

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577



FOCUS ON EXCELLENCE

20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: EMELSHA P E

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1

Batch: A

Roll No: 55

MARCH 2022

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577



FOCUS ON EXCELLENCE

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done by EMELSHA P E (FIT21MCA-2055) in the 20MCA131 PROGRAMMING LAB Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.

Signature of Staff in Charge

Name:

Signature of H O D

Name:

Date of University practical examination

Signature of
Internal Examiner

Signature of
External Examiner

CONTENT

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
COURSE OUTCOME-1				
1	28/10/21	Display future leap years from current year to a final year entered by user.	1	
2	28/10/21	List comprehensions: (a) Generate positive list of numbers from a given list of integers (b) Square of N numbers (c) Form a list of vowels selected from a given word (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)	2	
3	28/10/21	Count the occurrences of each word in a line of text.	4	
4	28/10/21	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.	5	
5	10/11/21	Store a list of first names. Count the occurrences of 'a' within the list	6	
6	10/11/21	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	7	
7	10/11/21	Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion ->oni\$n]	10	
8	10/11/21	Create a string from given string where first and last characters exchanged. [eg: python ->nythop]	11	
9	10/11/21	Accept the radius from user and find area of circle.	12	
10	11/11/21	Find biggest of 3 numbers entered.	12	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
11	11/11/21	Accept a file name from user and print extension of that.	13	
12	11/11/21	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	14	
13	11/11/21	Accept an integer n and compute $n+nn+nnn$.	14	
14	11/11/21	Print out all colors from color-list1 not contained in color-list2.	15	
15	17/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	16	
16	17/11/21	Sort dictionary in ascending and descending order.	17	
17	17/11/21	Merge two dictionaries.	17	
18	17/11/21	Find gcd of 2 numbers.	18	
19	17/11/21	From a list of integers, create a list removing even numbers.	19	
COURSE OUTCOME-2				
1	25/11/21	Program to find the factorial of a number.	19	
2	25/11/21	Generate Fibonacci series of N terms.	20	
3	25/11/21	Find the sum of all items in a list	20	
4	25/11/21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	21	
5	02/12/21	Display the given pyramid with step number accepted from user.Eg: N=4 1 2 4 3 6 9 8 12 16	22	
6	02/12/21	Count the number of characters (character frequency) in a string.	22	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
7	02/12/21	Add ing at the end of given string. If it already ends with ing replace it with ly	23	
8	09/12/21	Accept a list of words and return length of longest word.	23	
9	09/12/21	Construct following pattern using nested loop *	24	
10	09/12/21	Generate all factors of a number.	25	
COURSE OUTCOME-3				
1	29/01/22	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)	25	
COURSE OUTCOME-4				
1	13/01/22	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	29	
2	13/01/22	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.	30	

Sl No	Date of Experiment	Title of the Experiment	Page No:	Signature of Staff –In – Charge
3	13/01/22	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	31	
4	20/01/22	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	32	
5	20/01/22	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.	32	
COURSE OUTCOME-5				
1	03/02/22	Write a Python program to read a file line by line and store it into a list.	34	
2	03/02/22	Write a Python program to read each row from a given csv file and print a list of string.	35	

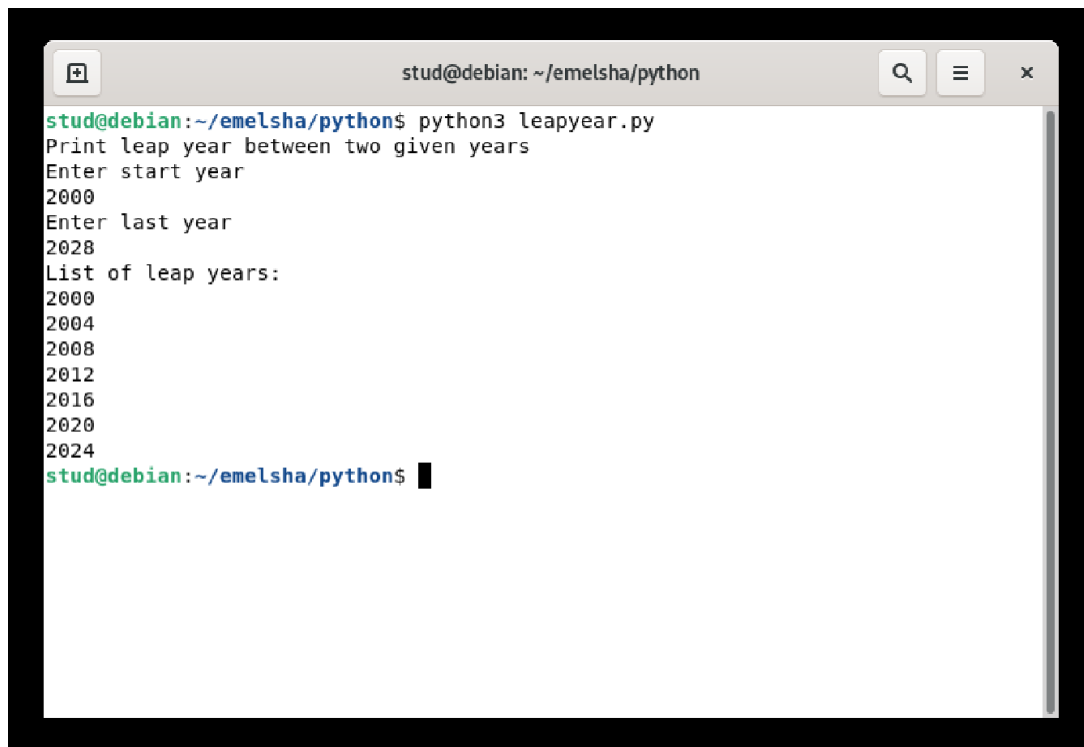
COURSE OUTCOME 1**PROGRAM-1**

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

Program Code:

```
print("Print leap year between two given years")
print("Enter start year")
start Year=int(input())
print("Enter last year")
end Year=int(input())
print("List of leap years:")
for year in range(start Year, end Year):
    if(0==year%4)and(0!=year%100)or(0==year%400):
        print(year)
```

Output:



```
stud@debian: ~/emelsha/python
Print leap year between two given years
Enter start year
2000
Enter last year
2028
List of leap years:
2000
2004
2008
2012
2016
2020
2024
stud@debian:~/emelsha/python$
```

PROGRAM-2

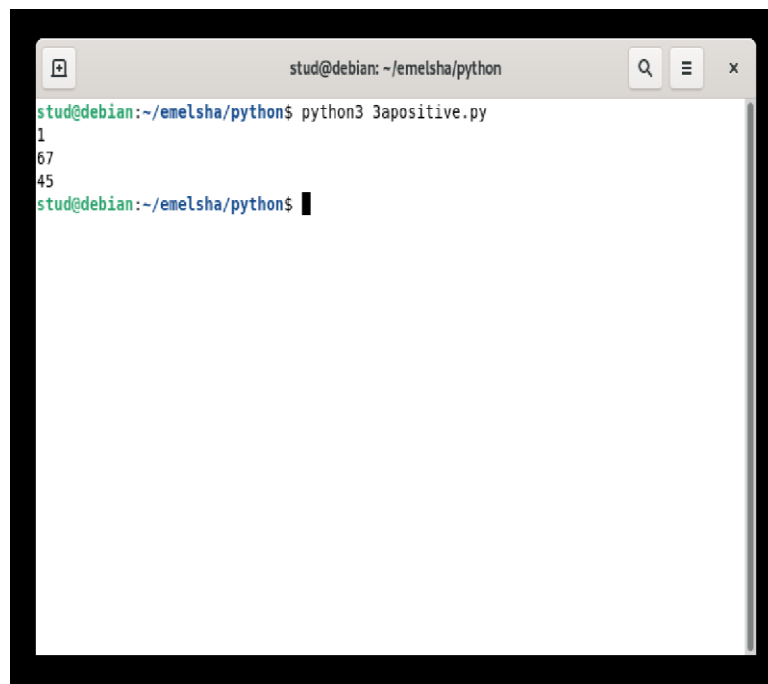
List Comprehensions:

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

Program Code:

```
list=[1,-2,67,45,-5]
for num in list:
    if num>0:
        print(num)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The terminal shows the command 'python3 3apositive.py' being executed. The output of the script is displayed as three lines: '1', '67', and '45'. The prompt 'stud@debian:~/emelsha/python\$' is visible at the bottom of the terminal window.

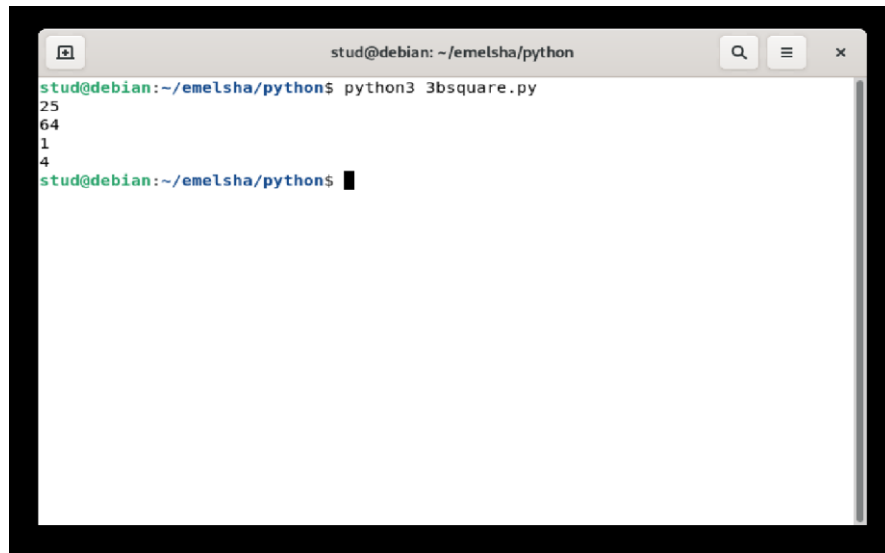
```
stud@debian:~/emelsha/python$ python3 3apositive.py
1
67
45
stud@debian:~/emelsha/python$
```

b)Square of Nnumbers.

Program code:

```
list=[5,8,-1,-2]
for num in list:
    print(num * num)
```


Output:



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 3bsquare.py
25
64
1
4
stud@debian:~/emelsha/python$
```

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

Program Code:

```
stringA="EmElSha"
print("Given String:\n",stringA)
vowels="AaEeIiOoUu"
li=[]
for r in stringA:
    if r in vowels:
        li.append(r)
        print(li)
```

Output:



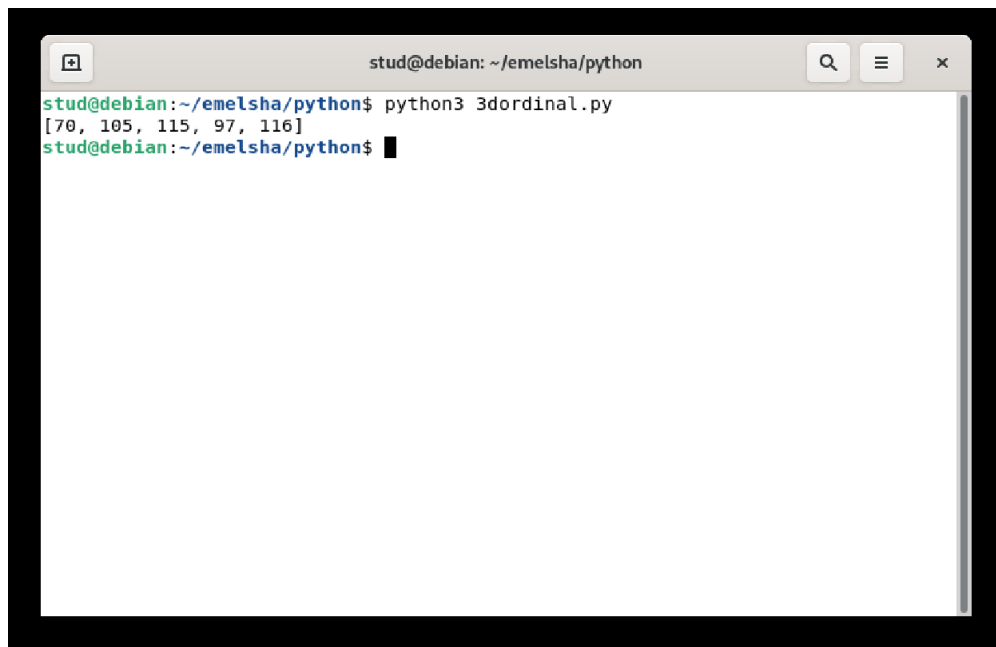
```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 3cvowels.py
Given String:
EmElSha
['E', 'E', 'a']
stud@debian:~/emelsha/python$
```

d) List ordinal value of each element of a word.

Program code:

```
stringp="Fisat"
"""for c in stringp:
print(ord(c))"""
s=[ord(p)for p in stringp]
print(s)
```

Output:



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 3dordinal.py
[70, 105, 115, 97, 116]
stud@debian:~/emelsha/python$
```

PROGRAM-3

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

Program code:

```
s="HI hello, i am mia and i am dia"
l=s.split()
d={x:l.count(x) for x in l}
print(d)
```

Output:

```
{'HI': 1, 'hello,': 1, 'i': 2, 'am': 2, 'mia': 1, 'and': 1, 'dia': 1}
```

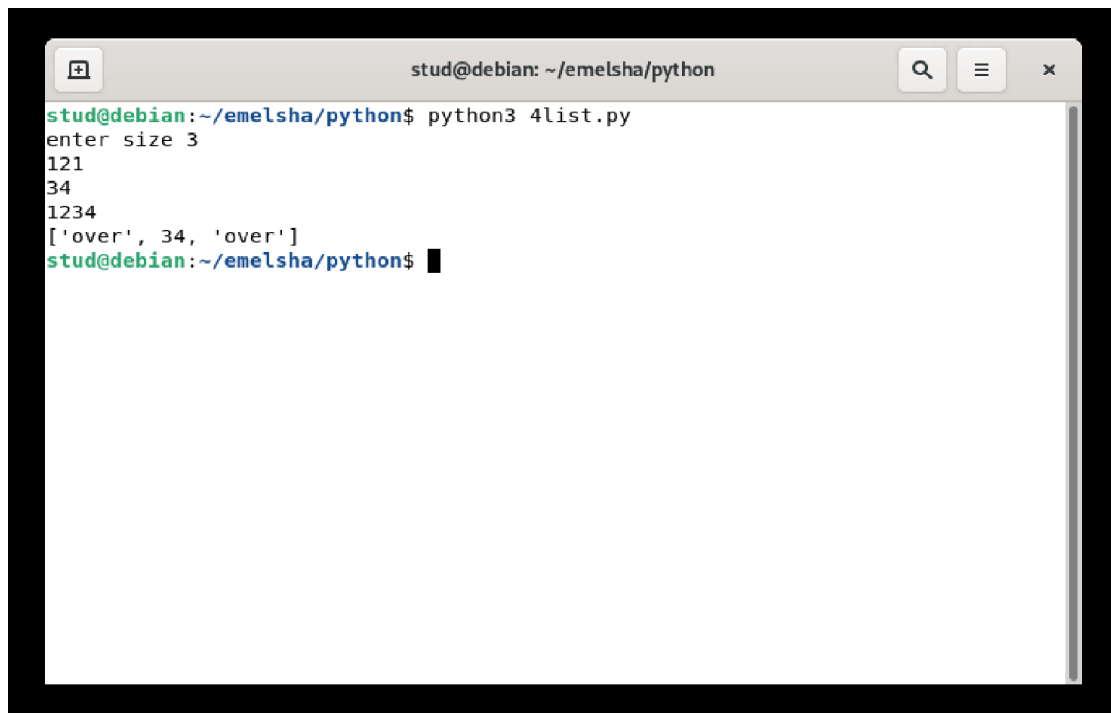
PROGRAM-4

Prompt the user for a list of integers. For all values greater than 100 store “over” instead.

Program code:

```
x=int(input("enter size"))
li=[]
for i in range(0,x):
    el=int(input())
    if(el>=100):
        li.append("over")
    else:
        li.append(el)
print(li)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The terminal shows the execution of a Python script named '4list.py'. The user enters '3' for the size. Then, they enter three integers: '121', '34', and '1234'. The program outputs the list ['over', 34, 'over'].

```
stud@debian:~/emelsha/python$ python3 4list.py
enter size 3
121
34
1234
['over', 34, 'over']
stud@debian:~/emelsha/python$
```

PROGRAM-5

Store the list of first names. Count the occurrence of "a" within the list

Program code:

```
list=["anu","ann","hima"]
count=0
print(list)
foriinlist:
    forkini:
        if(k=='a'):
            count=count+1
            print(count)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The window shows the execution of a Python script named '6name.py'. The output of the script is displayed as follows:

```
stud@debian:~/emelsha/python$ python3 6name.py
['anu', 'ann', 'hima']
3
stud@debian:~/emelsha/python$
```

The terminal window has a standard Linux desktop environment interface with a search icon, a menu icon, and a close button in the title bar.

PROGRAM-6

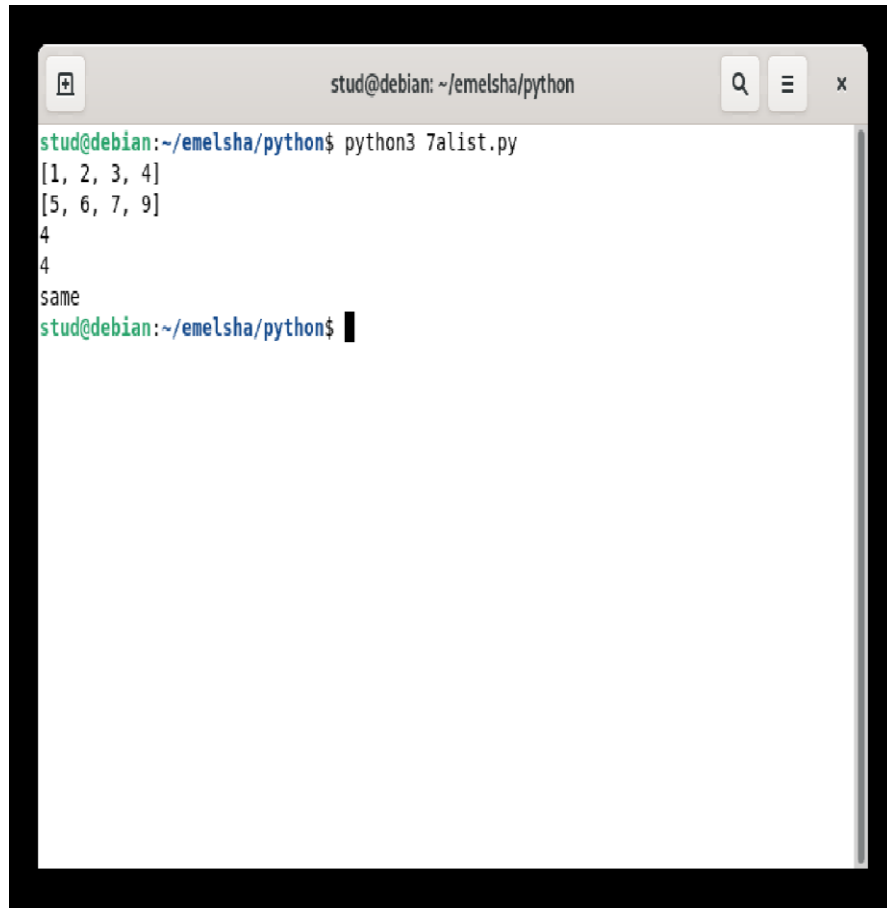
Enter 2 lists of integers

a)check whether list are of same length

Program code:

```
list1=[1,2,3,4]
print(list1)
list2=[5,6,7,9]
print(list2)
p=len(list1)
q=len(list2)
print(p)
print(q)
if(p==q):
    print("same")
else:
    print("not same")
```

Output:



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 7alist.py
[1, 2, 3, 4]
[5, 6, 7, 9]
4
4
same
stud@debian:~/emelsha/python$
```

b)whether list sums to same value

```
l1=[1,2,3]
print(l1)
l2=[5,6,7,2]
print(l2)
s=0
for i in l1:
    s=s+i
print("sum of l1=",s)
p=0
for j in l2:
    p=p+j
print("sum of l2=",p)
if(s==p):
    print("same")
else:
    print("not same")
```

Output:

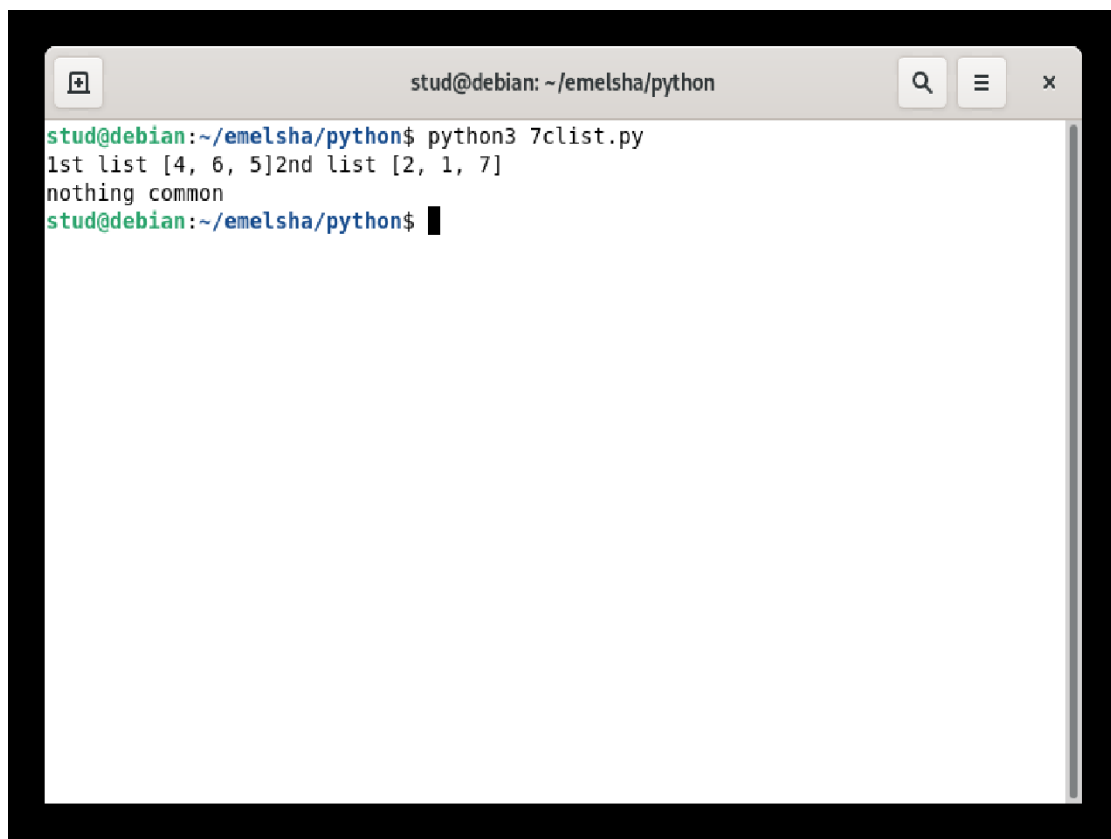
A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The terminal shows the execution of a Python script named '7blist.py'. The output of the script is displayed as follows:

```
stud@debian:~/emelsha/python$ python3 7blist.py
[1, 2, 3]
[5, 6, 7, 2]
sum of l1= 6
sum of l2= 20
not same
stud@debian:~/emelsha/python$
```

c)whether any value occur in both

```
l1=[4,6,5]
l2=[2,1,7]
count=0
print("1st list",str(l1)+"2nd list",str(l2))
forxinl1:
    ifxinl2:
        print("yes there is",x)
        count=count+1
    if(count==0):
        print("nothing common")
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The terminal shows the execution of a Python script named '7clist.py'. The output of the script is: '1st list [4, 6, 5]2nd list [2, 1, 7]' followed by 'nothing common'. The prompt 'stud@debian:~/emelsha/python\$' is visible at the bottom of the terminal window.

```
stud@debian:~/emelsha/python$ python3 7clist.py
1st list [4, 6, 5]2nd list [2, 1, 7]
nothing common
stud@debian:~/emelsha/python$
```

PROGRAM-7

Get a string from an input string where all occurrences of first character replaced with '\$', except first character.

[eg: onion->oni\$n]

Program code:

```
s=input("enter a string\n")
print("entered string is:",s)
a=s[0]
str=s.replace(a,"$")
str1=a+str[1:]
print(str1)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The window shows the execution of a Python script named 'program1.py'. The user enters 'onion' in response to the prompt 'enter a string'. The program outputs 'entered string is: onion' and then 'oni\$n' after replacing the first character 'o' with '\$'. The terminal prompt returns to 'stud@debian: ~/emelsha/python\$' with a cursor.

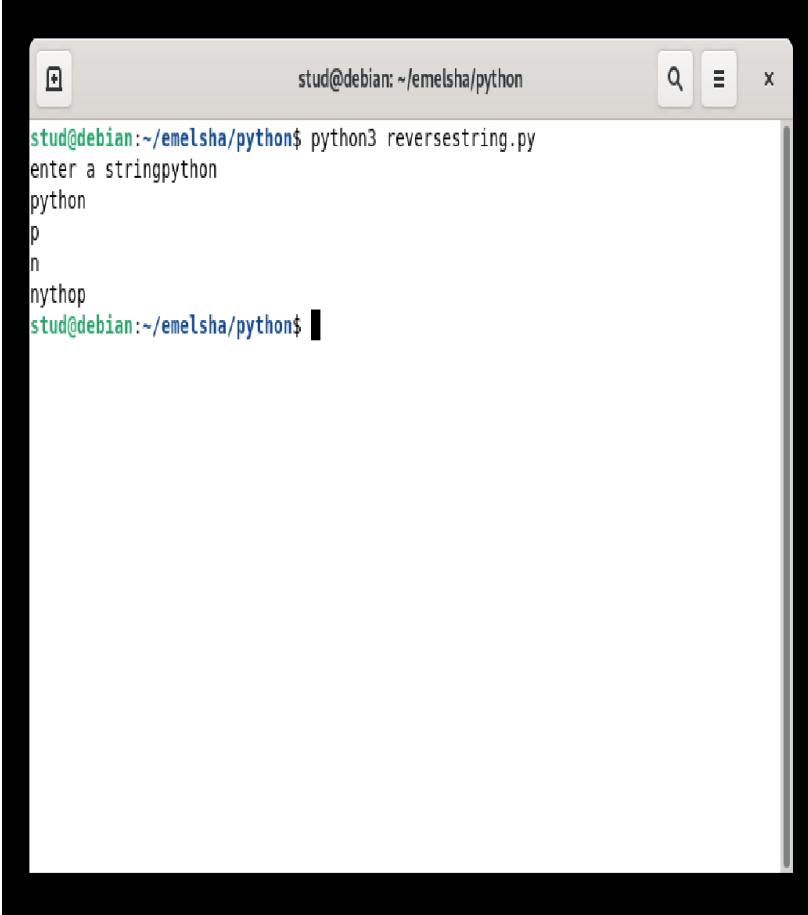
```
stud@debian: ~/emelsha/python$ python3 program1.py
enter a string
onion
entered string is: onion
oni$
stud@debian: ~/emelsha/python$
```


PROGRAM-8

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

```
a=input("enter a string")
print(a)
a1=a[0]
a2=a[-1]
print(a1)
print(a2)
rev=(a2+a[1:len(a)-1]+a1)
print(rev)
```

Output:



The screenshot shows a terminal window titled 'stud@debian: ~/emelsha/python'. The user runs the command 'python3 reversestring.py'. The program prompts 'enter a string' and the user enters 'python'. The program then prints the first character 'p' and the last character 'n' on separate lines. Finally, it prints the reversed string 'nythop'. The prompt 'stud@debian: ~/emelsha/python\$' is visible at the bottom.

```
stud@debian: ~/emelsha/python$ python3 reversestring.py
enter a stringpython
python
p
n
nythop
stud@debian: ~/emelsha/python$
```

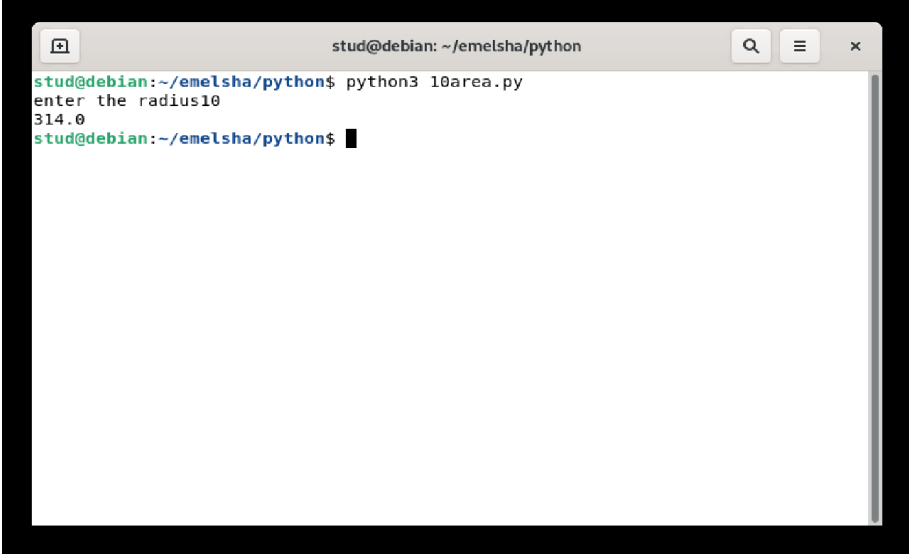
PROGRAM-9

Accept the radius from user and find area of circle.

Program code:

```
p=int(input("enter the radius"))
a=3.14*p*p
print(a)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The prompt is 'stud@debian:~/emelsha/python\$'. The user has entered 'python3 10area.py'. The program prompts 'enter the radius' and the user has entered '10'. The program outputs '314.0'. The prompt is now 'stud@debian:~/emelsha/python\$' with a cursor.

```
stud@debian:~/emelsha/python$ python3 10area.py
enter the radius10
314.0
stud@debian:~/emelsha/python$
```

PROGRAM-10

Find biggest of 3 numbers entered.

Program code:

```
a=int(input("enter 1st number"))
b=int(input("enter 2nd number"))
c=int(input("enter 3rd number"))
ifa>b and a>c:
print(a)
ifc>b and c>a:
print(c)
else:
print(b)
```

Output:



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 11biggest.py
enter 1st number4
enter 2nd number3
enter 3rd number6
6
stud@debian:~/emelsha/python$
```

PROGRAM-11

Accept a filename from user and print extension of that.
Program code:

```
Import os
a=input("Enter the filename\n")
print(os.path.splitext(a))
```

Output:



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 12FILE.PY
Enter the file name
FILE.PY
('FILE', '.PY')
stud@debian:~/emelsha/python$
```

PROGRAM-12

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

Program code:

```
list1=[]  
string=input("Enter colors separated by comma:\n")  
for i in string.split(","):  
    list1.append(i)  
print("First and last colors in the list are",list1[0],"and",list1[-1])
```

Output:



```
stud@debian: ~/emelsha/python  
stud@debian:~/emelsha/python$ python3 13color.py  
Enter colors separated by comma:  
red,yellow,blue  
First and last colors in the list are red and blue  
stud@debian:~/emelsha/python$
```

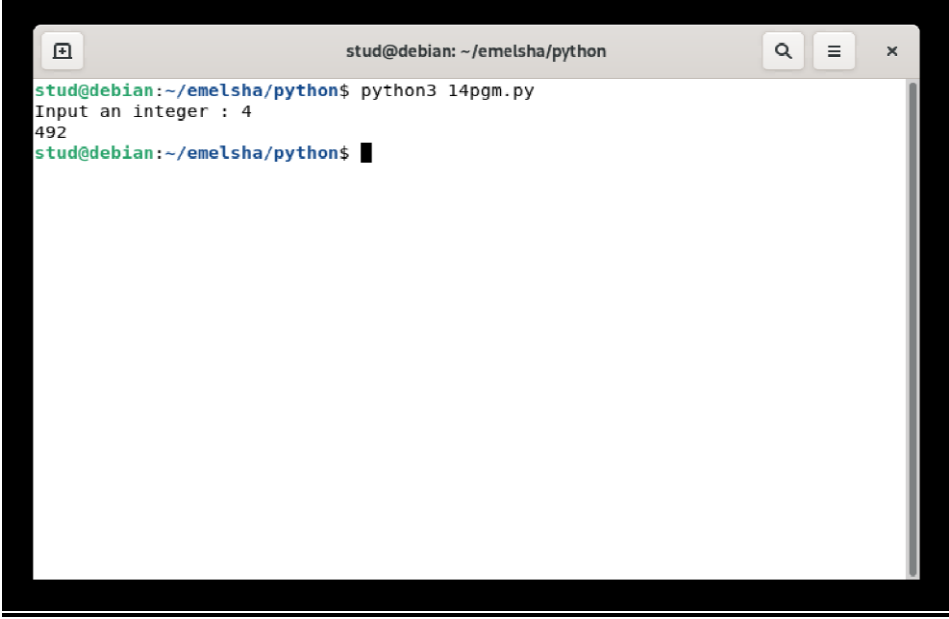
PROGRAM-13

Accept an integer n and compute $n+nn+nnn$.

Program code:

```
a=int(input("Input an integer:"))  
n1=(a*1)  
n2=(a*11)  
n3=(a*111)  
print(n1+n2+n3)
```

Output:

A screenshot of a terminal window titled 'stud@debian: ~/emelsha/python'. The terminal shows the command 'python3 14pgm.py' being executed. The program prompts 'Input an integer : 4' and outputs '492'. The prompt returns to 'stud@debian: ~/emelsha/python\$'.

```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 14pgm.py
Input an integer : 4
492
stud@debian:~/emelsha/python$
```

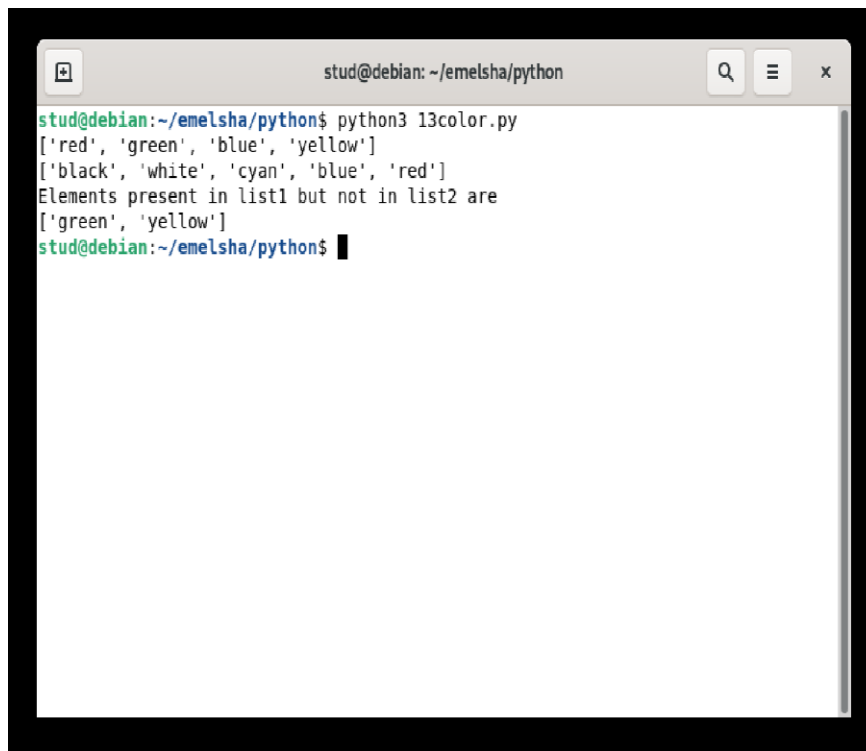
PROGRAM-14

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

Program code:

```
list1=["red","green","blue","yellow"]
list2=["black","white","cyan","blue","red"]
l3=[]
print(list1)
print(list2)
foriinlist1:
    ifnotinlist2:
        l3.append(i)
print("Elementspresentinlist1butnotinlist2are")
print(l3)
```

Output:



```
stud@debian:~/emelsha/python$ python3 13color.py
['red', 'green', 'blue', 'yellow']
['black', 'white', 'cyan', 'blue', 'red']
Elements present in list1 but not in list2 are
['green', 'yellow']
stud@debian:~/emelsha/python$
```

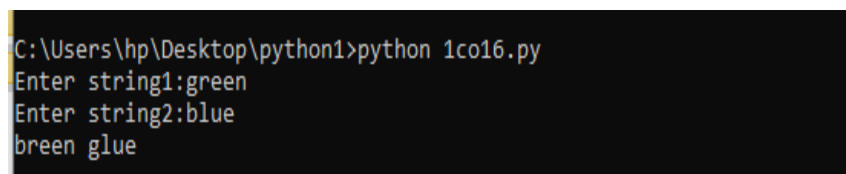
PROGRAM-15

Create a single string separated with space from 2 strings swapping the character at position.

Program code:

```
str1=input("Enter string1:")
str2=input("Enter string2:")
temp=str1[0]
str1=str1.replace(str1[0],str2[0])
str2=str2.replace(str2[0],temp)
str=str1+" "+str2
print(str)
```

Output:



```
C:\Users\hp\Desktop\python1>python 1co16.py
Enter string1:green
Enter string2:blue
green blue
```

PROGRAM-16

Sort a dictionary in ascending and descending order

Program code:

```
d={1:2,3:4,4:3,2:1,0:0}
list1=list(d.items())
dict=dict(list1)
print("Dictionary=",dict)
list1.sort()
print('Ascending order is',list1)
list1=list(d.items())
list1.sort(reverse=True)
print('Descending order is',list1)
```

Output:

```
C:\Users\hp\Desktop\python1>python 1co17.py
Dictionary= {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
Ascending order is [(0, 0), (1, 2), (2, 1), (3, 4), (4, 3)]
Descending order is [(4, 3), (3, 4), (2, 1), (1, 2), (0, 0)]
```

PROGRAM-17

Merge 2 dictionaries

Program code:

```
d2={"sex":"Female","qualification":"PG"}
d1.update(d2)
print(d)
```



```
stud@debian: ~/emelsha/python
stud@debian:~/emelsha/python$ python3 18dic.py
{'name': 'Ashna', 'age': '21', 'sex': 'Female', 'qualification': 'PG'}
```

PROGRAM-18

Find gcd of 2 numbers

Program code:

```
a=int(input("enter 1st number"))  
  
b=int(input("enter 2nd number"))  
z=min(a,b)  
for i in range(1,z+1):  
    if((a % i== 0)and(b % i==0)):  
        gcd=i  
print("gcd is=",gcd)
```

Output:



The screenshot shows a terminal window titled 'stud@debian: ~/emelsha/python'. The user has run the command 'python3 gcd.py'. The program prompts for the first number, which is entered as '12', and the second number, which is entered as '24'. The program then outputs 'gcd is = 12'. The terminal window has a search icon, a menu icon, and a close icon in the top right corner.

```
stud@debian:~/emelsha/python$ python3 gcd.py  
enter 1st number12  
enter 2nd number24  
gcd is = 12  
stud@debian:~/emelsha/python$
```


PROGRAM-19

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

Program code:

```
list=[12,13,14,15,16,21]
```

```
l1=[]
```

```
print(list)
```

```
print("New list")
```

```
for i in list:
```

```
    if i%2!=0:
```

```
        l1.append(i)
```

```
print(l1)
```

Output:

```
_____
```

```
[1, 3, 5, 7, 9, 11]
```

```
_____
```

COURSE OUTCOME 2**PROGRAM-1**

Program to find the factorial of a number

Program code:

```
fact=1
```

```
n=int(input('enter the value'))
```

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

```
    fact=fact*i
```

```
print(fact)
```

Output:

```
_____
```

```
enter the number4
```

```
24
```

```
_____
```

PROGRAM-2

Generate fibonacci series of N terms

Program code:

```
f1=0
f2=1
n=int(input('enter the number'))
print(f1)
print(f2)
for i in range(2,n):
    f3=f1+f2
    print(f3)
    f1=f2
    f2=f3
```

Output:

```
enter a number5
1
2
3
5
8
```

PROGRAM-3

Find the sum of all items in list?

Program code:

```
list=[1,2,3,4,5,6,7,8,9,10]
sum=0
for i in list:
    sum=sum +int(i)
print("sum:" ,sum)
```

Output:

```
55
```

PROGRAM-4

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

Program code:

```
limit1=1000
limit2=9999
list1=[]
for i in range(limit1,limit2):
    j=i
    digit=[]
    while(i!=0):
        digit.append(i%10)
        i=int(i/10)
    count=0
    for n in digit:
        if n%2==0:
            count=count+1
    if count==4:
        for k in range(31,100):
            if((k**2)==j):
                list1.append(j)
                print(k)
print(list1)
```

Output:

```
68
78
80
92
[4624, 6084, 6400, 8464]
```

PROGRAM-5

Display the given pyramid with step number accepted from user.

Program code:

```
l=int(input('Enter the limit:'))
for i in range(1,l+1):
    for j in range(1,i+1):
        c=i*j
        print(c,end=" ")
    print("\n")
```

Output:

```
Enter a number:4
1
2      4
3      6      9
4      8      12     16
```

PROGRAM-6

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

Program code:

```
string=input("Enter a string:")
list1=[]
for i in string:
    if i not in list1:
        list1.append(i)
for i in list1:
    count=0
    for j in string:
        if(i==j):
            count=count+1
    print(i,"\t:",count)
```

Output:

```
Enter a string:emelsha
e      : 2
m      : 1
l      : 1
s      : 1
h      : 1
a      : 1
```

PROGRAM-7

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

Program code:

```
string=input("Enter a string:")
if(string[-3:]=="ing"):
    string+="ly"
else:
    string+="ing"
print(string)
```

Output:

```
Enter a string:sleeping
sleepingly
```

PROGRAM-8

Accept a list of words and return length of longest word.

Program code:

```
lis=[]
n=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,n):
    lis.append(input(""))
longest=lis[0]
for i in range(1,n):
    if(len(lis[i])>len(longest)):
        longest=lis[i]
print("Length of longest word is",len(longest))
```

Output:

```
Enter the range:3
Enter the words:
cat
danger
fear
Length of longest word is 6
```

PROGRAM-9

Construct following patterns using nested loop

```

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
  
```

Program code:

```

k='*'
for i in range(1,6):
    for j in range(1,i+1):
        print(k,end=" ")
    print("\n")
for i in range(4,0,-1):
    for j in range(1,i+1):
        print(k,end=" ")
    print("\n")
:
  
```

Output:

```

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
  
```

PROGRAM-10

Generate all factors of a number.

Program code:

```
n=int(input("Enter a number:"))
print("Factors are")
for i in range(1,n+1):
    if(n%i==0):
        print(i)
```

Output:

```
Enter a number:100
Factors are
1
2
4
5
10
20
25
50
100
```

COURSE OUTCOME 3**PROGRAM-1**

Work with built-in packages

Create a package graphics with modules rectangle, circle and sub package 3D (td)- graphics with modules cuboid & 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 code:

Circle.py

```
from math import pi
def area_circle(radius):
    return pi*radius*radius
def perimeter_circle(radius):
    return 2*pi*radius
```

rectangle.py

```
def area_rec(length,width):
    return length*width
def perimeter_rec(length,width):
    return 2*(length+width)
```

```

sphere.py
from math import pi
def area_sphere(radius):
    return 4*(pi*radius*radius)
def perimeter_sphere(radius):
    return 2*pi*radius

```

```

cuboid.py
def area_cuboid(l,b,h):
    return 2*(l*h + b*h + l*b)
def volume_cuboid(l,b,h):
    return l*b*h

```

```

driver1.py
import Graphics1
from Graphics1 import circle,rectangle
from Graphics1.tdgraphics import cuboid,sphere
from Graphics1.circle import *
print("Area of a circle with radius 10 is : ",circle.area_circle(10))
print("Perimeter of a circle with radius 10 is ",circle.perimeter_circle(10))
print("\n")
print("Area of a Rectangle with length and width 10 is :
",rectangle.area_rec(10,10))
print("Perimeter of a Rectangle with length and width 10 is :
",rectangle.perimeter_rec(10,10))
print("\n")
print("Area of a cuboid with length,width,height 10 is :
",cuboid.area_cuboid(10,10,10))
print("Volume of a cuboid with length,width,height 10 is :
",cuboid.volume_cuboid(10,10,10))
print("\n")
print("Area of a sphere with radius 10 is : ",sphere.area_sphere(10))
print("Perimeter of a sphere with radius 10 is ",sphere.perimeter_sphere(10))

```

Output:


```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd Desktop

C:\Users\ASUS\Desktop>cd python

C:\Users\ASUS\Desktop\python>md Graphics1

C:\Users\ASUS\Desktop\python>cd graphics1

C:\Users\ASUS\Desktop\python\Graphics1>notepad circle.py

C:\Users\ASUS\Desktop\python\Graphics1>notepad rectangle.py

C:\Users\ASUS\Desktop\python\Graphics1>md tdgraphics

C:\Users\ASUS\Desktop\python\Graphics1>cd tdgraphics

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>notepad cuboid.py

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>notepad sphere.py

C:\Users\ASUS\Desktop\python\Graphics1\tdgraphics>cd..

C:\Users\ASUS\Desktop\python\Graphics1>cd..
```

```
C:\Users\ASUS\Desktop\python>notepad driver1.py

C:\Users\ASUS\Desktop\python>python driver1.py
Area of a circle with radius 10 is : 314.1592653589793
Perimeter of a circle with radius 10 is 62.83185307179586

Area of a Rectangle with length and width 10 is : 100
Perimeter of a Rectangle with length and width 10 is : 40

Area of a cuboid with length,width,height 10 is : 600
Volume of a cuboid with length,width,height 10 is : 1000

Area of a sphere with radius 10 is : 1256.6370614359173
Perimeter of a sphere with radius 10 is 62.83185307179586

C:\Users\ASUS\Desktop\python>
```

COURSE OUTCOME 4**PROGRAM-1**

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

Program code:

```
class Rectangle:
def __init__(self,length,breadth):
self.length = length
self.breadth = breadth
def area(self):
return self.length * self.breadth
def perimeter(self):
return 2*(self.length + self.breadth)
c=int(input("enter length of 1st rectangle"))
d=int(input("enter breadth 1st rectangle"))
u=int(input("enter length of 2nd rectangle"))
v=int(input("enter breadth of 2nd rectangle"))
r1= Rectangle(c,d)
r3= Rectangle(u,v)
a=r1.area()
b=r3.area()
print("area of 1st rectangle is:",a)
print("perimeter is:",r1.perimeter())
print("area of 2nd rectangle is:",b)
if (a>b):
print("1st is greater")
else:
    print("2nd is greater")
```

Output:

```
enter length of 1st rectangle3
enter breadth 1st rectangle4
enter length of 2nd rectangle5
enter breadth of 2nd rectangle7
area of 1st rectangle is: 12
perimeter is: 14
area of 2nd rectangle is: 35
2nd is greater
```

PROGRAM-2

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 code:

```
class Bank:
def __init__(self,acc_no,name,type_of_acc,balance):
self.acc_no= acc_no
self.name=name
self.type_of_acc=type_of_acc
self.balance=balance
def deposit(self,x):
self.balance=self.balance+x
print("balance after deposit is=",self.balance)
def withdraw(self,y):
self.balance=self.balance-y
print("balance after withdrawal is=",self.balance)
x=int(input("amount to be deposited"))
y=int(input("amount to withdraw"))
ob1=Bank(1,"aaa","ccc",300000)
ob2=Bank(2,"bbb","ccc",500000)
ob1.deposit(x)
ob1.withdraw(y)
ob2.deposit(x)
ob2.withdraw(y)
```

Output:

```
amount to be deposited20000
amount to withdraw30000
balance after deposit is= 320000
balance after withdrawal is= 290000
balance after deposit is= 520000
balance after withdrawal is= 490000
```

PROGRAM-3

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

Program code:

```
class Rectangle:
def __init__(self,length,breadth):
self.length = length
self.breadth = breadth
def area(self):
return self.length * self.breadth
def perimeter(self):
return 2*(self.length + self.breadth)
def __lt__(self,rr):
if (self.length >rr.length and self.breadth > rr.breadth):
print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
c=int(input("enter length of 1st rectangle"))
d=int(input("enter breadth 1st rectangle"))
u=int(input("enter length of 2nd rectangle"))
v=int(input("enter breadth of 2nd rectangle"))
r1= Rectangle(c,d)
r3= Rectangle(u,v)
a=r1.area()
b=r3.area()
print("area of 1st rectangle is:",a)
print("perimeter is:",r1.perimeter())
print("area of 2nd rectangle is:",b)
    r1 < r3
```

Output:

```
enter length of 1st rectangle2
enter breadth 1st rectangle3
enter length of 2nd rectangle1
enter breadth of 2nd rectangle5
area of 1st rectangle is: 6
perimeter is: 10
area of 2nd rectangle is: 5
Area of second rectangle is greater
```

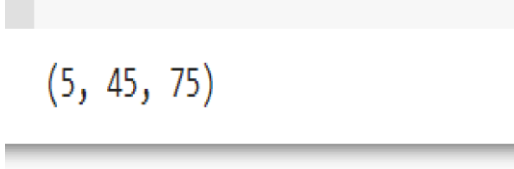
PROGRAM-4

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

Program code:

```
class Time:
    def __init__(self,hr,min,sec):
        self.hr=hr
        self.min=min
        self.sec=sec
    def __add__(self,t):
        return(self.hr+t.hr,self.min+t.min,self.sec+t.sec)
t1=Time(3,20,35)
t2=Time(2,25,40)
print(t1+t2)
```

Output:



(5, 45, 75)

PROGRAM-5

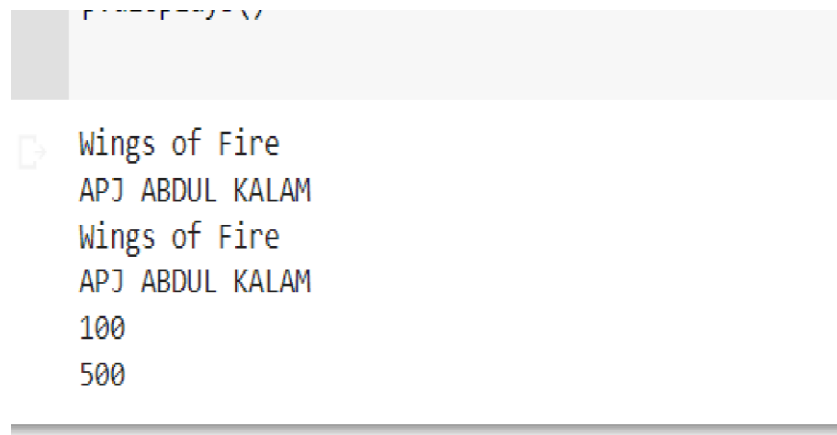
Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Program code:

```
class Publisher(object):
    def __init__(self,name):
        self.name=name
    def display1(self):
        print(self.title)
        print(self.author)
class Book(Publisher):
    def __init__(self,name,title,author):
        super().__init__(name)
        self.title=title
```

```
self.author=author
def display2(self):
    super().display1()
    print(self.title)
    print(self.author)
class Python(Book):
    def __init__(self,name,title,author,price,no_of_pages):
        super().__init__(name,title,author)
        self.price=price
        self.no_of_pages=no_of_pages
    def display3(self):
        super().display2()
        print(self.price)
        print(self.no_of_pages)
p=Python("XYZ Publications","Wings of Fire","APJ ABDUL
KALAM",100,500)
p.display3()
```

Output:



```
Wings of Fire
APJ ABDUL KALAM
Wings of Fire
APJ ABDUL KALAM
100
500
```

COURSE OUTCOME 5**PROGRAM-1**

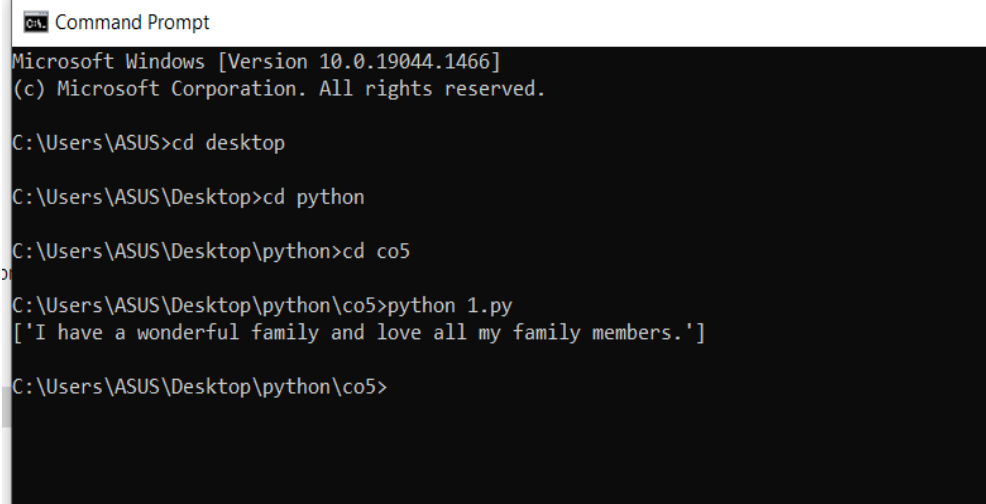
Write a python program to read a file line by line and store it into a list.

Program code:

```
fp=open("text.txt",'r')
lines=[]
for line in fp:
    lines.append(line.strip())
print(lines)
text.txt
```

I have a wonderful family and love all my family members.

Output:



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd desktop
C:\Users\ASUS\Desktop>cd python
C:\Users\ASUS\Desktop\python>cd co5
C:\Users\ASUS\Desktop\python\co5>python 1.py
['I have a wonderful family and love all my family members.']
C:\Users\ASUS\Desktop\python\co5>
```


PROGRAM-2

Write a python program to read each row from a given csv file and print a list of strings

Program code:

```
import csv
with open('work.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

work.csv

	A	B	C	D
1	Name	Age	Profession	
2	Das	40	Manager	
3	Vinu	38	Ass.Manager	
4	Manu	35	Staff	
5	Janaki	30	Nurse	
5				

Output:

```

C:\> Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd desktop

C:\Users\ASUS\Desktop>cd python

C:\Users\ASUS\Desktop\python>cd co5

C:\Users\ASUS\Desktop\python\co5>python 2.py
['Name', 'Age', 'Profession']
['Das', '40', 'Manager']
['Vinu', '38', 'Ass.Manager']
['Manu', '35', 'Staff']
['Janaki', '30', 'Nurse']

C:\Users\ASUS\Desktop\python\co5>
  
```

