

**SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING**

**KADAYIRUPPU, KOLENCHERY 682 311**

**(Affiliated to APJ Abdul Kalam Technological University)**

**ACADEMIC YEAR 2021-23**



**20 MCA 132|PROGRAMMING LABORATORY RECORD**

*Submitted by*

**SANILA LORANCE**

**REG NO: SNG21MCA-2031**

*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)**



**MCA PROGRAMMING LABORATORY RECORD**

*Certified that this is a Bonafide record of practical work done by **sanila loranceto** 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 2021-23.*

Kadayiruppu

Course Instructor

Date: .....

Head of the Department

**Prof.Dr. SANDHYA R**

Submitted for University Practical Examination

**Reg No:SNG21MCA-2031 on -----**

External Examiner

Internal examiner

SL NO .	DATE	NAME OF EXPERIMENT	PAGE NO.	REMARKS
<b>1</b>	<b>CO1</b>			
1	24/11/21	Familiarizing Text Editor, IDE, Code Analysis Tools etc	4	
2	24/11/21	Leap Year	5	
3	24/11/21	List comprehensions	6	
4	24/11/21	occurrences of each word	8	
5	24/11/21	Prompt the user for a list of integers.	9	
6	24/11/21	Store a list of first names.	10	
7	24/11/21	Checking list are of same length, sums to same value, any value occur in both	11	
8	24/11/21	Get a string from an input string and replacing a character	12	
9	24/11/21	Create a string from given string where first and last characters exchanged.	13	
10	24/11/21	Accept the radius from user and find area of circle	14	
11	29/11/21	Find biggest of 3 numbers entered	15	
12	29/11/21	Accept a file name from user and print extension of that	16	
13	29/11/21	Create a list of colors, Display first and last colors.	17	
14	29/11/21	Accept an integer n and compute n+nn+nnn	18	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	19	
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	20	
17	29/11/21	Sort dictionary in ascending and descending order	21	
18	29/11/21	Merge two dictionaries	22	
19	29/11/21	Find gcd of 2 numbers.	23	

20	29/11/21	From a list of integers, create a list removing even numbers.	24	
II	CO2			
1	1/12/21	Program to find the factorial of a number	25	
2	1/12/21	Generate Fibonacci series of N terms	26	
3	1/12/21	Find the sum of all items in a list	27	
4	1/12/21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	28	
5	1/12/21	Display the given pyramid with step number accepted from user	29	
6	1/12/21	Count the number of characters (character frequency) in a string	30	
7	8/12/21	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	31	
8	8/12/21	Accept a list of words and return length of longest word	32	
9	8/12/21	Construct pattern using nested loop	33	
10	8/12/21	Generate all factors of a number. def print_factors(x):	34	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	35	
III	CO3			
1	15/12/21	Work with built-in packages	36	
2	15/12/21	Creation of packages	41	
IV	CO4			
1	9/1/22	Compare two Rectangle objects by their area	43	
2	9/1/22	Create a Bank account with members account number, name, type of account and balance.	45	
3	9/1/22	Overload '<' operator to compare the area of 2 rectangles.	47	
4	9/1/22	Overload '+' operator to find sum of 2 time	48	

5	9/1/22	Use base class constructor invocation and method overriding.	50	
V	CO5			
1	30/1/22	Write a Python program to read a file line by line and store it into a list.	52	
2	30/1/22	Python program to copy odd lines of one file to other	53	
3	30/1/22	Write a Python program to read each row from a given csv file and print a list of strings.	54	
4	30/1/22	Write a Python program to read specific columns of a given CSV file	55	
5	30/1/22	Write a Python program to write a Python dictionary to a csv file.	56	

## **I CO1 PROGRAMS**

**PROGRAM NO: 1**

**DATE:24/11/2021**

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

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,... And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs.

Comparison between IDLE and Thonny

Thonny is built for education and you can download the latest version from the Thonny website. The download options are at the top right. Thonny looks quite different to IDLE - it has different panels for the editor, the shell and the variables watcher plus (show view) lots of other options as well. It has a powerful debugger built in and other tools which let you manage packages and plugins.

The Idle editor comes built-in with Python and is the one that many tutorials use by default. It's a fine, basic, editor that also has a Python shell built in for interactive programming. When you start Idle up, you get the shell window. This allows you to execute python commands and see the results immediately without having to create a program. This can be useful for trying things out.

## PROGRAM NO: 2

DATE:24/11/2021

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

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

### **output**

```
>>> %Run leapyear.py  
  
enter start year:2020  
enter end year:2030  
leap years are: 2020 2024 2028  
  
>>> |
```

## PROGRAM NO: 3

DATE:24/11/2021

### AIM:List comprehensions:

- **Generate positive list of numbers from a given list of integers**

```
list1 =[-10,20,35,-67,70]
re=[num for num in list1 if num>=0]
print(re)
```

### output

```
>>> %Run posetivelist.py
[20, 35, 70]
>>>
```

- **Square of Nnumber**

```
n=int(input("enter the limit"))
l=[i**2 for i in range(0,n+1)]
print("square are ",l)
```

### output

```
>>> %Run squrelist.py
enter the limit5
square are [0, 1, 4, 9, 16, 25]
>>> |
```



- **Form a list of vowels selected from a given word**

```
string=str(input("enter the string"))
print("original string:",string)
print("vowels are:",end="")
for i in string:
    if i in 'aeiouAEIOU':
        print([i],end="")
```

### output

```
>>> %Run stringvowel.py

enter the stringPytho program
original string: Pytho program
vowels are: ['o'] ['o'] ['a']

>>> |
```

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

```
w=input("Enter a word:")
print("Ordinal values corresponding to each element is:")
for i in w:
    print(i,end=":")
    print(ord(i),end="")
```

### output

```
>>> %Run ord.py

Enter a word:PYTHON
Ordinal values corresponding to each element is:
P:80 Y:89 T:84 H:72 O:79 N:78

>>> %Run occurrence.py
```

## PROGRAM NO: 4

DATE:24/11/2021

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

```
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**

```
>>> %Run occurrence.py
```

```
Enter a string : pyhton is a programming language
```

```
count of the occurrence:[('pyhton', 1), ('is', 1), ('a', 1), ('programming', 1), ('language', 1)]
```

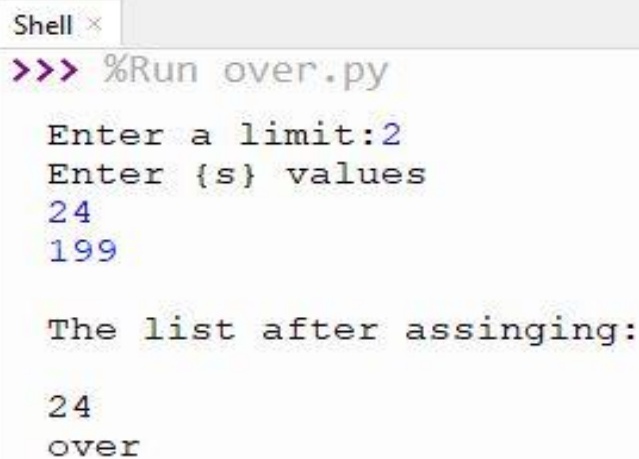
## PROGRAM NO: 5

DATE:24/11/2021

**AIM:**Prompt the user for a list of integers. For all values greater than 100, store 'over' instead

```
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**



```
Shell x
>>> %Run over.py

Enter a limit:2
Enter {s} values
24
199

The list after assinging:

24
over
```

## PROGRAM NO: 6

DATE:24/11/2021

**AIM:** Store a list of first names. Count the occurrences of 'a' within the list

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

### **output**

```
>>> %Run nooccur.py
```

```
count of occurrences of a : 2
```

```
\\
```

**PROGRAM NO: 7**

**DATE:24/11/2021**

**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**

```
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**

```
>>> %Run co17same.py
Lists are of same length
not same sum
Elements that matched are:
[1, 5, 7]
```

## PROGRAM NO: 8

DATE:24/11/2021

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

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

### **output**

```
>>> %Run replacingletter.py  
malayala$
```

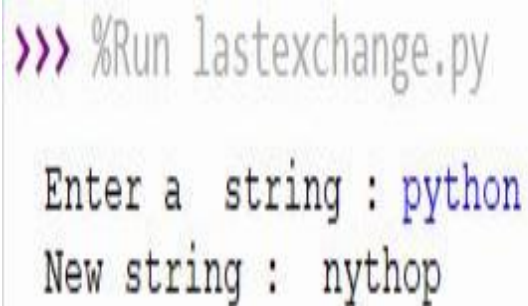
## PROGRAM NO: 9

DATE:24/11/2021

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

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

### **output**



```
>>> %Run lastexchange.py  
  
Enter a string : python  
New string : nythop
```

**PROGRAM NO: 10**

**DATE:24/11/2021**

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

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

**output**



```
>>> %Run areacircle.py
```

```
Enter radius of the circle : 56
```

```
The area of the circle: 9847.04
```



## PROGRAM NO: 11

DATE:29/11/2021

**AIM:Find biggest of 3 numbers entered**

```
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**

```
>>> %Run Largestof3.py
Enter 1st number: 56
Enter 2nd number: 45
Enter 3rd number: 89
The largest number is 89
```

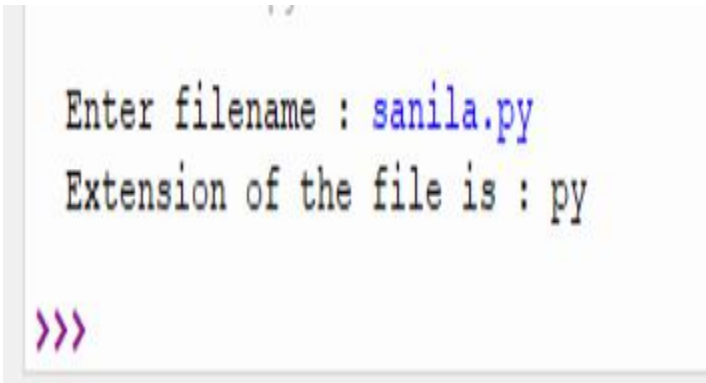
## PROGRAM NO: 12

DATE:29/11/2021

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

```
str= input("Enter filename : ")  
file=str.split(".")  
print("Extension of the file is : " + file[-1])
```

### **output**



```
Enter filename : sanila.py  
Extension of the file is : py
```

```
>>>
```

## 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.

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

### **output**

```
>>> %Run listcolor.py

enter the color:red
enter the color:blue
enter the color:gren
['red', 'blue', 'gren']
red
gren
```

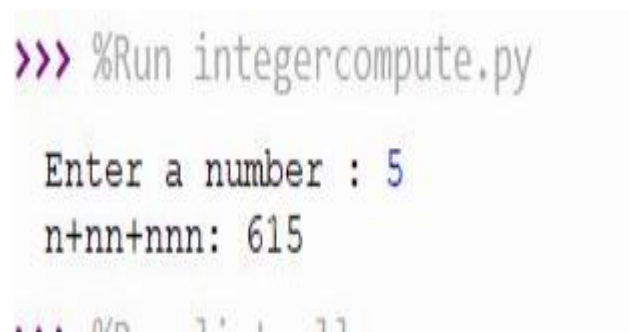
## PROGRAM NO: 14

DATE:29/11/2021

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

```
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**



```
>>> %Run integercompute.py

Enter a number : 5
n+nn+nnn: 615
```

**PROGRAM NO: 15**

**DATE:29/11/2021**

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

```
color_list_1 = set(["White", "pink", "Red","Blue"])  
color_list_2 = set(["Red", "Green","pink"])  
print(color_list_1.difference(color_list_2))
```

**output**



```
>>> %Run listcollurss.py  
  
{'White', 'Blue'}
```

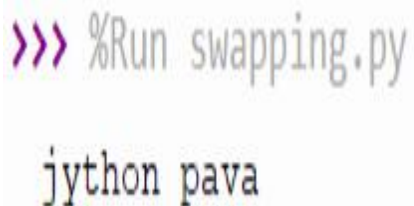
**PROGRAM NO: 16**

**DATE:29/11/2021**

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

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

**output**



```
>>> %Run swapping.py  
jython pava
```

## PROGRAM NO: 17

DATE:29/11/2021

**AIM:Sort dictionary in ascending and descending order.**

```
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**

```
>>> %Run 'dictasc&dec.py'

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:29/11/2021**

### **AIM:Merge two dictionaries**

```
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**

```
>>> %Run mergedict.py

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



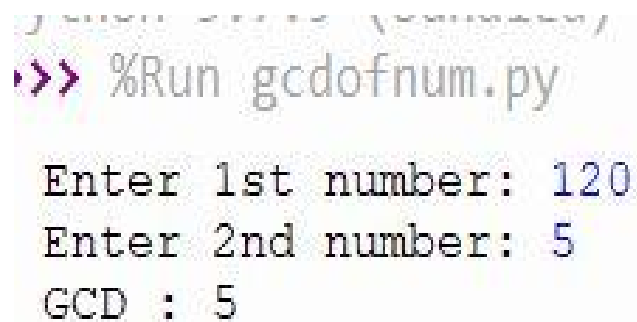
## PROGRAM NO: 19

DATE:29/11/2021

**AIM:**Find gcd of 2 numbers.

```
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**



```
> Run %Run gcdofnum.py
Enter 1st number: 120
Enter 2nd number: 5
GCD : 5
```

**PROGRAM NO: 20**

**DATE:29/11/2021**

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

```
num = [7,8, 120, 25, 44, 20, 27]
print( "Original list:",num)
num = [x for x in num if x%2!=0]
print("list after removing Even numbers:",num)
```

**output**

```
>>> %Run removeeven.py

Original list: [7, 8, 120, 25, 44, 20, 27]
list after removing Even numbers: [7, 25, 27]
```

## **II CO2 PROGRAMS**

**PROGRAM NO: 1**

**DATE:1/12/2021**

**AIM: Program to find the factorial of a number**

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

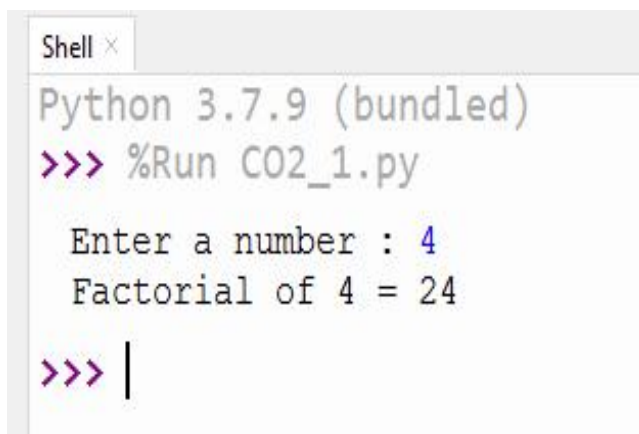
```
f=1
```

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

```
    f=f*i
```

```
print ('Factorial of',n, '=',f)
```

### **output**



```
Shell x
Python 3.7.9 (bundled)
>>> %Run CO2_1.py

Enter a number : 4
Factorial of 4 = 24

>>> |
```

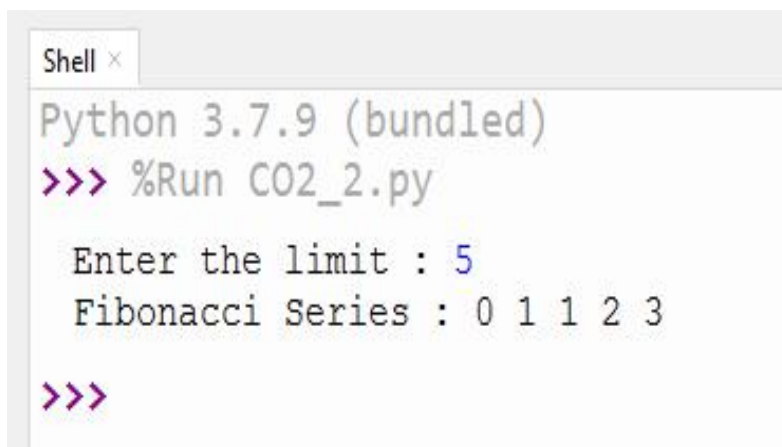
## PROGRAM NO: 2

DATE:1/12/2021

### AIM:Generate Fibonacci series of N terms

```
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



```
Shell x
Python 3.7.9 (bundled)
>>> %Run C02_2.py

Enter the limit : 5
Fibonacci Series : 0 1 1 2 3

>>>
```

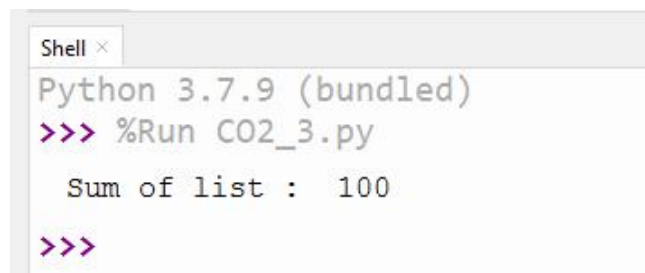
## PROGRAM NO: 3

DATE:1/12/2021

**AIM:**Find the sum of all items in a list

```
list1=[10, 15, 20, 25, 30]
total=sum(list1)
print("Sum of list : ",total)
```

### **output**

A screenshot of a Python shell window titled "Shell x". The window shows the prompt "Python 3.7.9 (bundled)" followed by the command ">>> %Run C02\_3.py". The output of the program is displayed as "Sum of list : 100". The prompt ">>>" is shown again at the bottom of the window.

```
Shell x
Python 3.7.9 (bundled)
>>> %Run C02_3.py
Sum of list : 100
>>>
```

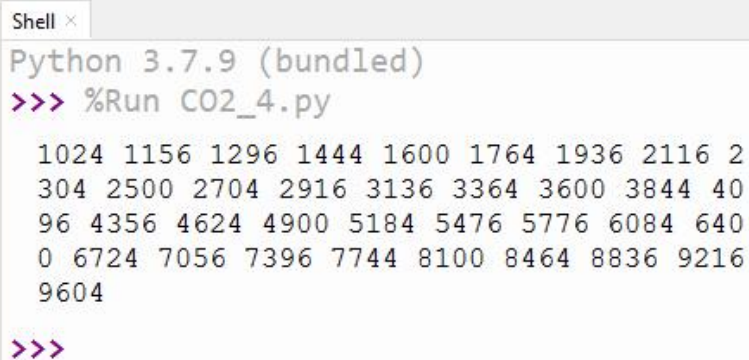
## PROGRAM NO: 4

DATE:1/12/2021

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

```
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**



```
Shell x
Python 3.7.9 (bundled)
>>> %Run CO2_4.py

1024 1156 1296 1444 1600 1764 1936 2116 2
304 2500 2704 2916 3136 3364 3600 3844 40
96 4356 4624 4900 5184 5476 5776 6084 640
0 6724 7056 7396 7744 8100 8464 8836 9216
9604

>>>
```

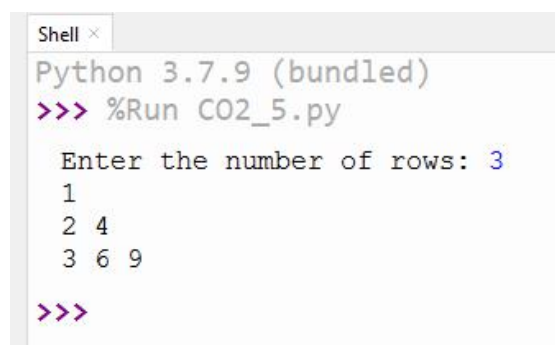
## PROGRAM NO: 5

DATE:1/12/2021

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

```
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**



```
Shell x
Python 3.7.9 (bundled)
>>> %Run C02_5.py
Enter the number of rows: 3
1
2 4
3 6 9
>>>
```

## PROGRAM NO: 6

DATE:1/12/2021

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

```
test_str=str(input("Enter the string : "))  
freq = {}  
for i in test_str:  
    if i in freq:  
        freq[i] += 1  
    else:  
        freq[i] = 1  
print ("Count of all characters : "+ str(freq))
```

### **output**

```
>>> %Run C02_06.py  
Enter the string : python  
Count of all characters : {'p': 1, 'y': 1, 't': 1, 'h': 1, 'o': 1, 'n': 1}
```



## PROGRAM NO:7

DATE:8/12/2021

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

```
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**

```
>>> %Run C02_07.py
enter a string:programming
inputed string is : programming
the formated string is : programingly
```

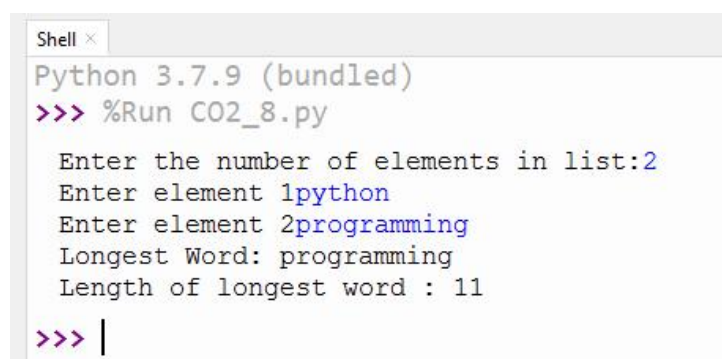
## PROGRAM NO:8

DATE:8/12/2021

**AIM:Accept a list of words and return length of longest word**

```
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**



```
Shell x
Python 3.7.9 (bundled)
>>> %Run CO2_8.py
Enter the number of elements in list:2
Enter element 1python
Enter element 2programming
Longest Word: programming
Length of longest word : 11
>>> |
```

## PROGRAM NO:9

DATE:8/12/2021

AIM: Construct following pattern using nested l

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

```
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

```
Shell x
Python 3.7.9 (bundled)
>>> %Run CO2_9.py
Enter the limit:4

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

>>>
```

## PROGRAM NO:10

DATE:8/12/2021

**AIM:** Generate all factors of a number. def print\_factors(x)

```
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**

```
>>> %Run C02_10.py  
Enter a number :10  
The factors of 10 are:  
1  
2  
5  
10  
>>> |
```

## PROGRAM NO:11

DATE:8/12/2021

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

```
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(10,20))
print("Area of Rectangle:", r_area(30,20))
print("Area of Square :", s_area(15))
```

### **output**

```
>>> %Run C02_11.py
Area of Triangle : 100.0
Area of Rectangle: 600
Area of Square : 225
>>>
```

### **III CO3 PROGRAMS**

**PROGRAM NO: 1**

**DATE:15/12/2021**

**AIM:Work with built-in packages**

**(a) Module math**

```
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**

```
>>> %Run 'module math.py'
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**

```
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**

```
>>> %Run 'module time.py'

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

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

### output

```
>>> %Run 'module calendar.py'
```

```
Enter month: 12
Enter year :2021
December 2021
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
```

2015

January

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	

February

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	

March

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					

April

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			

May

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

June

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					

July

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		

August

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						

September

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				

October

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	

November

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						

December

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			

```
>>>
```



#### (d) Module datetime

```
import datetime
t=datetime.time(22,56,44) #time class
print(t)
print("Hour :",t.hour)
print("Minute :",t.minute)
print("Second :",t.second)
print("Microsecond :",t.microsecond)
print(".....\n")
d=datetime.date.today()
print(d)
print("Year",d.year)
print("Month",d.month)
print("Day",d.day)
print(".....\n")
d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2)
print(td)
d2=d1+td
print(d2)
print(".....\n")
dt=datetime.datetime.combine(d,t)
print(dt)
```

#### output

```
>>> %Run 'module datetime.py'
22:56:44
Hour : 22
Minute : 56
Second : 44
Microsecond : 0
.....
2021-12-20
Year 2021
Month 12
Day 20
.....
2021-12-20
2 days, 0:00:00
2021-12-22
.....
2021-12-20 22:56:44
>>>
```

### (e) Module random

```
import random
mylist = ["apple", "banana", "cherry"]

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

print(random.choices(mylist, k=2))

print(random.sample(mylist, k=2))     #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 'module random.py'
apple
['apple', 'cherry']
['apple', 'banana']
['cherry', 'banana', 'apple']
4
>>>
```

### (f) Module statistics

```
import statistics
print(statistics.mean([10,20,30,40,50,60])) #To calculate the mean of given numbers
print(statistics.median([10,20,30])) #To find the median of given numbers
print(statistics.harmonic_mean([10,20,30,40,50,60])) #To calculate the harmonic mean of given
numbers
```

#### output

```
>>> %Run 'module statistics (1).py'
35
20
24.489795918367346
>>>
```

## 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)

### **Package graphics**

(1) **\_\_init\_\_.py**

(2) **circle.py**

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

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

(3) **rectangle.py**

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

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

### **Subpackage ThreeDgraphics**

(1) **\_\_init\_\_.py**

(2) **cuboid.py**

```
def perimeter(l,b,h):  
    print ("Perimeter : ",4*(l+b+h))
```

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

(3) **sphere.py**

```
def volume(r):  
    print ("Volume : ",(4/3)*3.14*r*r*r)
```

```
def area(r):  
    print ("Surface Area : ",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,l : "))  
b=int(input("Enter the breadth,b : "))  
rectangle.perimeter(l,b)  
rectangle.area(l,b)
```

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

```
l=int(input("Enter the length,l : "))  
b=int(input("Enter the breadth,b : "))  
h=int(input("Enter the height,h : "))  
cuboid.perimeter(l,b,h)  
cuboid.area(l,b,h)
```

```
r=int(input("Enter the radius,r : "))  
sphere.volume(r)  
sphere.area(r)
```

### output

```
>>> %Run graphicsuse.py  
Enter the length,l : 4  
Enter the breadth,b : 5  
Perimeter : 18  
Area : 20  
Enter the radius,r : 5  
Perimeter : 31.400000000000002  
Area : 78.5  
Enter the length,l : 4  
Enter the breadth,b : 5  
Enter the height,h : 6  
Perimeter : 60  
Area : 148  
Enter the radius,r : 5  
Volume : 523.3333333333334  
Surface Area : 314.0  
>>> %Run graphicsuse.py
```

## **IV CO4 PROGRAMS**

**PROGRAM NO: 1**

**DATE:9/1/2022**

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

```
class Rectangle:
    def __init__(self,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("Area of Rectangle1")
R1.display()

print("Area of Rectangle2")
R2.display()
```

```
if (R1.ar>R2.ar):  
    print(R1.ar,"is graeter")  
else:  
    print(R2.ar,"is greater")
```

### **output**

```
>>> %Run co4_1.py  
Area of Rectangle1  
area= 8  
perimeter= 12  
Area of Rectangle2  
area= 12  
perimeter= 14  
12 is greater  
>>>
```

## PROGRAM NO: 2

DATE:9/1/2022

**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.**

```
class Bank:
```

```
    def __init__(self,bal=0):
        #self.accno=accno
        #self.name=name
        #self.acctype=acctype
        self.bal=bal
        name=input("Enter name : ")
```

```
    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'):
```

```
    #print("your choice: 1. deposit \n 2. withdarw \n 3. display\n")
```

```
    choice=int(input("Choices are: \n1. Deposit\n2. Withdarw \n3. Display\n\nEnter your choice: "))
```

```
    if(choice == 1):
```

```
        b1.deposit()
```

```
elif(choice==2):
    b1.withdarw()
elif(choice==3):
    b1.display()
else:
    print("Invalid Choice")
```

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

### **output**

```
Account Number: 7856543453
Account Name: anu
Account Type: savings
Account Balance: 500 .00
-----

--WELCOME TO PYTHON BANK--

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

Select your option:2

Enter the Amount to Deposit: 2000
Rs. 2000 Deposited Successfully...
-----

--WELCOME TO PYTHON BANK--

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

Select your option:3

Enter the Amount to Withdraw: 500
Rs. 500 Withdrawn Successfully...
-----

--WELCOME TO PYTHON BANK--

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

Select your option:|
```



## PROGRAM NO: 3

DATE:9/1/2022

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

```
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

```
>>> %Run co4_3.py
Enter the Details of Rectangle:1
Length : 50
Width : 60
Enter the Details of Rectangle:2
Length : 5
Width : 6
Rectangle 1 is larger!!
```

## PROGRAM NO: 4

**DATE:9/1/2022**

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

```
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**

```
>>> %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:**9/1/2022

**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.

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

    def publisherdisplay(self):
        print(self.publishername)

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

    def bookdisplay(self):
        print(self.title)
        print(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("Publisher Name: ",self.publishername)
        print("Title: ",self.title)
        print("Author: ",self.author)
        print("Price: ",self.price)
        print("No. of Pages: ",self.page)

obj=python("Akshaya publishers","Python","Guido van Rossum",236,215);
obj.pythondisplay();
```

## **output**

```
>>> %Run C04_5.py
Publisher Name: Akshaya publishers
Title: Python
Author: Guido van Rossum
Price: 236
No. of Pages: 215
>>>
```

## **V CO5 PROGRAMS**

### **PROGRAM NO: 1**

**DATE:**30/1/2022

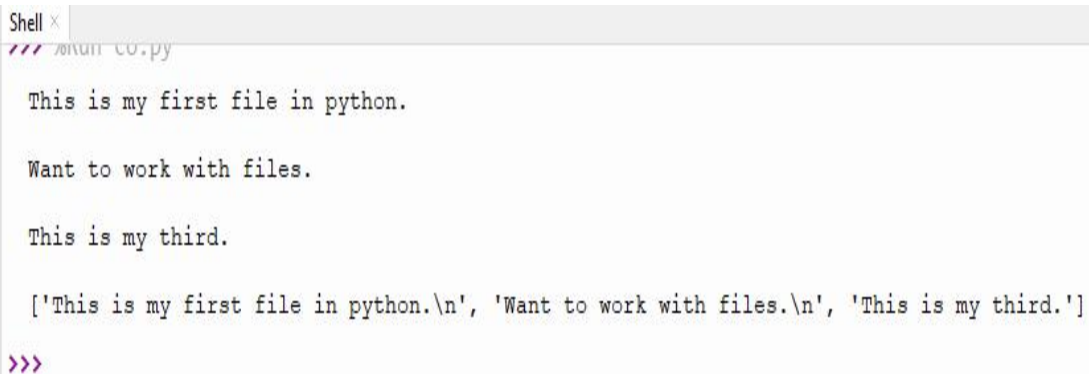
**AIM:**write a program to read a file line by line and store it into a list.

```
f1=open("firstfile.txt","w")
f1.write("This is my first file in python.\nWant to work with files.\nThis is my third.")
f1.close()
```

```
f1=open("firstfile.txt","r")
f1.seek(0,0)
ff=f1.readlines()
for x in range(0,len(ff)):
    print(ff[x])
```

```
print()
print(ff)
f1.close()
```

### **output**



```
Shell x
Python 3.9.0
>>>
This is my first file in python.

Want to work with files.

This is my third.

['This is my first file in python.\n', 'Want to work with files.\n', 'This is my third.']
>>>
```

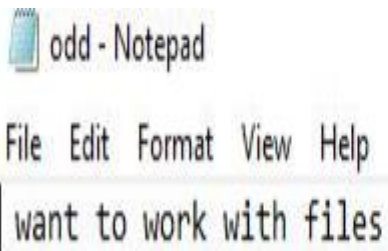
## PROGRAM NO: 2

DATE:30/1/2022

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

```
f1=open("secfile.txt","r")  
  
ff=f1.readlines()  
with open("odd.txt","w") as f2:  
  
for x in range(0,len(ff)):  
  
    if(x%2!=0):  
  
        f2.write(ff[x])
```

### **output**



odd - Notepad

File Edit Format View Help

want to work with files

## PROGRAM NO: 3

DATE:30/1/2022

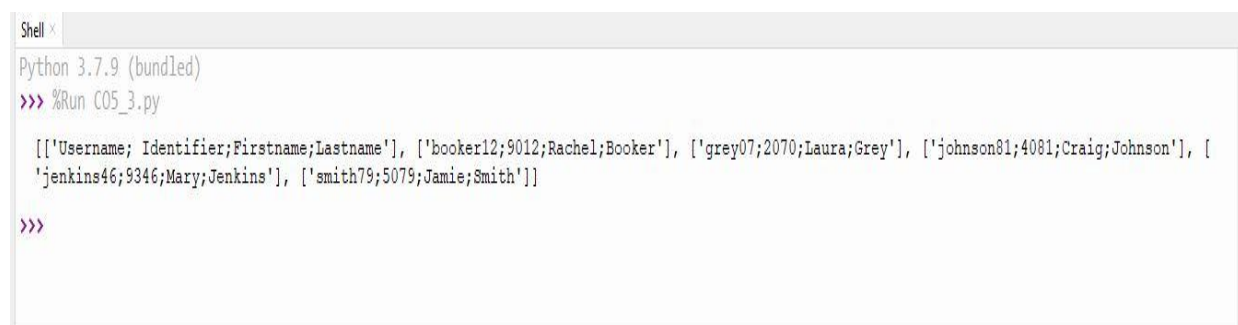
**AIM:**Write a Python program to read each row from a given csv file and print a list of strings..

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

### **username.csvs**

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

### **output**



```
Shell x
Python 3.7.9 (bundled)
>>> %Run C05_3.py

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

>>>
```



## PROGRAM NO: 4

DATE:30/1/2022

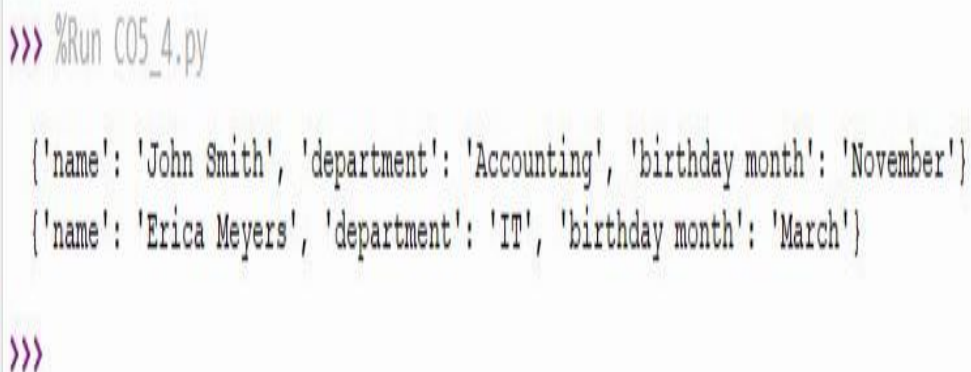
**AIM:** Write a Python program to read specific columns of a given CSV file and print the content of the columns.

```
import csv
filename = "emp.txt"
fields = []
rows = []
cf=open(filename, 'r')
csvreader = csv.DictReader(cf)
for r in csvreader:
    print(dict(r))
```

### emp.txt

```
name,department,birthday month
John Smith,Accounting,November
Erica Meyers,IT,March
```

### output



```
>>> %Run C05_4.py

{'name': 'John Smith', 'department': 'Accounting', 'birthday month': 'November'}
{'name': 'Erica Meyers', 'department': 'IT', 'birthday month': 'March'}

>>>
```

## PROGRAM NO: 5

DATE:30/1/2022

**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.

```
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)
```

## output

```
>>> %Run C05_5.py  
No Company Car Model  
  
1 Ferrari 488 GTB  
  
2 Porsche 918 Spyder  
  
3 Bugatti La Voiture Noire  
  
4 Rolls Royce Phantom  
  
5 BMW BMW X7  
  
>>> |
```







