'a': 100, 'b': 200}
Dictonary 2 :  {'c': 200, 'd': 544}
Merged  Dictionary:  {'a': 100, 'b': 200}
>>>
```

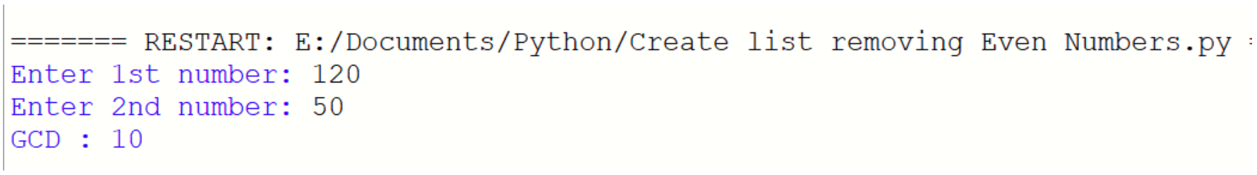
PROGRAM NO: 19 **DATE: 01/12/2021**

AIM : Write a program to find GCD of 2 numbers.

PROGRAM

```
x= int(input("Enter 1st number: "))
y= int(input("Enter 2nd number: "))
i = 1
while(i <= x and i <= y):
   if(x % i == 0 and y% i == 0):
       gcd = i
   i = i + 1
print("GCD :", gcd)
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter 1st number: 120
Enter 2nd number: 50
GCD : 10
```

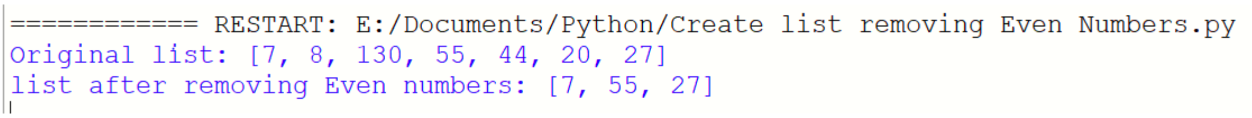
PROGRAM NO: 20 **DATE: 01/12/2021**

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

PROGRAM

```
number = [7,8, 130, 55, 44, 20, 27]

print( "Original list:",number)

number = [x for x in number if x%2!=0]

print("list after removing Even numbers:",number)
```

OUTPUT



```
============ RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Original list: [7, 8, 130, 55, 44, 20, 27]
list after removing Even numbers: [7, 55, 27]
```

COURSE OUTCOME 2 (CO2)

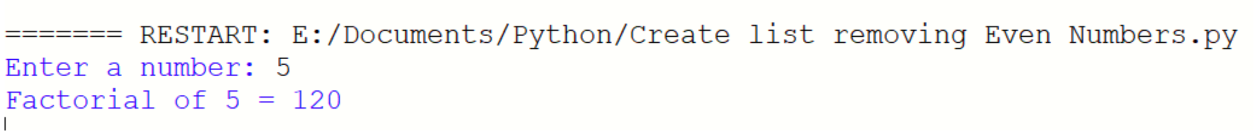
PROGRAM NO: 1 **DATE: 06/12/2021**

AIM : Write a 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 of',n, '=',f)
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter a number: 5
Factorial of 5 = 120
```

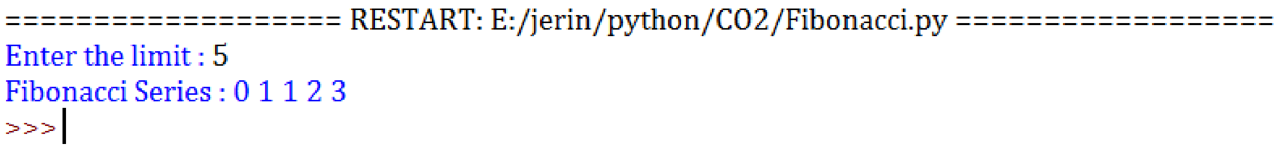
PROGRAM NO: 2 **DATE: 06/12/2021**

AIM : Write a program to generate Fibonacci series of N terms.

PROGRAM

```
n = int(input("Enter the limit : "))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci Series :",end= " ")
while(count <= n):
    print(sum, end = " ")
    count += 1
    a = b
    b = sum
    sum = a + b
```

OUTPUT



```
================== RESTART: E:/jerin/python/CO2/Fibonacci.py ==================
Enter the limit : 5
Fibonacci Series : 0 1 1 2 3
>>> 
```

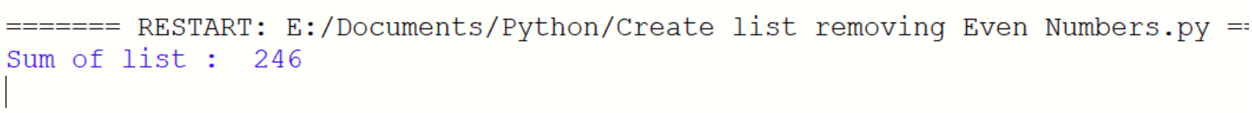
PROGRAM NO: 3 **DATE: 06/12/2021**

AIM : Write a program to find the sum of all items in a list.

PROGRAM

```
list1 = [20, 85, 20, 25, 56, 40]

total = sum(list1)

print("Sum of list : ",total)
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==
Sum of list :  246
```

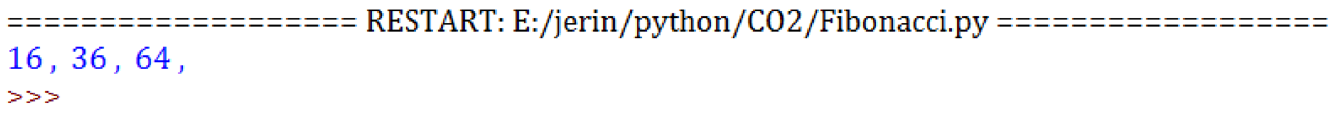
PROGRAM NO: 4 **DATE: 06/12/2021**

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(10,100):

   if s(i)==int(s(i)) and i%2==0:

    print(i,", ",end=" ")
```

OUTPUT



```
================== RESTART: E:/jerin/python/CO2/Fibonacci.py ==================
16 , 36 , 64 ,
>>>
```

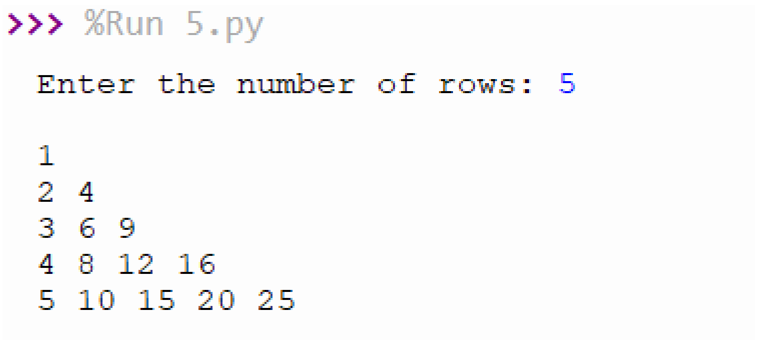
PROGRAM NO: 5 **DATE: 06/12/2021**

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

PROGRAM

```
rows = int(input("Enter the number of rows: "))
for i in range(1, rows+1):
    print()
    for j in range(1,i+1):
        print(i * j, end=' ')
print()
```

OUTPUT



```
>>> %Run 5.py
 Enter the number of rows: 5

 1
 2 4
 3 6 9
 4 8 12 16
 5 10 15 20 25
```

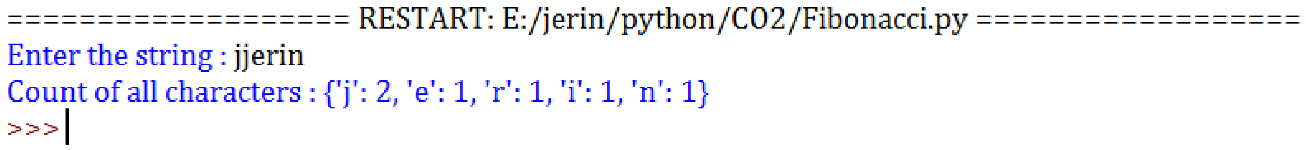
PROGRAM NO: 6 **DATE: 06/12/2021**

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



```
=================== RESTART: E:/jerin/python/CO2/Fibonacci.py ==================
Enter the string : jjerin
Count of all characters : {'j': 2, 'e': 1, 'r': 1, 'i': 1, 'n': 1}
>>>
```

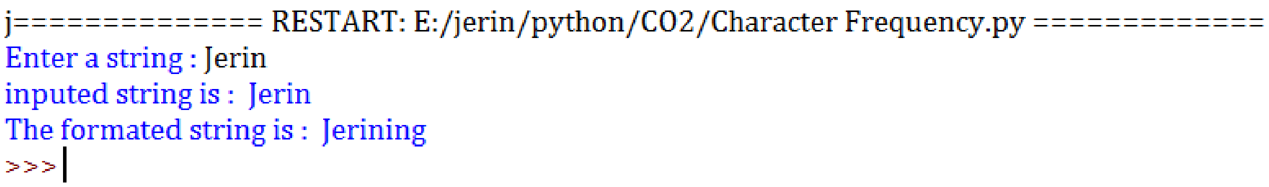
PROGRAM NO: 7 **DATE: 08-12-2021**

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

PROGRAM

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

print("inputed string is : ",str)

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

    str=str+'ly'

else:

    str=str+'ing'

print("The formated string is : ",str)
```

OUTPUT



```
j============== RESTART: E:/jerin/python/CO2/Character Frequency.py =============
Enter a string : Jerin
inputed string is :  Jerin
The formated string is :  Jerining
>>>
```

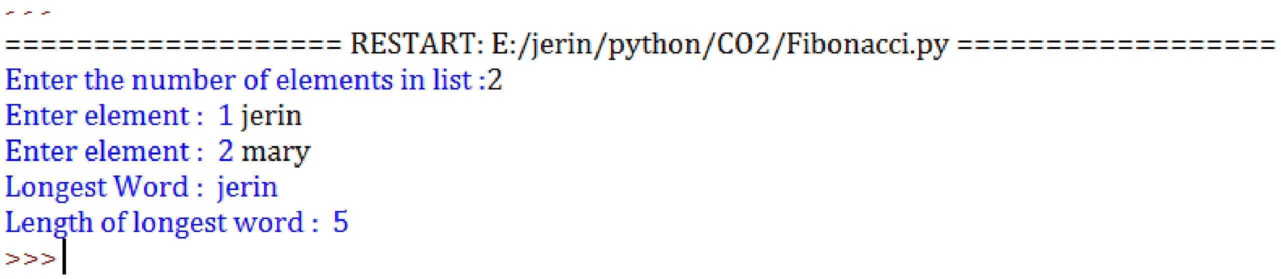
PROGRAM NO: 8 **DATE: 08-12-2021**

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

OUTPUT



```
=================== RESTART: E:/jerin/python/CO2/Fibonacci.py ==================
Enter the number of elements in list :2
Enter element :  1 jerin
Enter element :  2 mary
Longest Word :  jerin
Length of longest word :  5
>>>
```

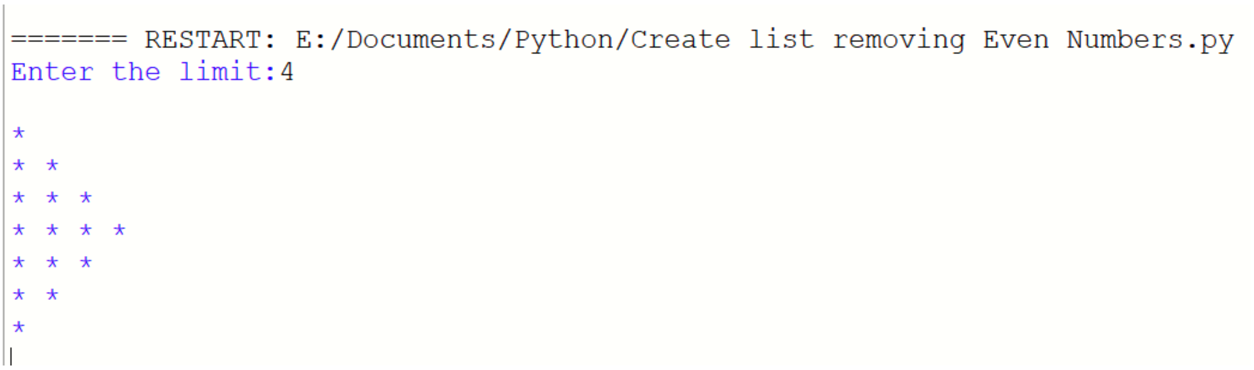
PROGRAM NO: 9 **DATE: 08-12-2021**

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



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py
Enter the limit:4

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

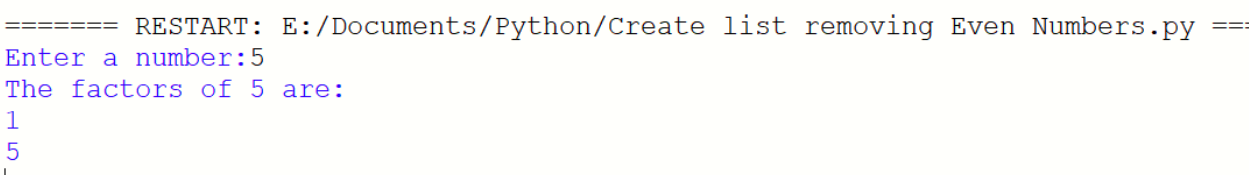
PROGRAM NO: 10 **DATE: 08-12-2021**

AIM : Generate all factors of a number. def print_factors(x):

PROGRAM

```
def factors(x):

 print("The factors of",x,"are:")

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

    if x % i == 0:

        print(i)

n=int(input("Enter a number:"))

factors(n)
```

OUTPUT



```
======= RESTART: E:/Documents/Python/Create list removing Even Numbers.py ==
Enter a number:5
The factors of 5 are:
1
5
```

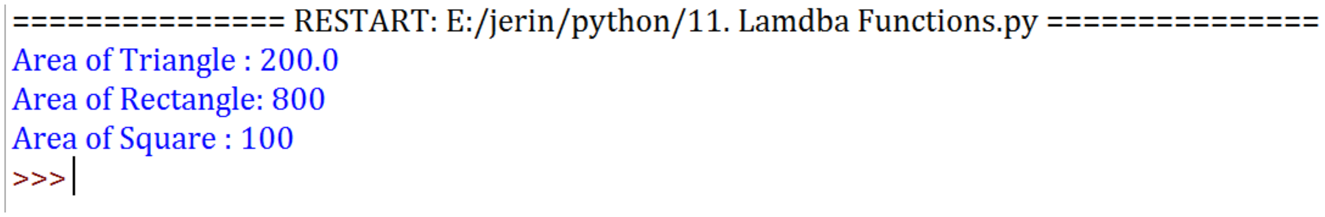
PROGRAM NO: 11 **DATE: 08-12-2021**

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

PROGRAM

```
import math

triangle = lambda b,h : 1/2*b*h

rectangle = lambda l,b : l*b

square = lambda a : a*a


print("Area of Triangle :", triangle(20,20))

print("Area of Rectangle:", rectangle(40,20))

print("Area of Square :", square(10))
```

OUTPUT



```
=============== RESTART: E:/jerin/python/11. Lamdba Functions.py ===============
Area of Triangle : 200.0
Area of Rectangle: 800
Area of Square : 100
>>>
```

COURSE OUTCOME 3 (CO3)

PROGRAM NO: 1 **DATE: 13-12-2021**

AIM : Work with built-in packages.

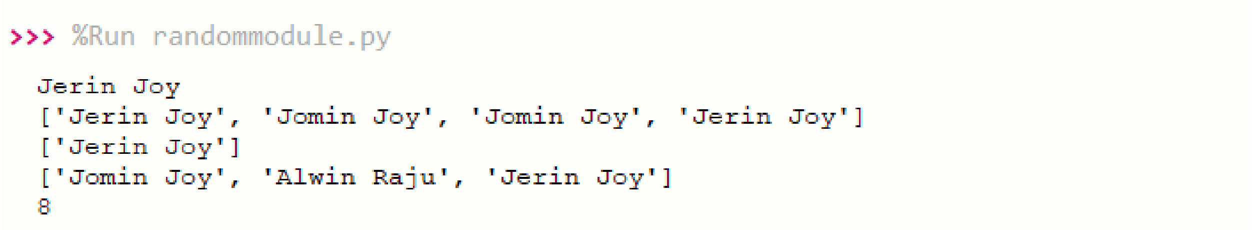
A) **random module.**
B) **time module**
C) **calendar module**
D) **math module**
E) **Statistics module**

A) random module

PROGRAM

```
import random

mylist = ["Jerin Joy", "Alwin Raju", "Jomin Joy"]

print(random.choice(mylist)) #Returns a random element from the given sequence

print(random.choices(mylist, k=4))

print(random.sample(mylist, k=1)) #Return a list that contains any 2 of the items

                         from a list:

random.shuffle(mylist)

print(mylist) #Takes a sequence and returns the sequence in a random order

print(random.randrange(3, 9)) #Return a number between 3 and 9
```

OUTPUT



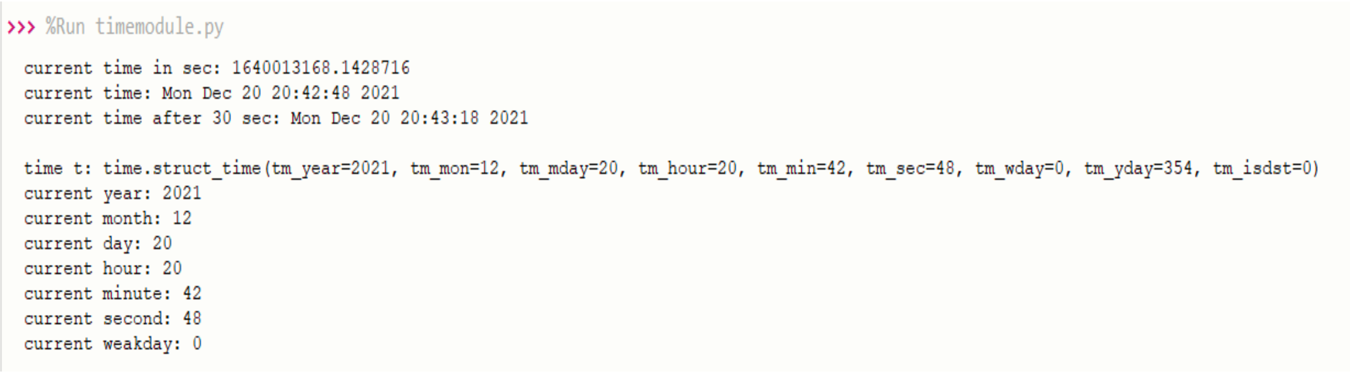
```
>>> %Run randommodule.py

  Jerin Joy
  ['Jerin Joy', 'Jomin Joy', 'Jomin Joy', 'Jerin Joy']
  ['Jerin Joy']
  ['Jomin Joy', 'Alwin Raju', 'Jerin Joy']
  8
```

B) time module

PROGRAM

```
import time

print("current time in sec:",time.time())

print("current time:",time.ctime())

print("current time after 30 sec:",time.ctime(time.time()+30))

print()

t=time.localtime()

print("time t:",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 minute:",t.tm_min)

print("current second:",t.tm_sec)

print("current weakday:",t.tm_wday)
```

OUTPUT



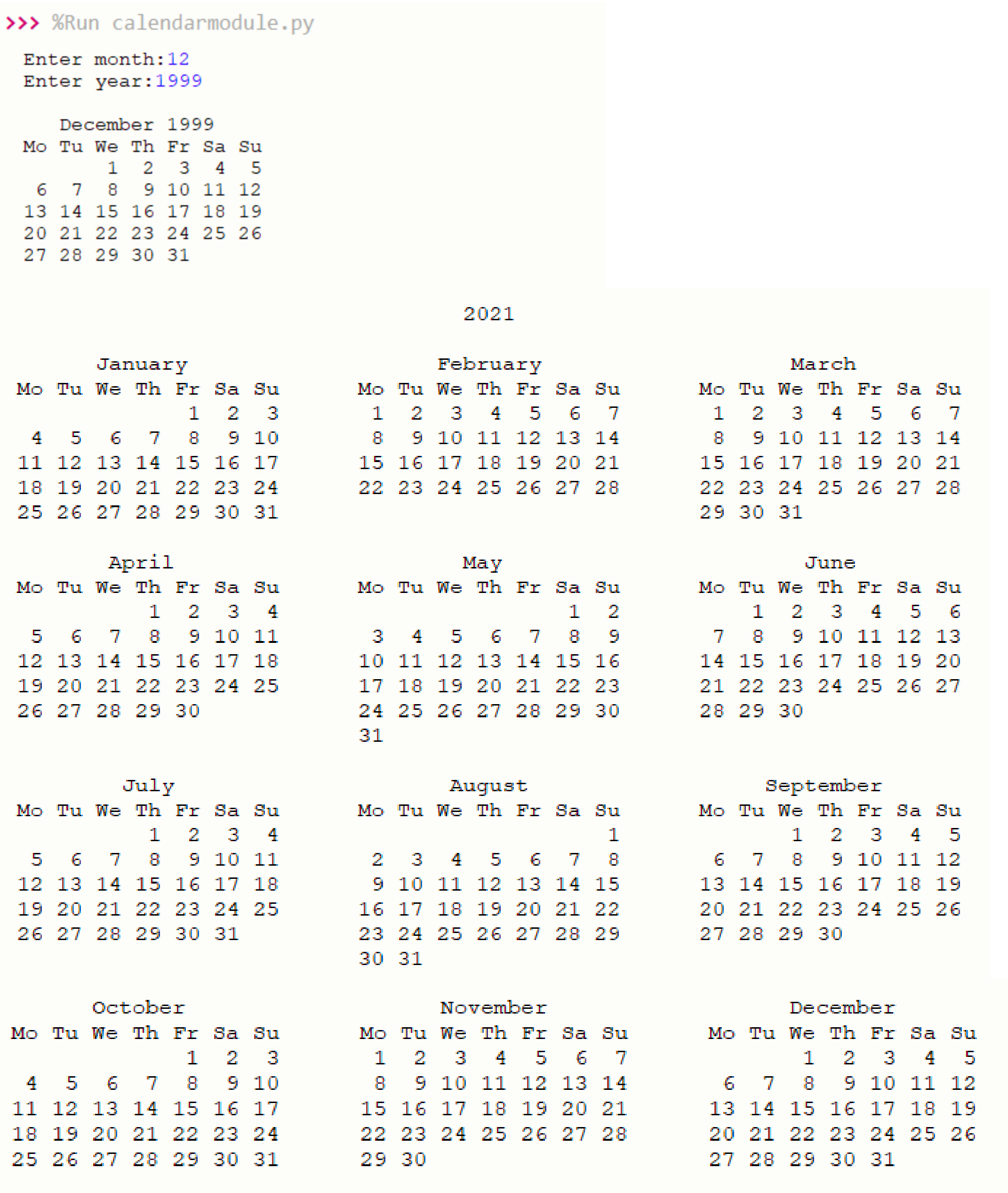
```
>>> %Run timemodule.py

  current time in sec: 1640013168.1428716
  current time: Mon Dec 20 20:42:48 2021
  current time after 30 sec: Mon Dec 20 20:43:18 2021

  time t: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=20, tm_min=42, tm_sec=48, tm_wday=0, tm_yday=354, tm_isdst=0)
  current year: 2021
  current month: 12
  current day: 20
  current hour: 20
  current minute: 42
  current second: 48
  current weakday: 0
```

C) calendar module

PROGRAM

```
import calendar

month=int(input("Enter month:"))

year=int(input("Enter year:"))

print()

print(calendar.month(year,month)) #calendar of a given month

print(calendar.calendar(2021)) #calendar of a given year
```

OUTPUT



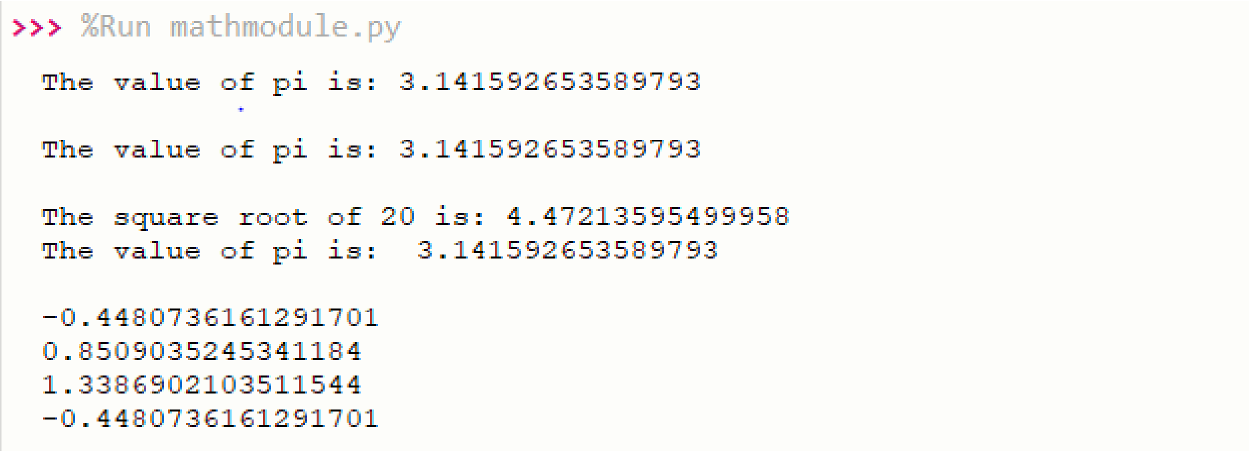
```
>>> %Run calendarmodule.py

 Enter month:12
 Enter year:1999

    December 1999
 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 31

                                  2021

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

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

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

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

D) math module

PROGRAM

```
import math

print("The value of pi is:",math.pi)

print()

import math as m

print("The value of pi is:",m.pi)

print()

from math import pi,sqrt

print("The square root of 20 is:",math.sqrt(20))

print("The value of pi is: ",math.pi)

print()

print(math.cos(90))

print(math.sin(45))

print(math.tan(180))

print(math.cos(90)
```

OUTPUT



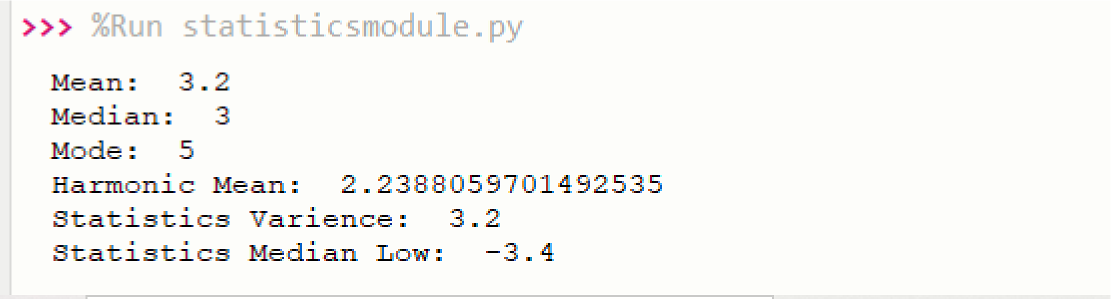
```
>>> %Run mathmodule.py
  The value of pi is: 3.141592653589793

  The value of pi is: 3.141592653589793

  The square root of 20 is: 4.47213595499958
  The value of pi is:  3.141592653589793

  -0.4480736161291701
  0.8509035245341184
  1.3386902103511544
  -0.4480736161291701
```

E) Statistics module

PROGRAM

```
import statistics

list1=[1,2,3,5,5]

print("Mean: ",statistics.mean(list1))

print("Median: ",statistics.median(list1))

print("Mode: ",statistics.mode(list1))

print("Harmonic Mean: ",statistics.harmonic_mean(list1))

print("Statistics Varience: ",statistics.variance(list1))

print("Statistics Median Low: ",statistics.median_low([-12, 6.6, -3.4, 7.1, -9, 22]))
```

OUTPUT



```
>>> %Run statisticsmodule.py

  Mean:  3.2
  Median:  3
  Mode:  5
  Harmonic Mean:  2.2388059701492535
  Statistics Varience:  3.2
  Statistics Median Low:  -3.4
```

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

graphicsuse.py

```
from graphics import rectangle

from graphics import circle

from graphics.ThreeD_graphics import cuboid

from graphics.ThreeD_graphics import sphere


l=int(input("Enter the length of rectangle: "))

b=int(input("Enter the breath of rectangle: "))

rectangle.area(l,b)

rectangle.perimeter(l,b)

print()

r=int(input("Enter the Radius of Circle: "))

circle.area(r)

circle.perimeter(r)

print()

l=int(input("Enter the length of Cuboid: "))

b=int(input("Enter the breadth of Cuboid: "))

h=int(input("Enter the height of Cuboid: "))

cuboid.area(l,b,h)

cuboid.perimeter(l,b,h)
```

```
print()

r=int(input("Enter the radius of Sphere: "))

sphere.area(r)

sphere.volume(r)
```

--

Package : graphics

circle.py

```
def area(r):
    print("Area of Circle: ",3.14*r*r)
def perimeter(r):
    print("Perimeter of Circle: ", 2*3.14*r)
```

--

rectangle.py

```
def area(l,b):
    print("Area of Rectangle: ", l*b)
def perimeter(l,b):
    print("Perimeter of Rectangle: ", 2*(l+b))
```

--

Sub-Package : ThreeD_graphics

cuboid.py

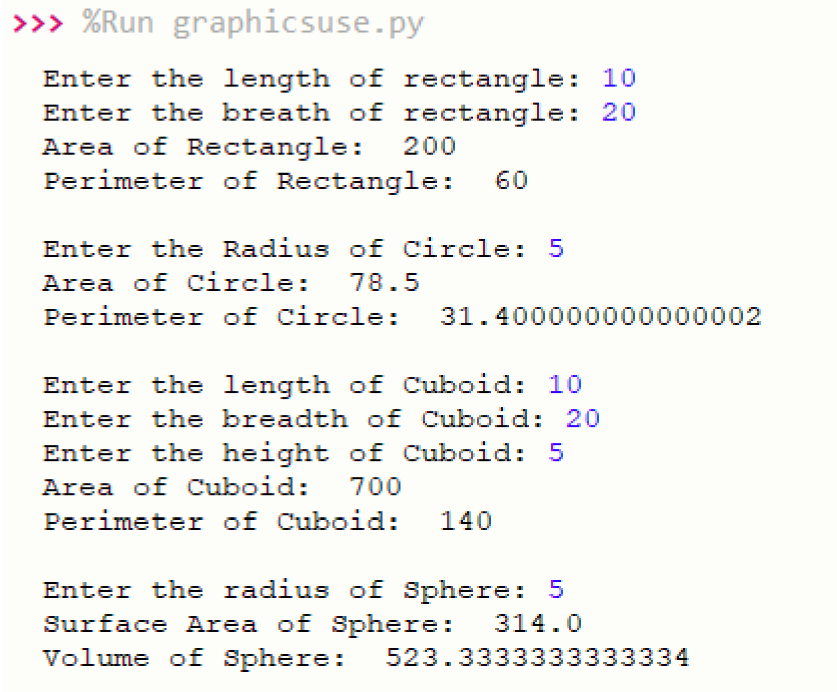
```
def area(l,b,h):
   print("Area of Cuboid: ",(2*l*b)+(2*l*h)+(2*h*b))
def perimeter(l,b,h):
   print("Perimeter of Cuboid: ", 4*(l+b+h))
```

----

sphere.py

```
def area(r):
   print("Surface Area of Sphere: ",4*3.14*r*r)
def volume(r):
   print("Volume of Sphere: ",(4/3)*3.14*r*r*r)
```

OUTPUT



```
>>> %Run graphicsuse.py
  Enter the length of rectangle: 10
  Enter the breath of rectangle: 20
  Area of Rectangle:  200
  Perimeter of Rectangle:  60

  Enter the Radius of Circle: 5
  Area of Circle:  78.5
  Perimeter of Circle:  31.400000000000002

  Enter the length of Cuboid: 10
  Enter the breadth of Cuboid: 20
  Enter the height of Cuboid: 5
  Area of Cuboid:  700
  Perimeter of Cuboid:  140

  Enter the radius of Sphere: 5
  Surface Area of Sphere:  314.0
  Volume of Sphere:  523.3333333333334
```

COURSE OUTCOME 4 (CO4)

PROGRAM NO: 1 **DATE: 3-01-2022**

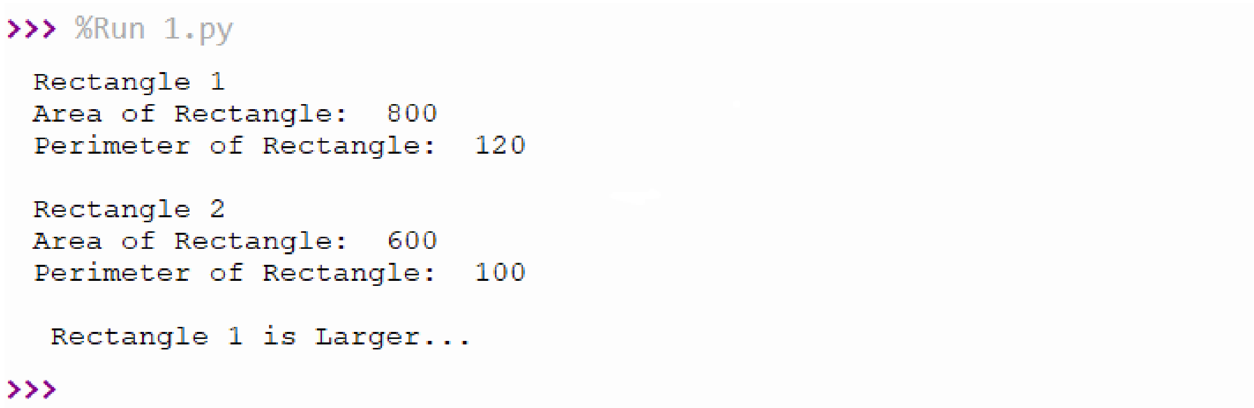
AIM : Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

PROGRAM

```
class rectangle:
  def __init__(self,l,b):
    self.length=l
    self.breadth=b

  def area(self):
    area=self.length*self.breadth
    print("Area of Rectangle: ",area)
    return(area)

  def perimeter(self):
    perimeter=2*(self.length+self.breadth)
    print("Perimeter of Rectangle: ",perimeter)

print("Rectangle 1")
obj1=rectangle(40,20)
a1=obj1.area()
obj1.perimeter()

print("\nRectangle 2")
obj2=rectangle(30,20)
a2=obj2.area()
obj2.perimeter()
```

```
if a1 > a2:
 print("\n Rectangle 1 is Larger...")
else:
 print("\n Rectangle 2 is Larger...")
```

OUTPUT



```
>>> %Run 1.py

 Rectangle 1
 Area of Rectangle:  800
 Perimeter of Rectangle:  120

 Rectangle 2
 Area of Rectangle:  600
 Perimeter of Rectangle:  100

  Rectangle 1 is Larger...

>>>
```

PROGRAM NO: 2 **DATE: 05-01-2022**

AIM : 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.

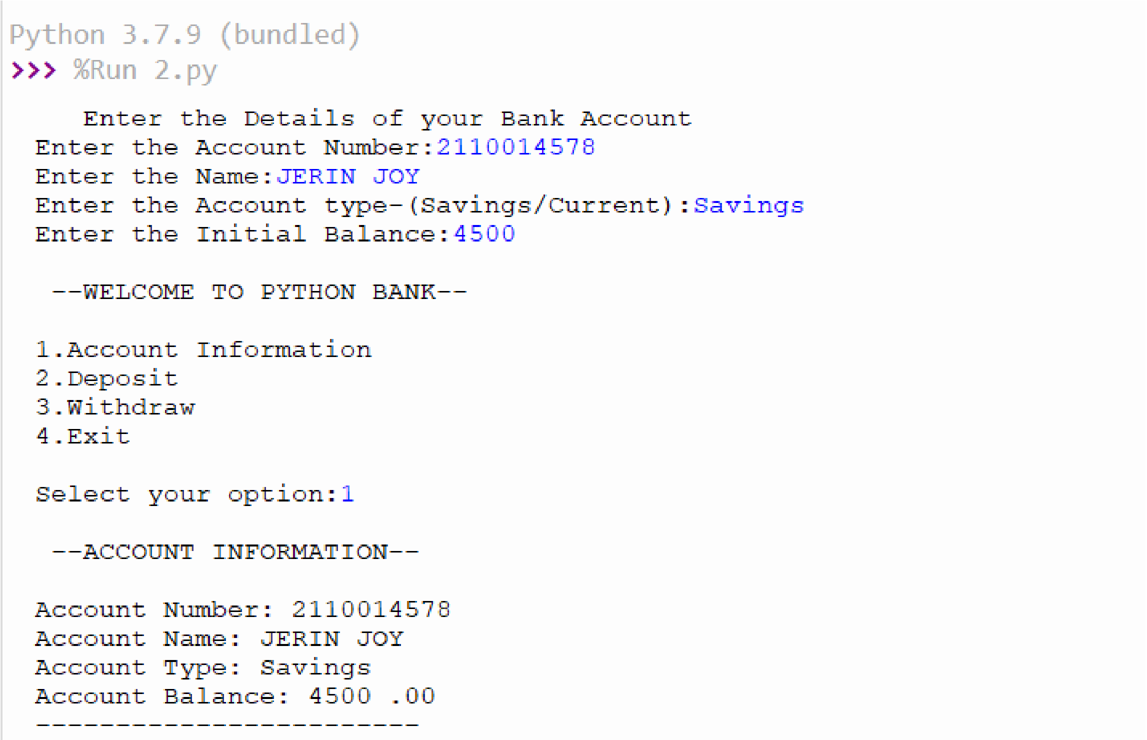
PROGRAM

```
class bank:
 balance=0
 def __init__(self,accountno,name,accounttype,balance):
  self.accountno=accountno
  self.name=name
  self.accounttype=accounttype
  self.balance=balance

 def accountinformation(self):
  print("\n --ACCOUNT INFORMATION--\n")
  print("Account Number:",self.accountno)
  print("Account Name:",self.name)
  print("Account Type:",self.accounttype)
  print("Account Balance:",self.balance,".00")
  print("-----------------------")

 def deposit(self):
  deposit=int(input("\n Enter the Amount to Deposit: "))
  print("Rs.",deposit,"Deposited Successfully...")
  print("-----------------------")
  self.balance=self.balance+deposit
```

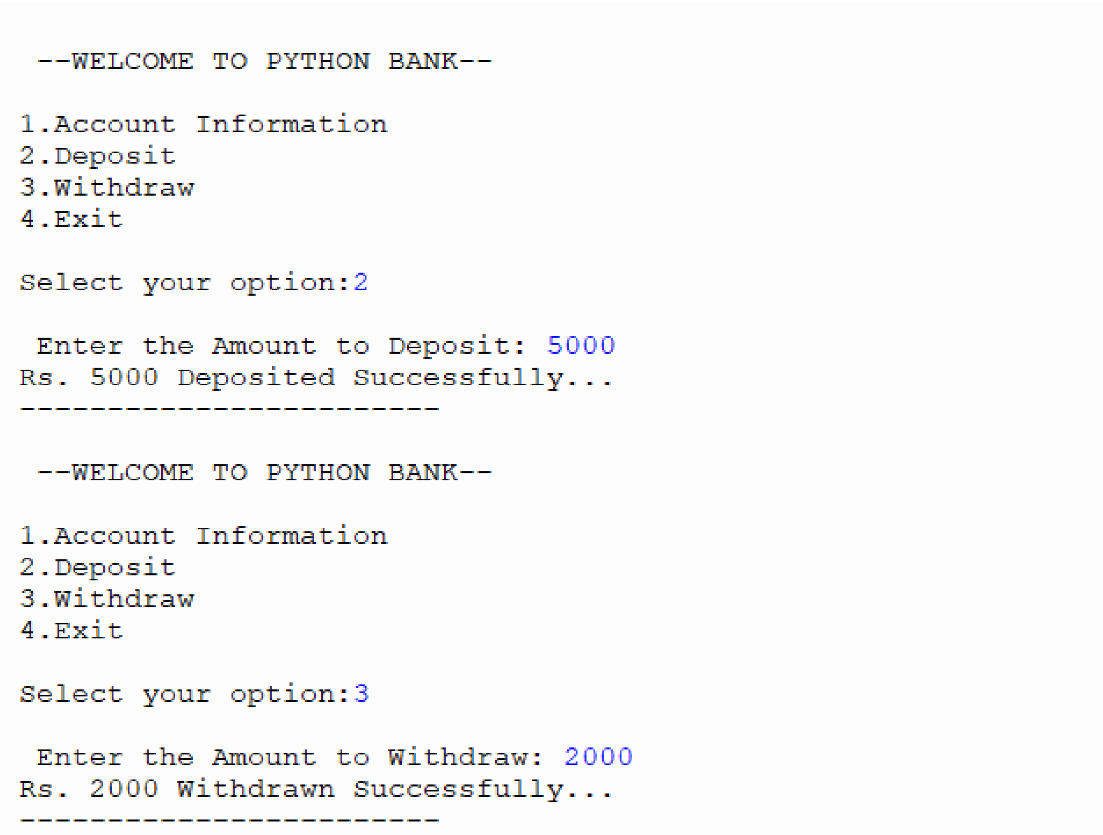
```
def withdraw(self):
  withdraw=int(input("\n Enter the Amount to Withdraw: "))
  if withdraw > self.balance:
    print("Your Account has Insufficient Balance...")
    print("------------------------")
  else:
    self.balance=self.balance-withdraw
    print("Rs.",withdraw,"Withdrawn Successfully...")
    print("------------------------")



print("   Enter the Details of your Bank Account")
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:"))

obj=bank(acc_no,acc_name,acc_type,balance)

while(1):
    print("\n --WELCOME TO PYTHON BANK--")
    print("\n1.Account Information\n2.Deposit\n3.Withdraw\n4.Exit\n")
    opt=int(input("Select your option:"))
    if opt == 1:
        obj.accountinformation()
    elif opt == 2:
        obj.deposit()
```

```
elif opt == 3:
    obj.withdraw()
 elif opt == 4:
    print("Exited")
    print("    Thank You Visit Again....\n")
    print("------------------------")
    break
 else:
    print("Invalid Option")
    print("------------------------")
```

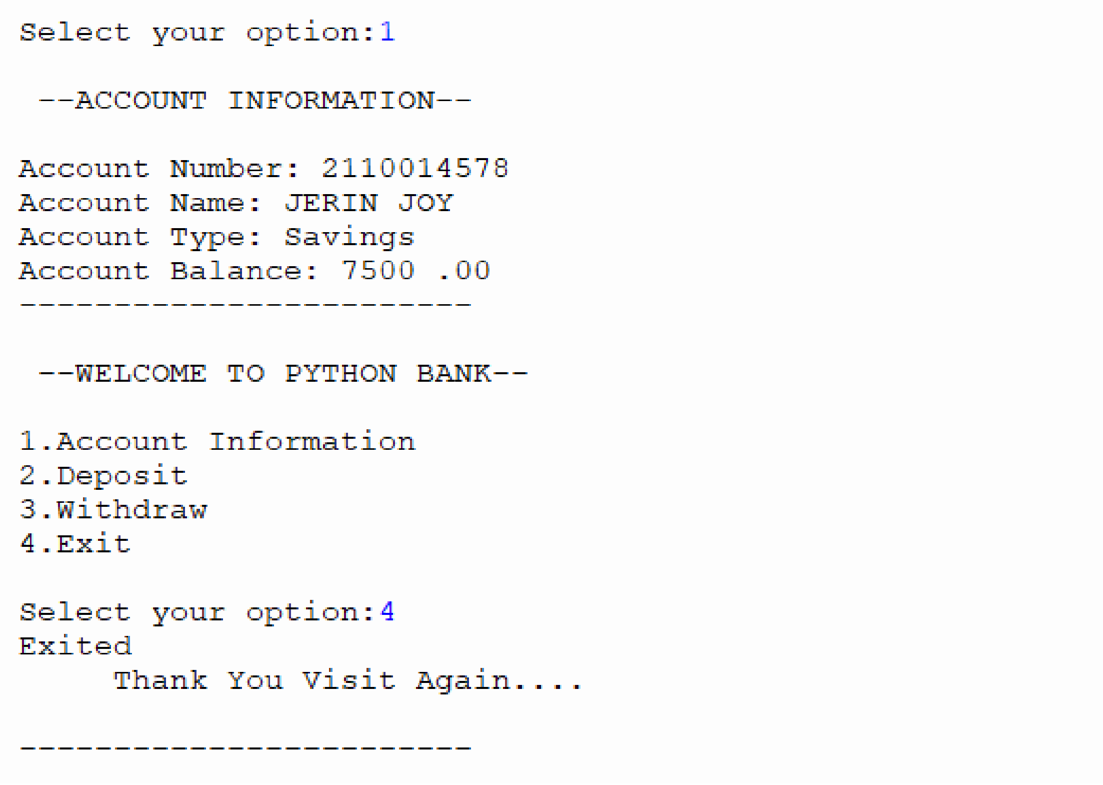
OUTPUT



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

    Enter the Details of your Bank Account
 Enter the Account Number:2110014578
 Enter the Name:JERIN JOY
 Enter the Account type-(Savings/Current):Savings
 Enter the Initial Balance:4500

  --WELCOME TO PYTHON BANK--

 1.Account Information
 2.Deposit
 3.Withdraw
 4.Exit

 Select your option:1

  --ACCOUNT INFORMATION--

 Account Number: 2110014578
 Account Name: JERIN JOY
 Account Type: Savings
 Account Balance: 4500 .00
 ------------------------
```



```
 --WELCOME TO PYTHON BANK--

1.Account Information
2.Deposit
3.Withdraw
4.Exit

Select your option:2

 Enter the Amount to Deposit: 5000
Rs. 5000 Deposited Successfully...
------------------------

 --WELCOME TO PYTHON BANK--

1.Account Information
2.Deposit
3.Withdraw
4.Exit

Select your option:3

 Enter the Amount to Withdraw: 2000
Rs. 2000 Withdrawn Successfully...
------------------------
```



```
Select your option:1

 --ACCOUNT INFORMATION--

Account Number: 2110014578
Account Name: JERIN JOY
Account Type: Savings
Account Balance: 7500 .00
------------------------

 --WELCOME TO PYTHON BANK--

1.Account Information
2.Deposit
3.Withdraw
4.Exit

Select your option:4
Exited
     Thank You Visit Again....

------------------------
```

PROGRAM NO: 3 **DATE: 05-01-2022**

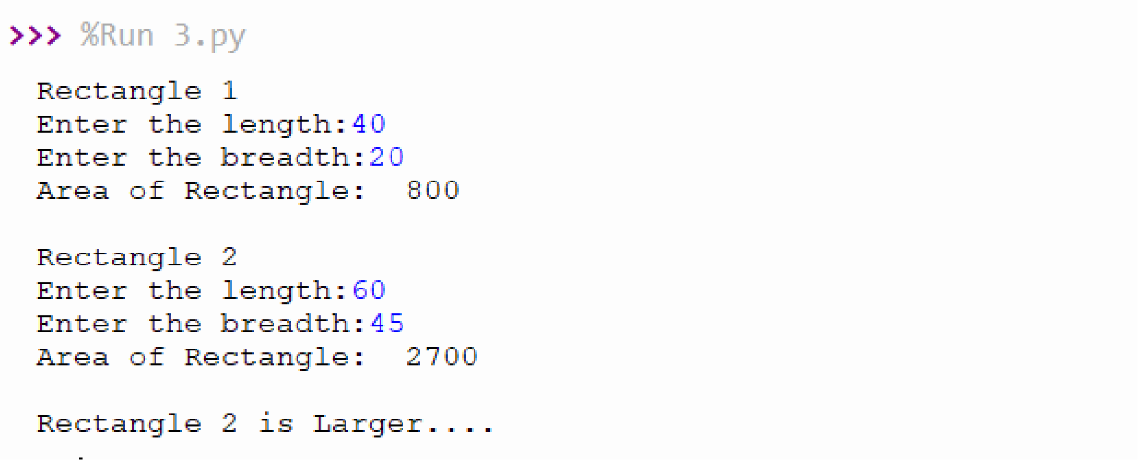
AIM : Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

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 of Rectangle: ",self.area)

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

print("Rectangle 1")
length1=int(input("Enter the length:"))
breadth1=int(input("Enter the breadth:"))
obj1=rectangle(length1,breadth1)
obj1.area()
```

```
print("\nRectangle 2")

length2=int(input("Enter the length:"))

breadth2=int(input("Enter the breadth:"))

obj2=rectangle(length2,breadth2)

obj2.area()


if obj1 > obj2 :

    print("\nRectangle 1 is Larger.....")

else:

    print("\nRectangle 2 is Larger....")
```

OUTPUT



```
>>> %Run 3.py

  Rectangle 1
  Enter the length:40
  Enter the breadth:20
  Area of Rectangle:  800

  Rectangle 2
  Enter the length:60
  Enter the breadth:45
  Area of Rectangle:  2700

  Rectangle 2 is Larger....
```

PROGRAM NO: 4 **DATE: 10-01-2022**

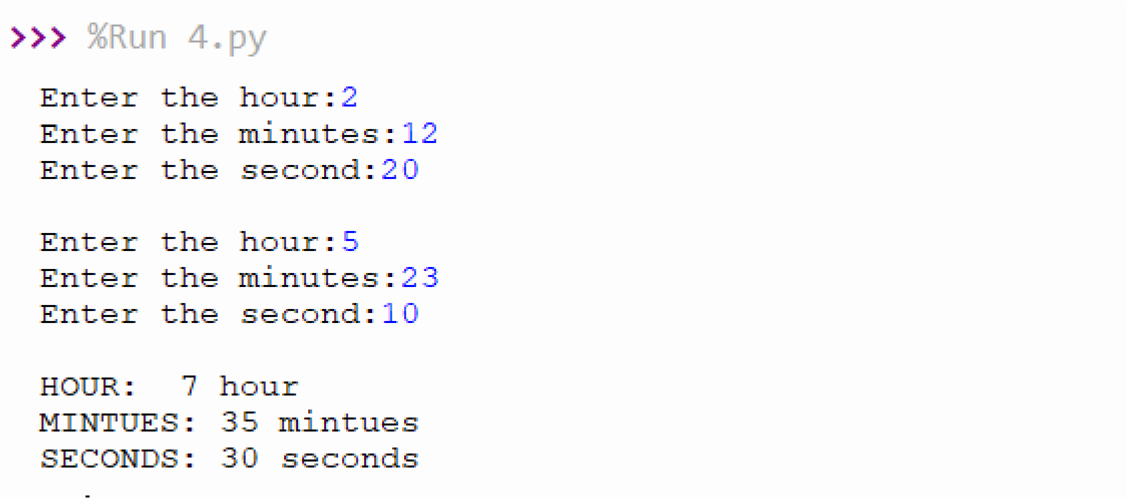
AIM : Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time

PROGRAM

```
class time:
 def __init__(self,hr,min,sec):
  self.__hour=hr
  self.__minute=min
  self.__second=sec

 def __add__(self,second):
   print("\nHOUR: ",self.__hour + second.__hour,"hour")

   if self.__minute + second.__minute > 60:
    hr1=(self.__minute + second.__minute)//60
    min1=(self.__minute + second.__minute)%60
    print("MINTUES:",h1," hour ",m1," minutes")
   else:
    print("MINTUES:",self.__minute + second.__minute,"mintues")

   if self.__second+second.__second > 60:
    min1=(self.__second+second.__second)//60
    sec1=(self.__second+second.__second)%60
    print("SECONDS:",m1," minutes ",s1," seconds")
```

```
else:
   print("SECONDS:",self.__second + second.__second,"seconds")

hour1=int(input("Enter the hour:"))
minute1=int(input("Enter the minutes:"))
second1=int(input("Enter the second:"))

obj1=time(hour1,minute1,second1)

hour2=int(input("\nEnter the hour:"))
minute2=int(input("Enter the minutes:"))
second2=int(input("Enter the second:"))

obj2=time(hour2,minute2,second2)

obj1 + obj2
```

OUTPUT



```
>>> %Run 4.py
  Enter the hour:2
  Enter the minutes:12
  Enter the second:20

  Enter the hour:5
  Enter the minutes:23
  Enter the second:10

  HOUR:   7 hour
  MINTUES: 35 mintues
  SECONDS: 30 seconds
```

PROGRAM NO: 5 **DATE: 10-01-2022**

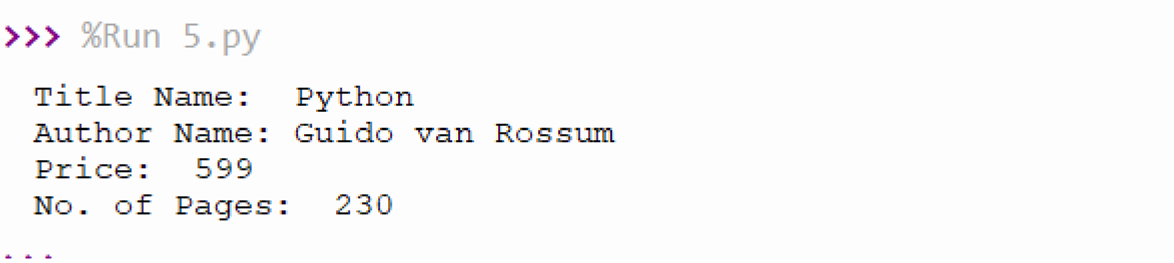
AIM : Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

PROGRAM

```
class publisher:
    def __init__ (self,pn):
        self.publishername=pn

    def display(self):
        print("Publisher Name:",self.publishername)

class book(publisher):
    def __init__ (self,pn,tt,aut):
        super(). __init__(pn)
        self.title=tt
        self.author=aut

    def display(self):
        print("Title Name: ",self.title)
        print("Author Name:",self.author)

class python(book):
    def __init__ (self,pn,tt,aut,pr,pg):
```

```
super(). __init__(pn,tt,aut)
        self.price=pr
        self.page=pg

    def pythondisplay(self):
        print("Price: ",self.price)
        print("No. of Pages: ",self.page)

obj=python("joy publishers","Python","Guido van Rossum",599,230);
obj.display()
obj.pythondisplay();
```

OUTPUT



```
>>> %Run 5.py
 Title Name:  Python
 Author Name: Guido van Rossum
 Price:  599
 No. of Pages:  230
...
```

COURSE OUTCOME 5 (CO5)

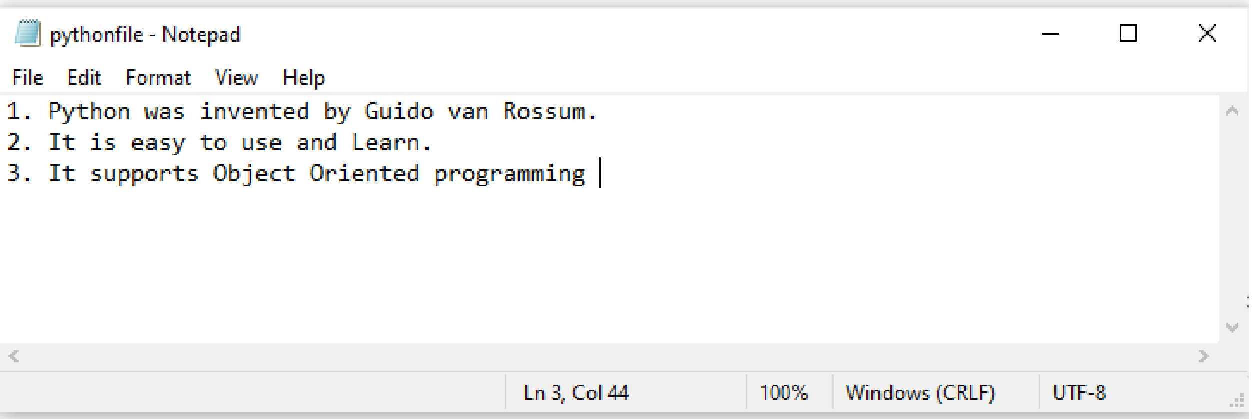
PROGRAM NO: 1 **DATE: 17-01-2022**

AIM : Write a program to read a file line by line and store it into a list.

PROGRAM

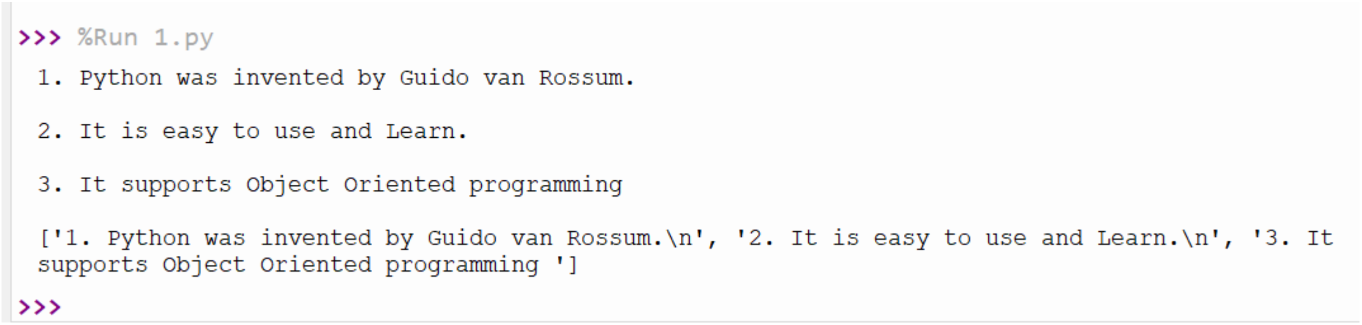
```
file=open("pythonfile.txt","w")
file.write("1. Python was invented by Guido van Rossum.\n2. It is easy to use and Learn.\n3. It supports Object Oriented programming ")
file.close()


file=open("pythonfile.txt","r") #("filename","mode of file")(there are 6 mode)
file.seek(0,0)
ff=file.readlines()


for x in range(0,len(ff)):
    print(ff[x])
print()
print(ff)
file.close()
```

pythonfile.txt



```
1. Python was invented by Guido van Rossum.
2. It is easy to use and Learn.
3. It supports Object Oriented programming 
```

OUTPUT



```
>>> %Run 1.py
 1. Python was invented by Guido van Rossum.

 2. It is easy to use and Learn.

 3. It supports Object Oriented programming

 ['1. Python was invented by Guido van Rossum.\n', '2. It is easy to use and Learn.\n', '3. It
 supports Object Oriented programming ']
>>>
```

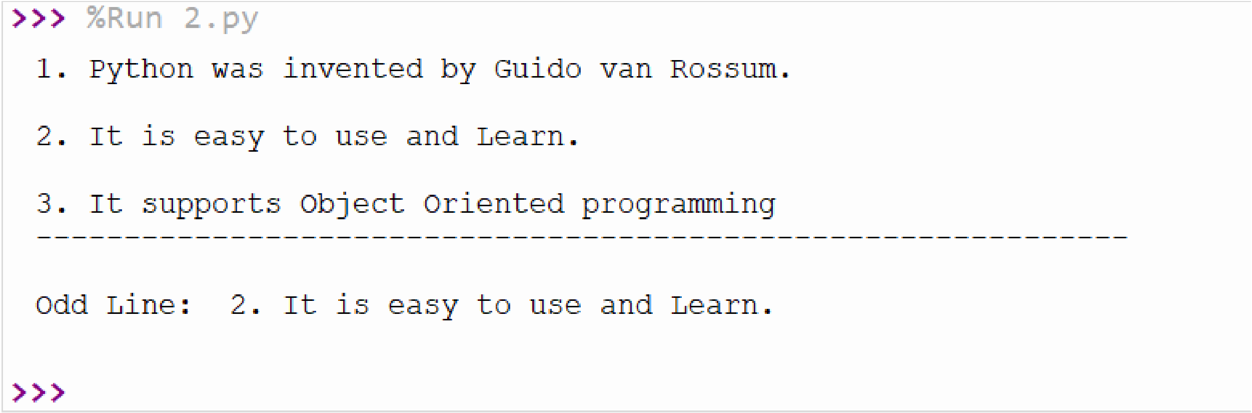
PROGRAM NO: 2 **DATE: 17-01-2022**

AIM : Write a program to copy odd lines of one file to other.

PROGRAM

```
file1=open("pythonfile.txt","r")

for x in file1:

    print(x)

file1.seek(0,0)

print("------------------------------------------------------------")

print()


print("Odd Line: ",end=" ")

file2=open("odd.txt","w")

ff=file1.readlines()

with open('odd.txt','w') as file2:

     for x in range(0,len(ff)):

        if(x%2!=0):

            print(ff[x])

            file2.write(ff[x])
```

OUTPUT



```
>>> %Run 2.py
  1. Python was invented by Guido van Rossum.

  2. It is easy to use and Learn.

  3. It supports Object Oriented programming
  --------------------------------------------------------------

  Odd Line:   2. It is easy to use and Learn.

>>>
```

PROGRAM NO: 3 **DATE: 31-01-2022**

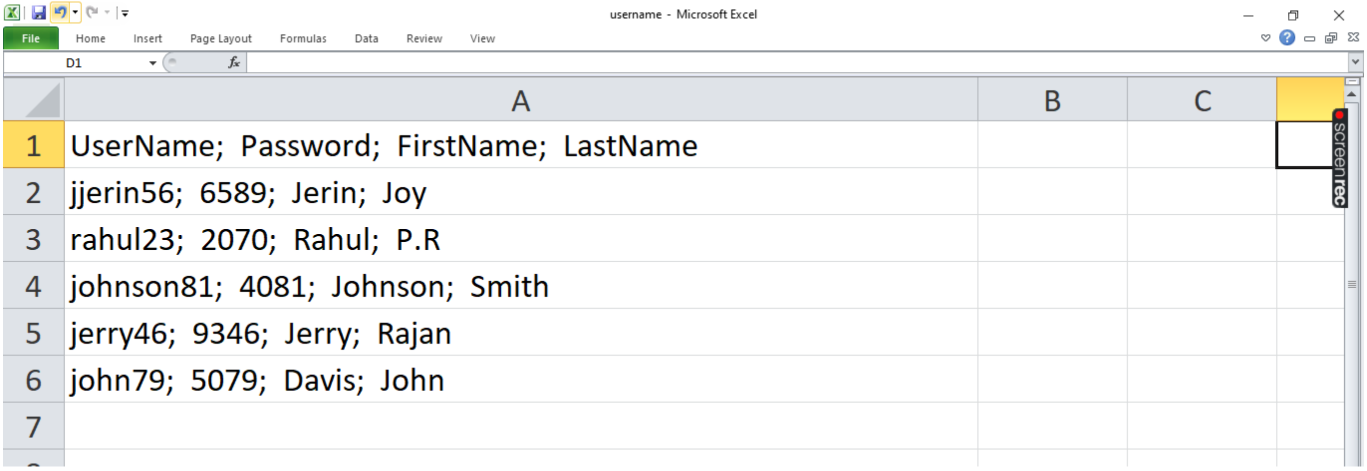
AIM : Write a Python program to read each row from a given csv file and print a list of strings.

PROGRAM

```
import csv


filename = "username.csv"


fields = []

rows = []


ff=open(filename, 'r')

csvreader = csv.reader(ff)

fields = next(ff)

print(fields)

for r in csvreader:

   rows.append(r)

print(rows)

print("--------------------------------------------------------------")

print('\nFirst 4 Rows are: \n')

for r in rows[:4]:

   print(*r)

print("--------------------------------------------------------------")

print()

print("The File Content")

print()
```

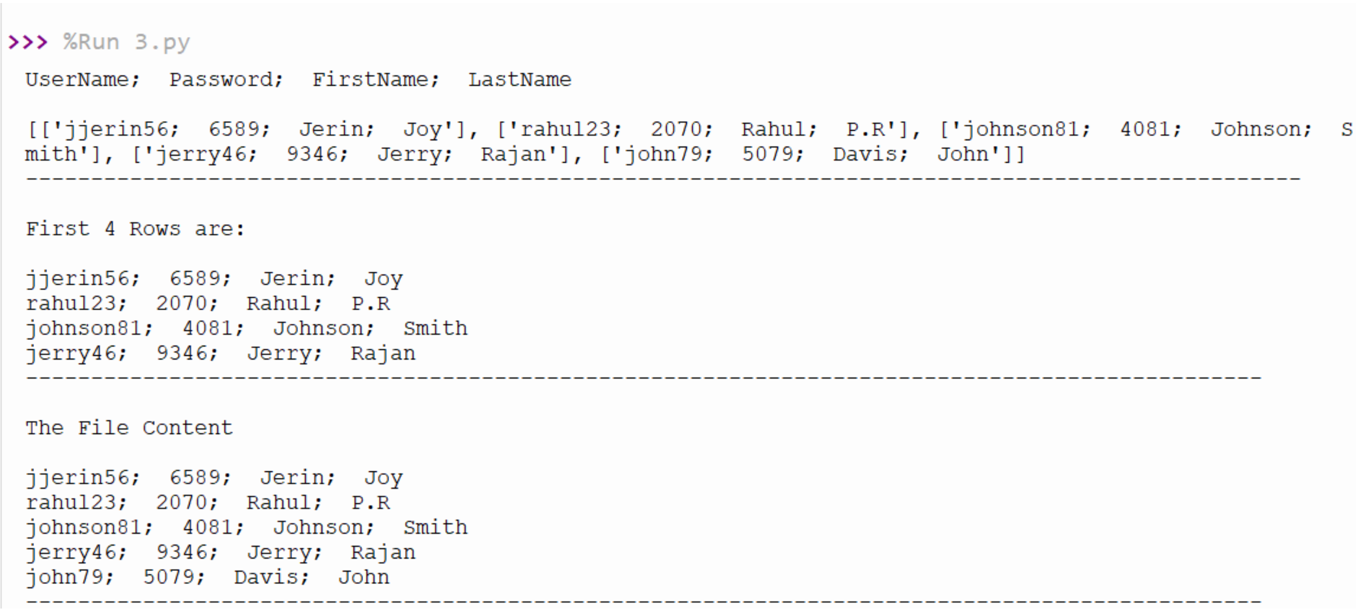
```
for xy in rows:
    for z in xy:
        print(z)
print("-------------------------------------------------------------")
print()
 #print(z,end=" ")
print()
ff.close()
```

username.csv



	A	B	C
1	UserName; Password; FirstName; LastName		
2	jjerin56; 6589; Jerin; Joy		
3	rahul23; 2070; Rahul; P.R		
4	johnson81; 4081; Johnson; Smith		
5	jerry46; 9346; Jerry; Rajan		
6	john79; 5079; Davis; John		
7			

OUTPUT



```
>>> %Run 3.py
 UserName;  Password;  FirstName;  LastName

 [['jjerin56;  6589;  Jerin;  Joy'], ['rahul23;  2070;  Rahul;  P.R'], ['johnson81;  4081;  Johnson;  S
 mith'], ['jerry46;  9346;  Jerry;  Rajan'], ['john79;  5079;  Davis;  John']]
 -------------------------------------------------------------------------------------------------

 First 4 Rows are:

 jjerin56;  6589;  Jerin;  Joy
 rahul23;  2070;  Rahul;  P.R
 johnson81;  4081;  Johnson;  Smith
 jerry46;  9346;  Jerry;  Rajan
 ----------------------------------------------------------------------------------------------

 The File Content

 jjerin56;  6589;  Jerin;  Joy
 rahul23;  2070;  Rahul;  P.R
 johnson81;  4081;  Johnson;  Smith
 jerry46;  9346;  Jerry;  Rajan
 john79;  5079;  Davis;  John
 ----------------------------------------------------------------------------------------------
```

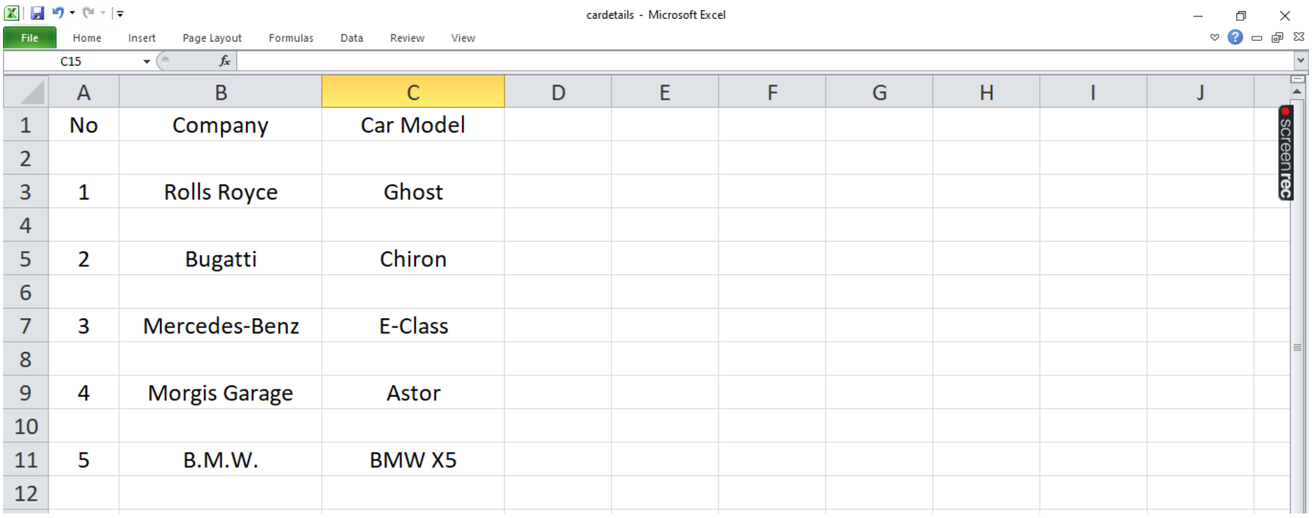
PROGRAM NO: 4 **DATE: 31-01-2022**

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

PROGRAM

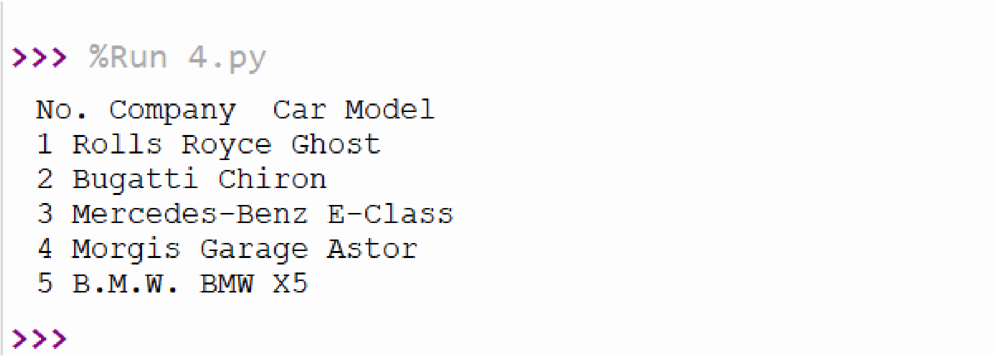
```
import csv
filename = "cardetails.csv"
ff=open(filename, 'r')
 #csvreader = csv.reader(ff)
data = csv.DictReader(ff)
print("No. Company  Car Model")
for x in data:
    print(x['No'], x['Company'], x['Car Model'])
```

cardetails.csv



	A	B	C	D	E	F	G	H	I	J
1	No	Company	Car Model							
2										
3	1	Rolls Royce	Ghost							
4										
5	2	Bugatti	Chiron							
6										
7	3	Mercedes-Benz	E-Class							
8										
9	4	Morgis Garage	Astor							
10										
11	5	B.M.W.	BMW X5							
12										

OUTPUT



```
>>> %Run 4.py
 No. Company  Car Model
 1 Rolls Royce Ghost
 2 Bugatti Chiron
 3 Mercedes-Benz E-Class
 4 Morgis Garage Astor
 5 B.M.W. BMW X5
>>>
```

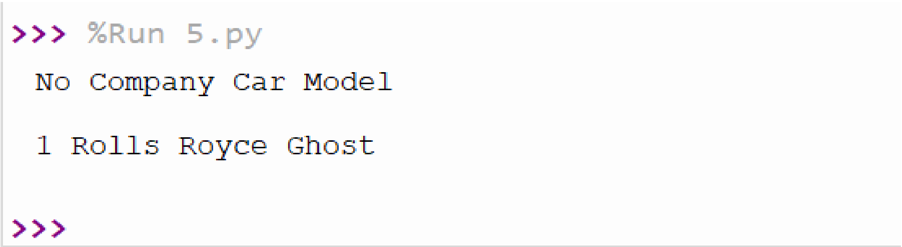
PROGRAM NO: 5 **DATE: 31-01-2022**

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': 'Rolls Royce', 'Car Model': 'Ghost'},
{'No': 2, 'Company': 'Bugatti', 'Car Model': 'Chiron'},
{'No': 3, 'Company': 'Mercedes-Benz', 'Car Model': 'E-Class'},
{'No': 4, 'Company': 'Morgis Garage', 'Car Model': 'Astor'},
{'No': 5, 'Company': 'B.M.W.', 'Car Model': 'BMW X5'},
 ]
with open("cardetails.csv", "w") as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_names)
    writer.writeheader()
    writer.writerows(cars)#print(".................")
filename = "cardetails.csv"
ff=open(filename, 'r')
rows=[]
csvreader = csv.reader(ff)
for r in csvreader:
    rows.append(r)
for r in rows[:4]:
    print(*r)
```

OUTPUT



```
>>> %Run 5.py
 No Company Car Model

 1 Rolls Royce Ghost

>>>
```