SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-2023

20 MCA 132 PROGRAMMING LABORATORY RECORD

Submitted by

MOHAMMED FAYAZ ISMAIL P K

REG NO : SNG21MCA-2020

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 ***MOHAMMED FAYAZ ISMAIL P K*** *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-2020 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.

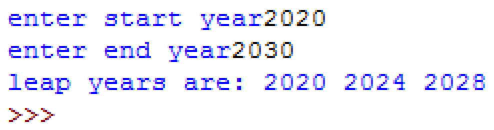
PROGRAM NO: 2 **DATE : 08/11/2021**

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

PROGRAM

```
s=int(input("enter start year"))
e=int(input("enter end year"))
if(s<e):
        print("leap years are:",end=" ")
        for i in range(s,e):
            if (i%4==0) and (i % 100 !=0) or (i%400==0) and (i%100==0):
                        print(i,end=" ")
```

OUTPUT



```
enter start year2020
enter end year2030
leap years are: 2020 2024 2028
>>>
```

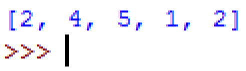
PROGRAM NO: 3 **DATE : 10/11/2021**

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

PROGRAM

```
list=[2,-6,4,-8,5,-7,1,2]

n=[num for num in list if num>0]

print(n)
```

OUTPUT



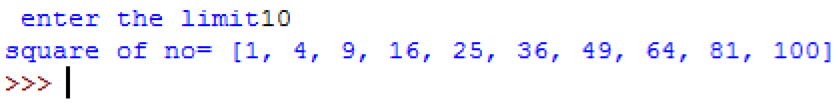
```
[2, 4, 5, 1, 2]
>>> 
```

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

PROGRAM

```
n=int(input(" enter the limit"))
zlist=[i**2 for i in range(1,n+1)]
print("square of no=",zlist)
```

OUTPUT



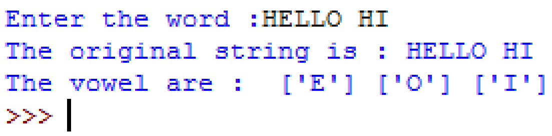
```
 enter the limit10
square of no= [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
>>> 
```

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

PROGRAM

```
word =str(input("Enter the word :"))
print("The original string is : "+word)
print("The vowel are : ",end=" ")
for i in word:
        if i in 'aeiouAEIOU':
                print([i],end=" ")
```

OUTPUT



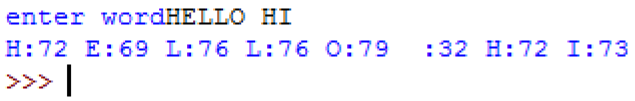
```
Enter the word :HELLO HI
The original string is : HELLO HI
The vowel are :  ['E'] ['O'] ['I']
>>> 
```

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

PROGRAM

```
y=input("enter word")
for i in y:
    print(i,end=":")
    print(ord(i),end=" "))
```

OUTPUT



```
enter wordHELLO HI
H:72 E:69 L:76 L:76 O:79  :32 H:72 I:73
>>> 
```

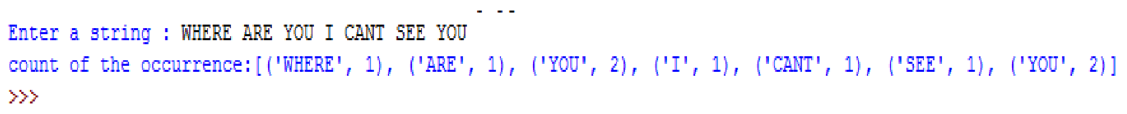
PROGRAM NO: 4 **DATE : 15/11/2021**

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



```
Enter a string : WHERE ARE YOU I CANT SEE YOU
count of the occurrence:[('WHERE', 1), ('ARE', 1), ('YOU', 2), ('I', 1), ('CANT', 1), ('SEE', 1), ('YOU', 2)]
>>>
```

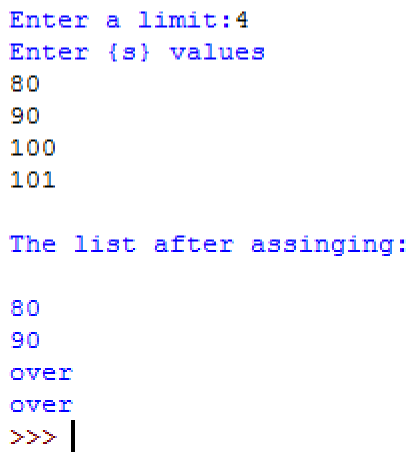
PROGRAM NO: 5 **DATE : 15/11/2021**

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 {s} values")
for i in range(0,s):
        n.append(int(input()))
print("\nThe list after assinging:\n")
for i in range(0,len(n)):
        if n[i]>=100:
                print("over")
        else:
                print(n[i])
```

OUTPUT



```
Enter a limit:4
Enter {s} values
80
90
100
101

The list after assinging:

80
90
over
over
>>> 
```

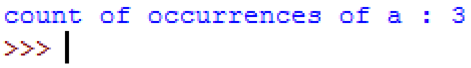
PROGRAM NO: 6 **DATE : 17/11/2021**

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

PROGRAM

```
a_list = ["a", "b", "a","c","d","a"]
occ = a_list.count("a")
print("count of occurrences of a :",occ)
```

OUTPUT



```
count of occurrences of a : 3
>>> 
```

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

AIM : Enter 2 lists of integers. Check

(a) Whether list are of same length.

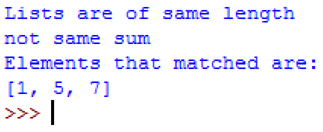
(b) whether list sums to same value.

(c) whether any value occur in both.

PROGRAM

```
lst=[1,3,5,7,9,11,34]
lst1=[5,13,45,7,20,65,1]
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



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

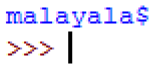
PROGRAM NO: 8 **DATE : 22/11/2021**

AIM : Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.

PROGRAM

```
str1="malayalam"
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print(str1)
```

OUTPUT



```
malayala$
>>> |
```

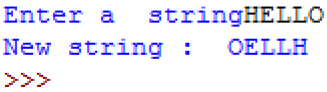
PROGRAM NO: 9 **DATE : 22/11/2021**

AIM : Create a string from given string where first and last characters exchanged.

PROGRAM

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

OUTPUT



```
Enter a  stringHELLO
New string :  OELLH
>>>
```

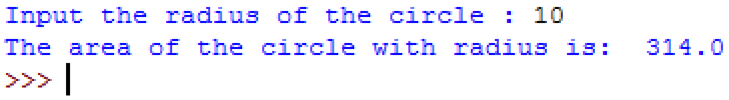
PROGRAM NO: 10 **DATE : 24/11/2021**

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

PROGRAM

```
pi=3.14
r=float(input("Input the radius of the circle:"))
result=3.14*r**2
print("The area of the circle with radius is:",result)
```

OUTPUT



```
Input the radius of the circle : 10
The area of the circle with radius is:  314.0
>>> 
```

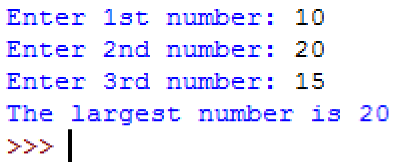
PROGRAM NO: 11 **DATE : 24/11/2021**

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



```
Enter 1st number: 10
Enter 2nd number: 20
Enter 3rd number: 15
The largest number is 20
>>> 
```

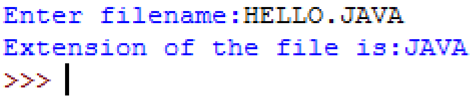
PROGRAM NO: 12 **DATE : 24/11/2021**

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



```
Enter filename:HELLO.JAVA
Extension of the file is:JAVA
>>> 
```

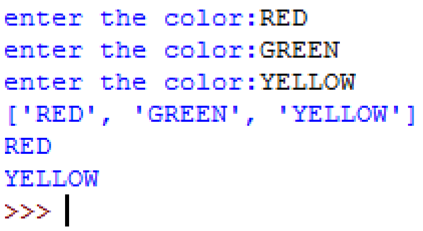
PROGRAM NO: 13 **DATE : 29/11/2021**

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

OUTPUT



```
enter the color:RED
enter the color:GREEN
enter the color:YELLOW
['RED', 'GREEN', 'YELLOW']
RED
YELLOW
>>> 
```

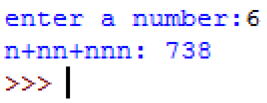
PROGRAM NO: 14 **DATE : 29/11/2021**

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



```
enter a number:6
n+nn+nnn: 738
>>> 
```

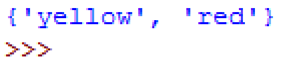
PROGRAM NO: 15 **DATE : 29/11/2021**

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

PROGRAM

```
li1=set(["red","yellow","green"])
li2=set(["green","violet","pink"])
print(li1.difference(li2))
```

OUTPUT



```
{'yellow', 'red'}
>>>
```

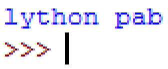
PROGRAM NO: 16 **DATE : 29/11/2021**

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

PROGRAM

```
a="python"
b="lab"
p1=a[0]
p2=b[0]
c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]
print(c)
```

OUTPUT



```
lython pab
>>> 
```

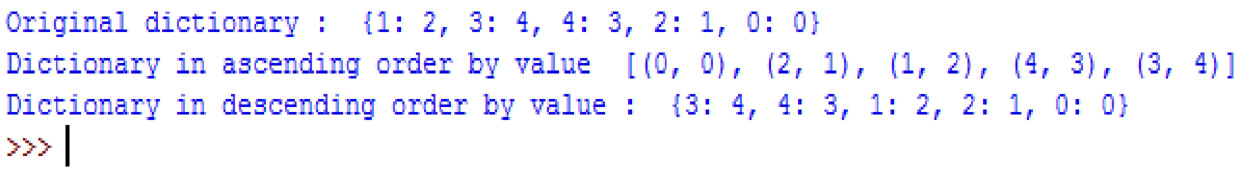
PROGRAM NO: 17 **DATE : 01/12/2021**

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)
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, 2), (4, 3), (3, 4)]
Dictionary in descending order by value :  {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
>>> 
```

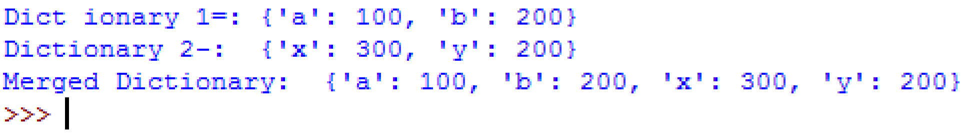
PROGRAM NO: 18 **DATE : 01/12/2021**

AIM : Write a program to merge two dictionaries.

PROGRAM

```
d1 ={ 'a': 100, 'b': 200}
d2 ={'x' : 300, 'y': 200}
print ("Dict ionary 1=:", d1)
print ("Dictionary 2-: ", d2)
d =d1. copy ()
d.update (d2)
print ("Merged Dictionary: ", d))
```

OUTPUT



```
Dict ionary 1=: {'a': 100, 'b': 200}
Dictionary 2-:  {'x': 300, 'y': 200}
Merged Dictionary:  {'a': 100, 'b': 200, 'x': 300, 'y': 200}
>>> |
```

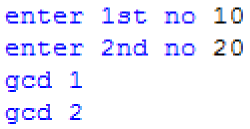
PROGRAM NO: 19 **DATE : 01/12/2021**

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

PROGRAM

```
x=int(input("enter 1st no"))
y=int(input("enter 2nd no"))
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



```
enter 1st no 10
enter 2nd no 20
gcd 1
gcd 2
```

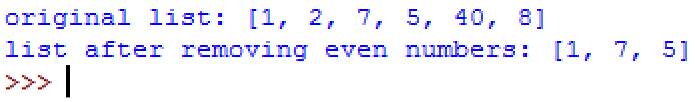
PROGRAM NO: 20 **DATE : 01/12/2021**

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

PROGRAM

```
num=[1,2,7,5,40,8]
print("original list:",num)
num=[x for x in num if x%2!=0]
print("list after removing even numbers:",num)
```

OUTPUT



```
original list: [1, 2, 7, 5, 40, 8]
list after removing even numbers: [1, 7, 5]
>>> 
```

COURSE OUTCOME 2 (CO2)

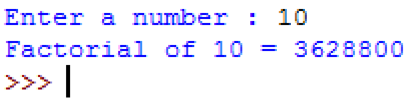
PROGRAM NO: 1 **DATE : 06/12/2021**

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



```
Enter a number : 10
Factorial of 10 = 3628800
>>> 
```

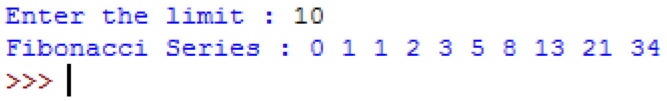
PROGRAM NO: 2 **DATE : 06/12/2021**

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 = count+1
    a = b
    b = sum
    sum = a + b
```

OUTPUT



```
Enter the limit : 10
Fibonacci Series : 0 1 1 2 3 5 8 13 21 34
>>> 
```

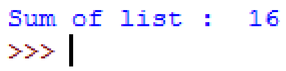
PROGRAM NO: 3 **DATE : 06/12/2021**

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

PROGRAM

```
list1 = [1, 5, 2, 6,2 ]
total = sum(list1)
print("Sum of list : ",total)
```

OUTPUT



```
Sum of list :  16
>>> 
```

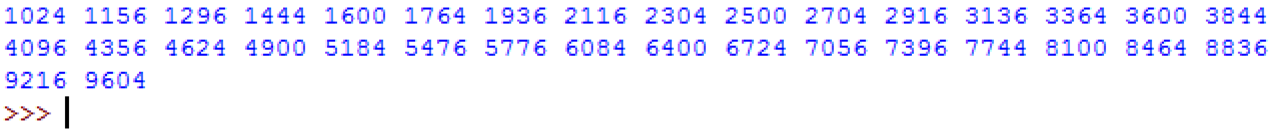
PROGRAM NO: 4 **DATE : 06/12/2021**

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(1000,10000):
   if s(i)==int(s(i)) and i%2==0:
      print(i,end=" ")
```

OUTPUT



```
1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 3136 3364 3600 3844
4096 4356 4624 4900 5184 5476 5776 6084 6400 6724 7056 7396 7744 8100 8464 8836
9216 9604
>>> 
```

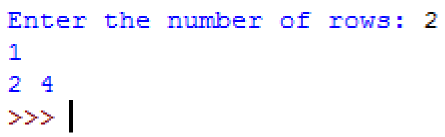
PROGRAM NO: 5 **DATE : 06/12/2021**

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):
    for j in range(1,i+1):
        print(i * j, end=' ')
    print()
```

OUTPUT



```
Enter the number of rows: 2
1
2 4
>>> 
```

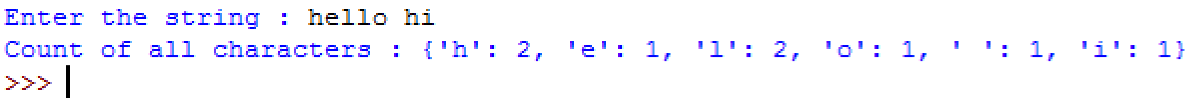
PROGRAM NO: 6 **DATE : 06/12/2021**

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



```
Enter the string : hello hi
Count of all characters : {'h': 2, 'e': 1, 'l': 2, 'o': 1, ' ': 1, 'i': 1}
>>> 
```

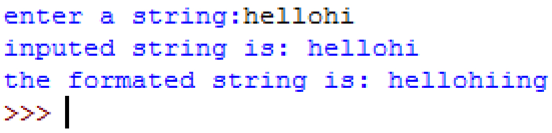
PROGRAM NO: 7 **DATE : 08/12/2021**

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



```
enter a string:hellohi
inputed string is: hellohi
the formated string is: hellohiing
>>> 
```

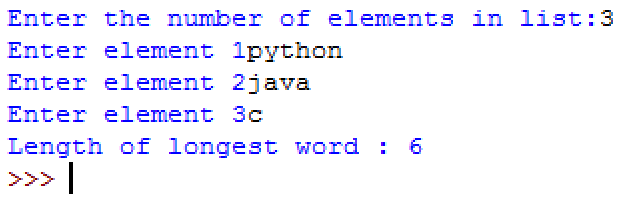
PROGRAM NO: 8 **DATE : 08/12/2021**

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



```
Enter the number of elements in list:3
Enter element 1python
Enter element 2java
Enter element 3c
Length of longest word : 6
>>> 
```

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

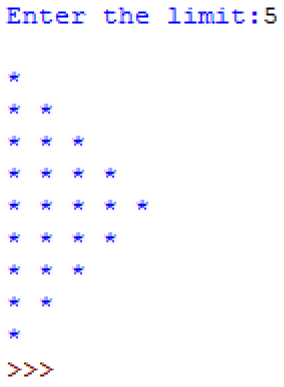
AIM : Construct following pattern using nested loop

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

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



```
Enter the limit:5

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
>>>
```

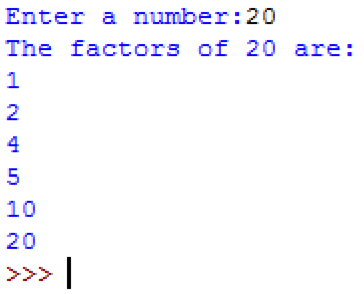
PROGRAM NO: 10 **DATE : 08/12/2021**

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



```
Enter a number:20
The factors of 20 are:
1
2
4
5
10
20
>>> 
```

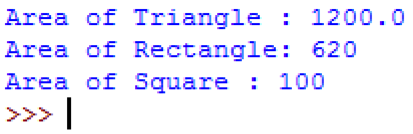
PROGRAM NO: 11 **DATE : 08/12/2021**

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

PROGRAM

```
import math
t_area = lambda b,h :1/2*b*h
r_area = lambda l,b :l*b
s_area = lambda a : a*a
print("Area of Triangle :", t_area(120,20))
print("Area of Rectangle:", r_area(31,20))
print("Area of Square :", s_area(10))
```

OUTPUT



```
Area of Triangle : 1200.0
Area of Rectangle: 620
Area of Square : 100
>>> 
```

COURSE OUTCOME 3 (CO3)

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

AIM : Work with built-in packages.

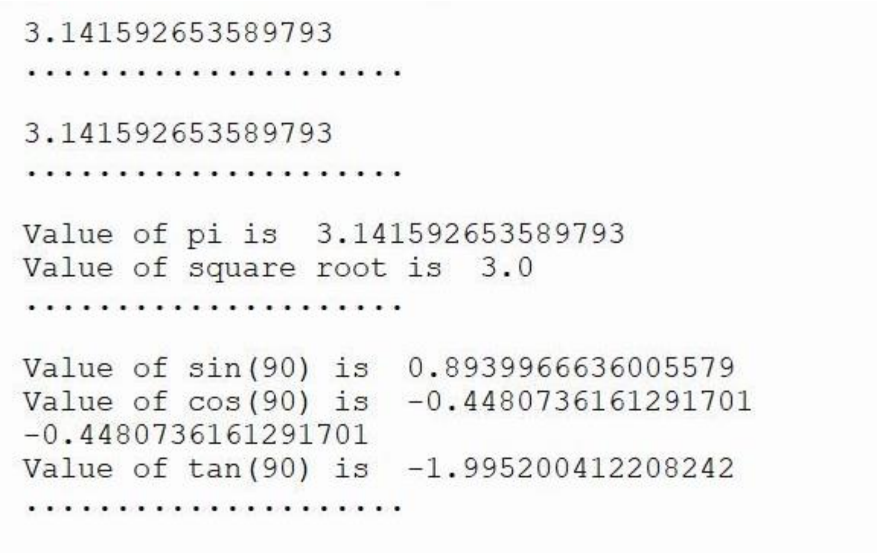
- **A) Module math**
- **B) Module time**
- **C) Module calendar**
- **D) Module random**
- **E) Module statistics**

A) Module math

PROGRAM

```
import math
print(math.pi)
print("....................\n")

import math as m
print(m.pi)
print("....................\n")

from math import pi,sqrt
print("Value of pi is ",pi)
print("Value of square root is ",sqrt(9))
print("....................\n")

from math import sin,cos,tan
print("Value of sin(90) is ",sin(90))
print("Value of cos(90) is ",cos(90))
print(math.cos(90))
print("Value of tan(90) is ",tan(90))
print("....................\n")
```

OUTPUT



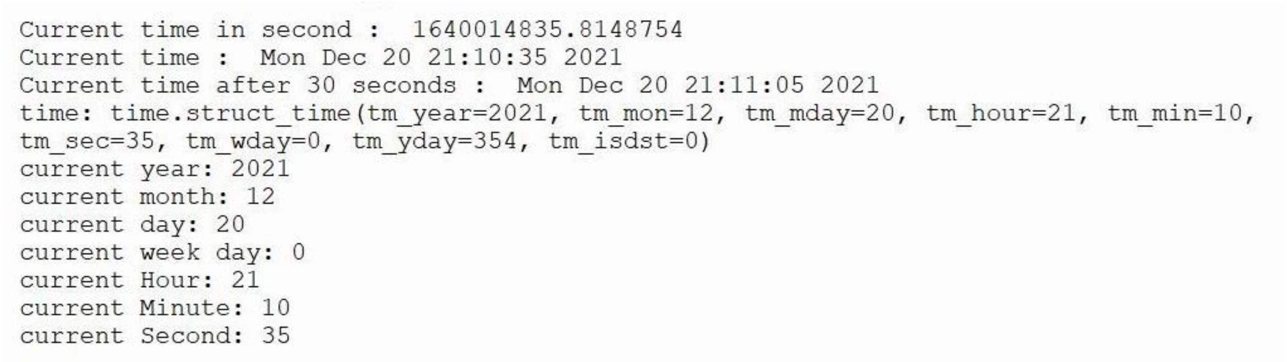
```
3.141592653589793
.....................

3.141592653589793
.....................

Value of pi is  3.141592653589793
Value of square root is  3.0
.....................

Value of sin(90) is  0.8939966636005579
Value of cos(90) is  -0.4480736161291701
-0.4480736161291701
Value of tan(90) is  -1.995200412208242
.....................
```

B) Module time

PROGRAM

```
import time
print("Current time in second : ",time.time())
print("Current time : ",time.ctime())
print("Current time after 30 seconds : ",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 week day:",t.tm_wday)
print("current Hour:",t.tm_hour)
print("current Minute:",t.tm_min)
print("current Second:",t.tm_sec)
```

OUTPUT



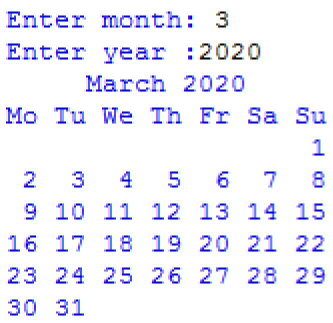
```
Current time in second :  1640014835.8148754
Current time :  Mon Dec 20 21:10:35 2021
Current time after 30 seconds :  Mon Dec 20 21:11:05 2021
time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=21, tm_min=10,
tm_sec=35, tm_wday=0, tm_yday=354, tm_isdst=0)
current year: 2021
current month: 12
current day: 20
current week day: 0
current Hour: 21
current Minute: 10
current Second: 35
```

C) Module calendar

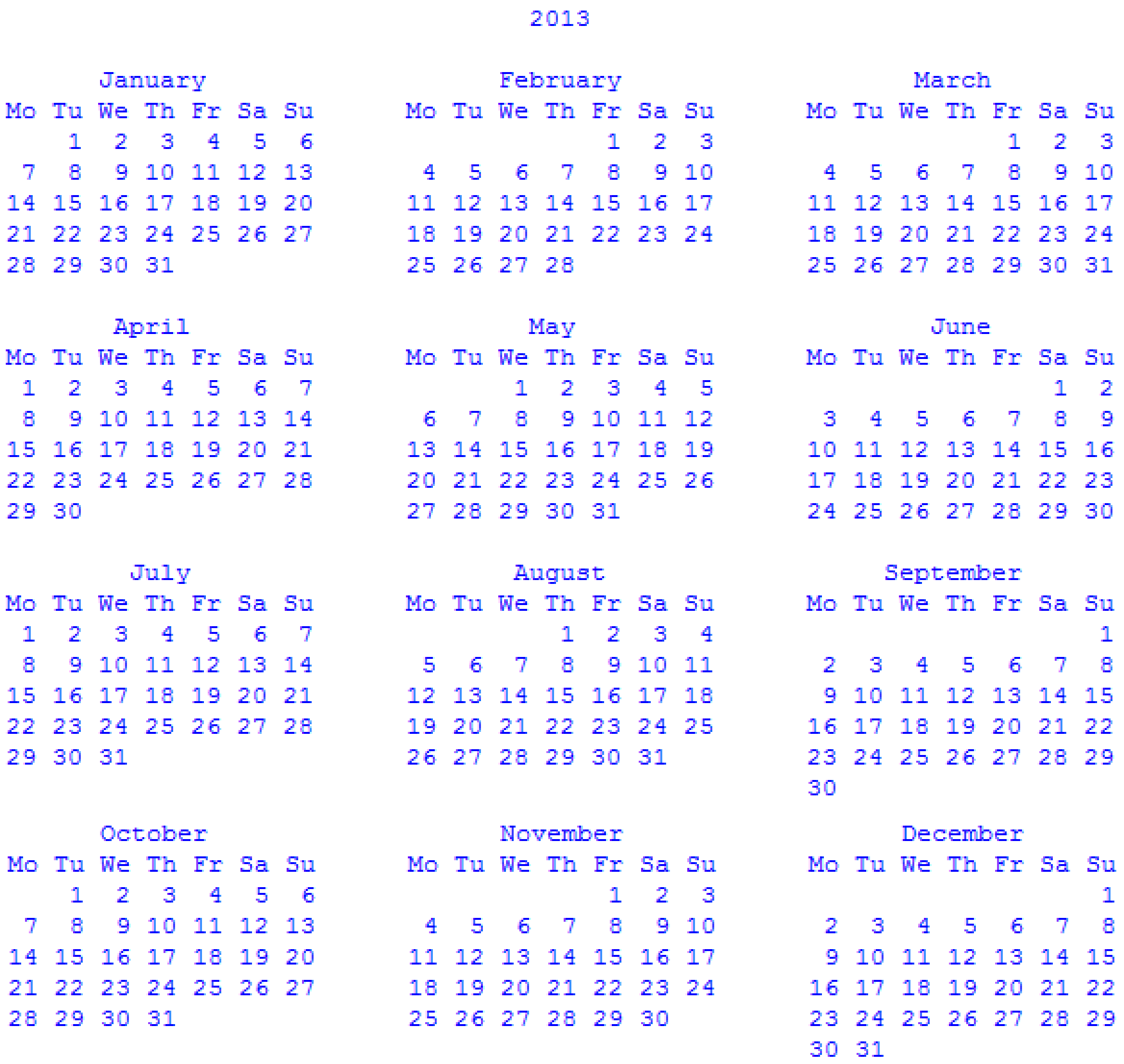
PROGRAM

```
import calendar
mm = int(input("Enter month: "))
yy = int(input("Enter year :"))
print(calendar.month(yy,mm))
print(calendar.calendar(2015))
```

OUTPUT



```
Enter month: 3
Enter year :2020
     March 2020
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
```



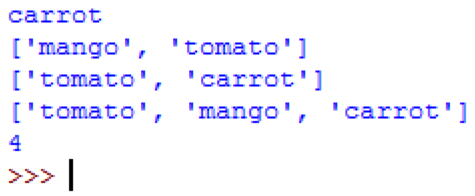
```
                                  2013

      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  4  5  6                   1  2  3                   1  2  3
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       4  5  6  7  8  9 10
14 15 16 17 18 19 20      11 12 13 14 15 16 17      11 12 13 14 15 16 17
21 22 23 24 25 26 27      18 19 20 21 22 23 24      18 19 20 21 22 23 24
28 29 30 31               25 26 27 28               25 26 27 28 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  5  6  7             1  2  3  4  5                      1  2
 8  9 10 11 12 13 14       6  7  8  9 10 11 12       3  4  5  6  7  8  9
15 16 17 18 19 20 21      13 14 15 16 17 18 19      10 11 12 13 14 15 16
22 23 24 25 26 27 28      20 21 22 23 24 25 26      17 18 19 20 21 22 23
29 30                     27 28 29 30 31            24 25 26 27 28 29 30

        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  5  6  7                1  2  3  4                         1
 8  9 10 11 12 13 14       5  6  7  8  9 10 11       2  3  4  5  6  7  8
15 16 17 18 19 20 21      12 13 14 15 16 17 18       9 10 11 12 13 14 15
22 23 24 25 26 27 28      19 20 21 22 23 24 25      16 17 18 19 20 21 22
29 30 31                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

      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  4  5  6                   1  2  3                         1
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       2  3  4  5  6  7  8
14 15 16 17 18 19 20      11 12 13 14 15 16 17       9 10 11 12 13 14 15
21 22 23 24 25 26 27      18 19 20 21 22 23 24      16 17 18 19 20 21 22
28 29 30 31               25 26 27 28 29 30         23 24 25 26 27 28 29
                                                    30 31
```

D) Module random

PROGRAM

```
import random
mylist = ["carrot", "tomato", "mango"]
print(random.choice(mylist))
print(random.choices(mylist, k=2))
print(random.sample(mylist, k=2))
random.shuffle(mylist)
print(mylist)
print(random.randrange(3, 9))
```

OUTPUT



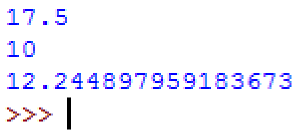
```
carrot
['mango', 'tomato']
['tomato', 'carrot']
['tomato', 'mango', 'carrot']
4
>>> |
```

E) Module statistics

PROGRAM

```
import statistics
print(statistics.mean([5,10,15,20,25,30]))
print(statistics.median([5,10,15]))
print(statistics.harmonic_mean([5,10,15,20,25,30]))
```

OUTPUT



```
17.5
10
12.244897959183673
>>> 
```

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

Package graphics

(1) __init .py

(2) circle.py

```
def perimeter(r):
    print ("Perimeter of the circle : ",2*3.14*r)

def area(r):
    print ("Area of the circle : ",3.14*r*r)
```

(3) rectangle.py

```
def perimeter(l,b):
    print ("Perimeter of the rectangle : ",2*(l+b))

def area(l,b):
     print ("Area of the rectangle : ",l*b)
```

(4) Subpackage ThreeDgraphics

i. __init .py

ii. cuboid.py

```
def perimeter(l,b,h):

    print ("Perimeter of the cuboid : ",4*(l+b+h))
```

```
def area(l,b,h):
    print ("Area of the cuboid : ",2*l*b+2*l*h+2*h*b)
```

iii. sphere.py

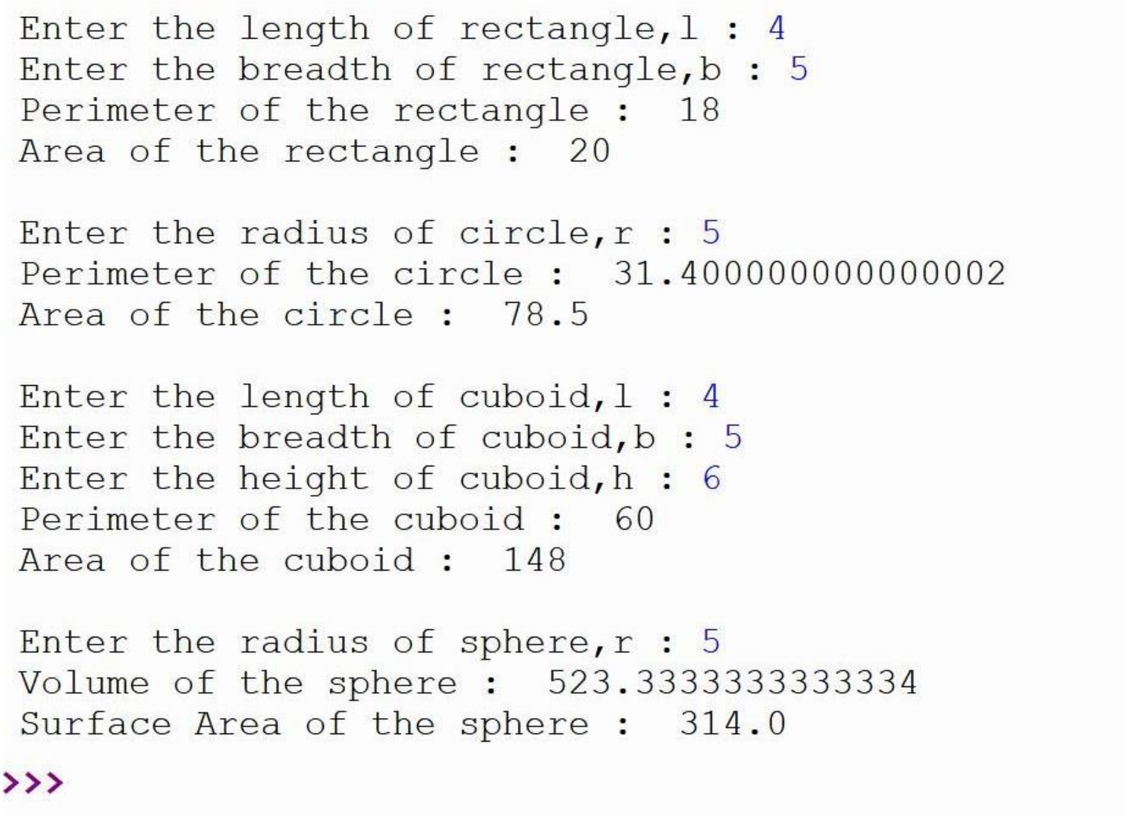
```
def volume(r):
    print ("Volume of the sphere : ",(4/3)*3.14*r*r*r)
def area(r):
    print ("Surface Area of the sphere : ",4*3.14*r*r)
```

graphicsuse.py

```
from graphics import rectangle
from graphics import circle
from graphics.ThreeDgraphics import cuboid
from graphics.ThreeDgraphics import sphere

l=int(input("Enter the length of rectangle,l : "))
b=int(input("Enter the breadth of rectangle,b : "))
rectangle.perimeter(l,b)
rectangle.area(l,b)
print()

r=int(input("Enter the radius of circle,r : "))
circle.perimeter(r)
circle.area(r)
print()
```

```
l=int(input("Enter the length of cuboid,l : "))

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

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

cuboid.perimeter(l,b,h)

cuboid.area(l,b,h)

print()


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

sphere.volume(r)

sphere.area(r)
```

OUTPUT



```
Enter the length of rectangle,l : 4
Enter the breadth of rectangle,b : 5
Perimeter of the rectangle :  18
Area of the rectangle :  20

Enter the radius of circle,r : 5
Perimeter of the circle :  31.400000000000002
Area of the circle :  78.5

Enter the length of cuboid,l : 4
Enter the breadth of cuboid,b : 5
Enter the height of cuboid,h : 6
Perimeter of the cuboid :  60
Area of the cuboid :  148

Enter the radius of sphere,r : 5
Volume of the sphere :  523.3333333333334
Surface Area of the sphere :  314.0
>>>
```

COURSE OUTCOME 4 (CO4)

PROGRAM NO: 1 **DATE : 03/01/2022**

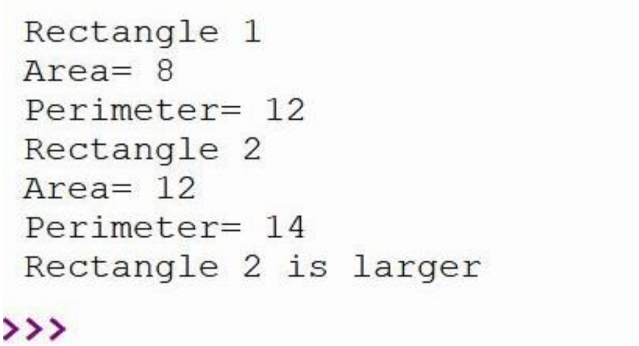
AIM : Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

PROGRAM

```
class Rectangle:

   def_init_(self,length,breadth,ar):

       self.length=length

       self.breadth=breadth

       self.ar=0

   def area(self):

       self.ar=self.length*self.breadth

       #print("area=",self.ar)

       return (self.ar)


   def perimeter(self):

       self.perimeter=2*(self.length+self.breadth)

       #print(perimeter)

       return (self.perimeter)


   def display(self):

       print("Area=",self.ar)

       print("Perimeter=",self.perimeter)



R1=Rectangle(2,4,0)
```

```
R2=Rectangle(3,4,0)


R1.area()

R1.perimeter()


R2.area()

R2.perimeter()

print("Rectangle 1")

R1.display()


print("Rectangle 2")

R2.display()




if (R1.ar>R2.ar):

    print("Rectangle 1 is larger")

else:

    print("Rectangle 2 is larger")
```

OUTPUT



```
Rectangle 1
Area= 8
Perimeter= 12
Rectangle 2
Area= 12
Perimeter= 14
Rectangle 2 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:
    def_init_(self,bal=0):
        self.bal=bal
        name=input("Enter name : ")
        print(".....Account for",name,"is created.... ")

    def deposit(self):
        amount=int(input("Amount to deposit : "))
        self.bal=self.bal+amount
        print("New balance:",self.bal)
    def withdarw(self):
        amount=int(input("Amount to withdraw : "))
        if(self.bal>amount):
            self.bal=self.bal-amount
            print("New balance:",self.bal)
        else:
            print("....Insufficient Balance... ")
            print("Current balance : ",self.bal)

    def display(self):
        print("Current Balance:",self.bal)

print("..........Account......... ")
b1=Bank()
opt='y'
while(opt=='y'):
    choice=int(input("Choices are: \n1. Deposit\n2. Withdarw \n3. Display\n\nEnter your choice: "))
    if(choice == 1):
        b1.deposit
    ()
    elif(choice=
    =2):
        b1.withdar
    w()
    elif(choice==
    3):
        b1.displa
    y() else:
        print("Invalid Choice")

opt=input("Do you want to continue? (Enter 'y'/'n') : ")
```

OUTPUT

```
Enter the Account Number:1233Enter
the name:Fayaz
Enter the account type-(savings/current):savings
Enter the initial balance:2000

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:1

Account Info:
Account Number: 1233
Account Name: Fayaz
```

```
 Account Type: savings
 Account Balance: 2200

 1.Account Info
 2.Deposit
 3.Withdraw 4.Exit
 Select your option:3
 Enter the amount to withdraw:100RS- 100
 Withdrawn successfully

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

 Account Info:
 Account Number: 1233
 Account Name: Fayaz
 Account Type: savings
 Account Balance: 2100

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

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,length,width):
        self.length=length
        self.width=width
    def_lt_(self,a2):
        area1=self.length*self.width
        area2=a2.length*a2.width
        if(area1<area2):
            return(True)
        else:
            return(False)
print("Enter the Details of Rectangle:1")
l1=int(input("Length : "))
w1=int(input("Width : "))
r1=rectangle(l1,w1)
print("Enter the Details of Rectangle:2")
l2=int(input("Length : "))
w2=int(input("Width : "))
r2=rectangle(l2,w2)
if(r1<r2):
    print("Rectangle 2 is larger!!")
else:
print("Rectangle 1 is larger!!")
```

OUTPUT

```
first Rectangle:
Enter the length:3
 Enter the breadth:3
Area= 9

Second Rectangle:
Enter the length:2
Enter the breadth:2
Area= 4

Area of first rectangle 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,hour,minute,second):
        self.hour=hour
         self.minute=minute
         self.second=second
           def_add_(self,a2):
        second=self.second+a2.second
        minute=self.minute+a2.minute
        hour=self.hour+a2.hour
        if(second>60):
            second=second-60
            minute=minute+1
        if(minute>60):
            minute=minute-60
            hour=hour+1
        return hour,minute,second
print("Enter time1 : ")
h1=int(input("hour : "))
m1=int(input("minute : "))
s1=int(input("second : "))

t1=Time(h1,m1,s1)

print("Enter time2 : ")
h2=int(input("hour : "))
m2=int(input("minute : "))
```

```
s2=int(input("second : "))


t2=Time(h2,m2,s2)


hr,min,sec=t1+t2
print(hr,end=":")
print(min,end=":")
print(sec,end=" ")
```

OUTPUT

```
Enter the hour:3
Enter the minutes:44Enter
the second:35

Enter the hour:3
Enter the minutes:50
Enter the second:45

Hour: 6
Minutes: 1       hour      34     minutes
seconds: 1       minutes       20     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,pname):
                   self.pname=pname

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

                 class book(publisher):

               def get(self,title,author):
                   self.title=title
                   self.author=author

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

           class python(book):

             def___init___(self,price,nop,pname):

             super().___init___(pname)
               self.price=price
               self.nop=nop

             def details(self):
               print("Price:",self.price)
```

```
       print("No of pages:",self.nop)

  s1=python(500,35,"K D")
  s1.get("Faya","P D") s1.display()
s1.details()
```

OUTPUT

```
  Title Name:Faya
  Author Name: P D
  Price: 500
No of pages: 35
```

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

```
        f1=open("sample.txt","w")
        f1.write("This is my first line.\n This is my second line \n This is my third line")
        f1=open("sample.txt","r")
ff=f1.readlines()
print(ff)
```

OUTPUT

```
['This is my first line.\n', ' This is my second line \n', ' This is my third line']

This is my first line.

This is my second line

This is my third line
```

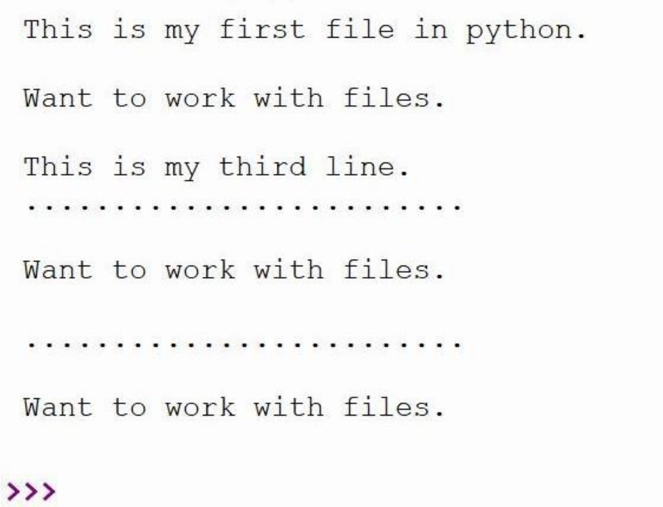
PROGRAM NO: 2 **DATE: 17/01/2022**

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

PROGRAM

```
f1=open("firstfile.txt","r")
for x in f1:
    print(x)

f1.seek(0,0)
print(" ........................\n")
f2=open("odd.txt","w")
ff=f1.readlines()



with open('odd.txt','w') as f2:
    for x in range(0,len(ff)):
        if(x%2!=0):
            print(ff[x])
            f2.write(ff[x])

print(" ........................\n")

f3=open("odd.txt","r")
for x in f3:
    print(x)
```

OUTPUT



```
This is my first file in python.

Want to work with files.

This is my third line.
.........................

Want to work with files.

.........................

Want to work with files.

>>>
```

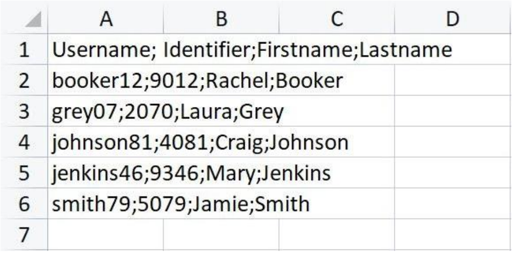
PROGRAM NO: 3 **DATE: 31/01/2022**

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

PROGRAM

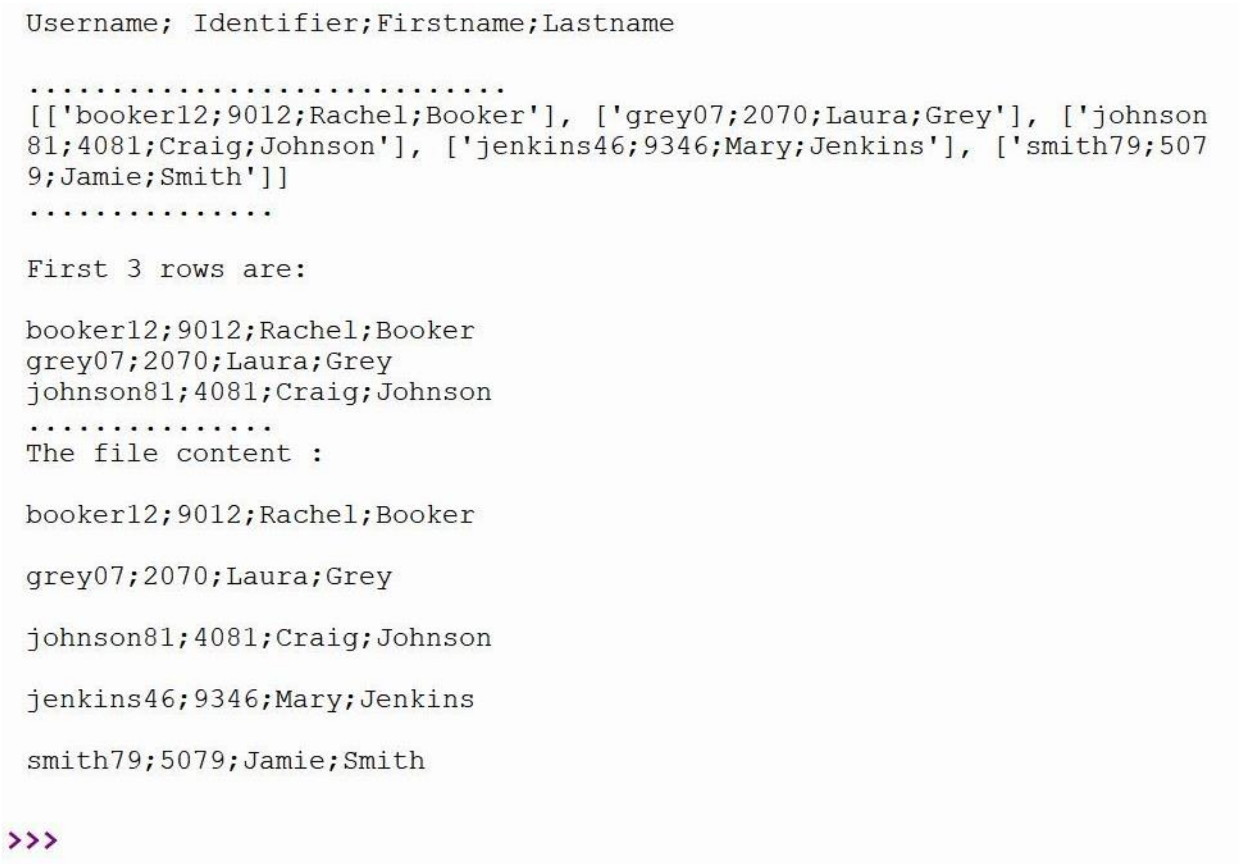
```
import csv

# csv file name
filename = "username.csv"

# initializing the titles and rows list
fields = []
rows = []

# reading csv file
cf=open(filename, 'r')

# creating a csv reader object
csvreader = csv.reader(cf)

# extracting field names through first row
fields = next(cf)
print(fields)
print(" ............................. ")

# extracting each data row one by one
for r in csvreader:
  rows.append(r)

#print the list containing the rows of csv file
print(rows)
print(" ............... ")
print('\nFirst 3 rows are:\n')
for r in rows[:3]:
      print(*r)
print(" .............. ")


print("The file content :\n")
for sl in rows:
  for l in sl:
      print(l)
      print()
cf.close()
```

username.csv



	A	B	C	D
1	Username; Identifier;Firstname;Lastname			
2	booker12;9012;Rachel;Booker			
3	grey07;2070;Laura;Grey			
4	johnson81;4081;Craig;Johnson			
5	jenkins46;9346;Mary;Jenkins			
6	smith79;5079;Jamie;Smith			
7				

OUTPUT



```
Username; Identifier;Firstname;Lastname

..............................
[['booker12;9012;Rachel;Booker'], ['grey07;2070;Laura;Grey'], ['johnson
81;4081;Craig;Johnson'], ['jenkins46;9346;Mary;Jenkins'], ['smith79;507
9;Jamie;Smith']]
...............

First 3 rows are:

booker12;9012;Rachel;Booker
grey07;2070;Laura;Grey
johnson81;4081;Craig;Johnson
...............
The file content :

booker12;9012;Rachel;Booker

grey07;2070;Laura;Grey

johnson81;4081;Craig;Johnson

jenkins46;9346;Mary;Jenkins

smith79;5079;Jamie;Smith


>>>
```

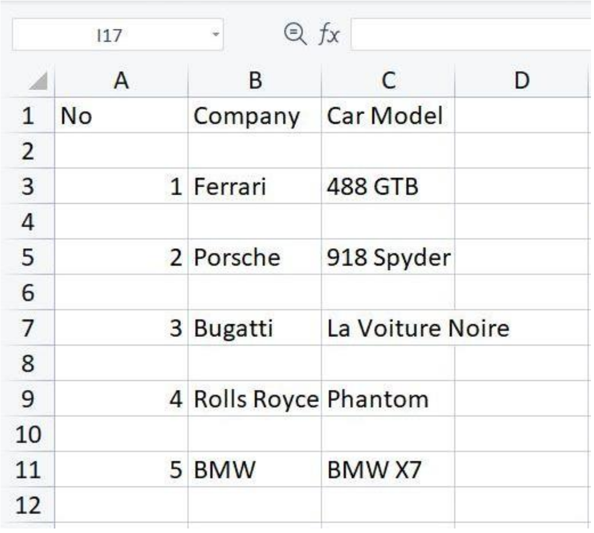
PROGRAM NO: 4 **DATE: 31/01/2022**

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

PROGRAM

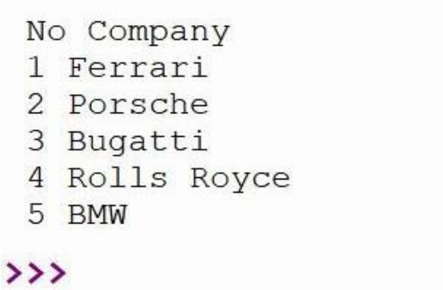
```
import csv
filename = "Names.csv"
cf=open(filename, 'r')
#csvreader = csv.reader(cf)
data = csv.DictReader(cf)
print("No Company")
for r in data:
    print(r['No'], r['Company']))
```

Names.csv



	A	B	C	D
1	No	Company	Car Model	
2				
3	1	Ferrari	488 GTB	
4				
5	2	Porsche	918 Spyder	
6				
7	3	Bugatti	La Voiture Noire	
8				
9	4	Rolls Royce	Phantom	
10				
11	5	BMW	BMW X7	
12				

OUTPUT



```
No Company
1 Ferrari
2 Porsche
3 Bugatti
4 Rolls Royce
5 BMW

>>>
```

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

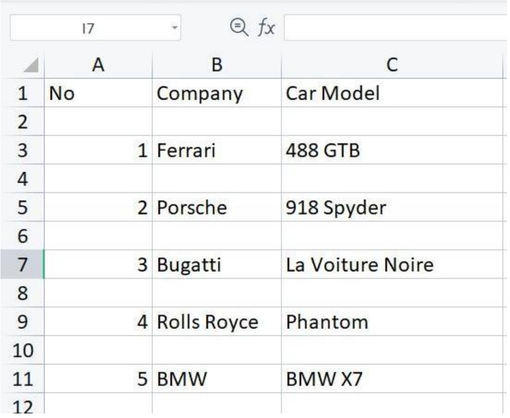
AIM : Write a Python program to write a Python dictionary to a csv file. after writing the CSV file read the CSV file and display the content.

PROGRAM

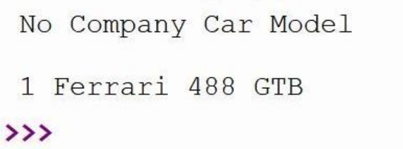
```
import csv
field_names = ['No', 'Company', 'Car Model']
cars = [
{'No': 1, 'Company': 'Ferrari', 'Car Model': '488 GTB'},
{'No': 2, 'Company': 'Porsche', 'Car Model': '918 Spyder'},
{'No': 3, 'Company': 'Bugatti', 'Car Model': 'La Voiture Noire'},
{'No': 4, 'Company': 'Rolls Royce', 'Car Model': '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)
```

Names.csv

OUTPUT



	A	B	C
1	No	Company	Car Model
2			
3	1	Ferrari	488 GTB
4			
5	2	Porsche	918 Spyder
6			
7	3	Bugatti	La Voiture Noire
8			
9	4	Rolls Royce	Phantom
10			
11	5	BMW	BMW X7
12			



```
No Company Car Model

1 Ferrari 488 GTB
>>>
```