

**FEDERAL INSTITUTE OF SCIENCE AND
TECHNOLOGY (FISAT)TM
HORMIS NAGAR, MOOKKANNOOR**

ANGAMALY-683577



'FOCUS ON EXCELLENCE'

LABORATORY RECORD
20MCA131 - PROGRAMMING LAB

Name: GOPIKA BABU

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch:** 2021 A **Roll No:** 60

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY
(FISAT)TM**

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577



‘FOCUS ON EXCELLENCE’

Name : GOPIKA BABU

Branch : MASTER OF COMPUTER APPLICATIONS

Semester : 1

Roll No: 60

University Exam.Reg. No: FIT21MCA-2060

CERTIFICATE

Certified that this is the Bonafide record of the Practical work done by Mrs. **GOPIKA BABU** in the **20MCA131- PROGRAMMING** Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.

Signature of Staff in Charge

Signature of H.O.D

Name:

Name:

Date:

Date of University practical examination

Signature of

Signature of

Internal Examiner

External Examiner

CONTENT

SI No:	Date :	Name of Experiment:	Page No:	Signature of Staff –In – Charge:
1	28/10/2021	Display future leap years from current year to a final year entered by user.	7	
2	28/10/2021	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)	7	
3	28/10/2021	Count the occurrences of each word in a line of text.	9	
4	28/10/2021	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	10	
5	10/11/2021	Store a list of first names. Count the occurrences of 'a' within the list	10	
6	10/11/2021	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.	11	
7	10/11/2021	Get a string from an input string where all occurrences of first character replaced with '\$', except first character	13	
8	10/11/2021	Create a string from given string where first and last characters exchanged. [eg: python -> nythop]	13	
9	10/11/2021	Accept the radius from user and find area of circle.	14	
10	11/11/2021	Find biggest of 3 numbers entered.	14	
11	11/11/2021	Accept a file name from user and print extension of that.	14	

12	11/11/2021	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	15	
13	11/11/2021	Accept an integer n and compute n+nn+nnn.	15	
14	11/11/2021	Print out all colors from color-list1 not contained in color-list2.	16	
15	17/11/2021	Create a single string separated with space from two strings by swapping the character at position 1.	16	
16	17/11/2021	Merge two dictionaries.	17	
17	17/11/2021	Find gcd of 2 numbers.	17	
18	17/11/2021	From a list of integers, create a list removing even numbers.	17	
CO2				
1	25/11/2021	Program to find the factorial of a number	19	
2	25/11/2021	Generate Fibonacci series of N terms	19	
3	25/11/2021	Find the sum of all items in a list	20	
4	25/11/2021	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	20	
5	02/12/2021	Display the given pyramid with step number accepted from user.	21	
6	02/12/2021	Count the number of characters (character frequency) in a string.	22	
7	02/12/2021	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.	22	

8	09/12/2021	Accept a list of words and return length of longest word.	23	
9	09/12/2021	Construct following pattern using nested loop. * * * * * * * * * * * * * * * * * * * * * * * * *	23	
10	09/12/2021	Generate all factors of a number.	24	
CO3				
1	29/01/2022	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	
CO4				
1	13/01/2022	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	28	
2	13/01/2022	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.	29	

3	13/01/2022	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	31	
4	20/01/2022	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	32	
5	20/01/2022	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.	33	
CO5				
1	03/02/2022	Write a Python program to read a file line by line and store it into a list.	34	
2	03/02/2022	Write a Python program to read each row from a given csv file and print a list of strings.	35	

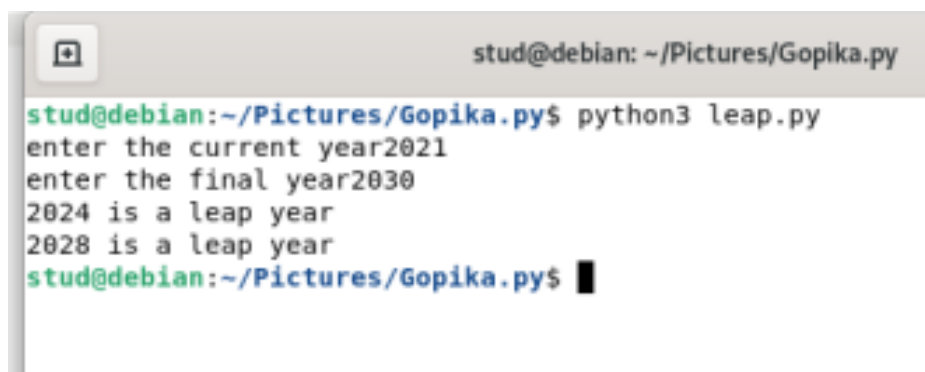
COURSE OUTCOME 1

- 1) Display future leap years from current year to a final year entered by User.

Source code

```
print("print leap year  
between two given years");  
startyear=2021  
endyear=int(input("Enter end year")) print("list of leap years")  
for year in  
    range(startyear,endyear  
): if(0==year%4):  
    print(year)
```

Output



```
stud@debian: ~/Pictures/Gopika.py  
stud@debian:~/Pictures/Gopika.py$ python3 leap.py  
enter the current year2021  
enter the final year2030  
2024 is a leap year  
2028 is a leap year  
stud@debian:~/Pictures/Gopika.py$
```

- 2) List comprehensions:

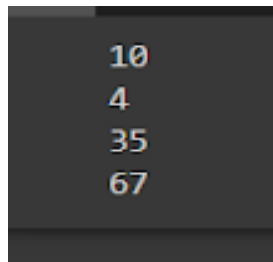
- a. Generate positive list of numbers from a given list of integers.

Source code

```
list=[-11,4,8,-34,10,14]
```

```
print("Elements in the list are:",list) print("Positive numbers in the list")
for num in list:
    if num>=0:
        print(num)
```

Output



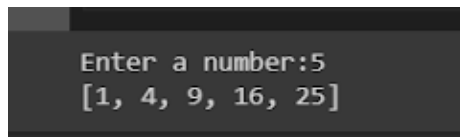
```
10
4
35
67
```

b. Square of N numbers

Source code

```
n=int(input('Enter range:'))
for num in range(1,n+1):
    num=num*num
    print(num)
```

Output



```
Enter a number:5
[1, 4, 9, 16, 25]
```

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

Source code

```
s=input("Enter a string: ")
list=[]
for i in s:
    if i in "aeiouAEIOU":
        list.append(i)
print("vowels in the list are:")
print(list)
```


Output

```
L=[]
s="India is my country"
for i in s:
    if i in ("aeiouAEIOU"):
        L.append(i)
print(L)

['I', 'i', 'a', 'i', 'o', 'u']
```

d. List ordinal values of each element of a word.

Source code

```
print("String: Welcome")
print("Ordinal Values")
for i in 'W','e','l','c','o','m','e':
    x=ord(i)
    print(x)
```

Output

```
Enter a word:python
The ASCII value of the letters in the word is
112
121
116
104
111
110
```

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

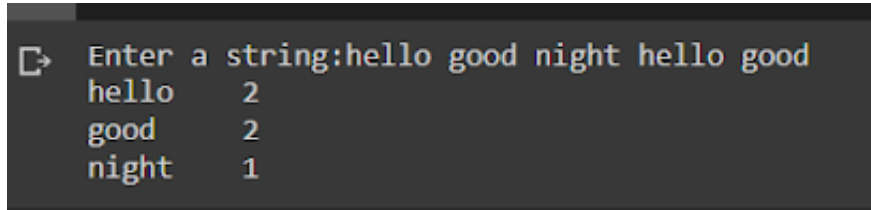
Source code

```
list1=[]
list2=[]
x=input("Enter a line of text:")
for i in x.split(" "):
    list1.append(i)
```

```
        if i not in list2:
            list2.append(i)

for i in list2:
    print(i,"\t",list1.count(i))
```

Output



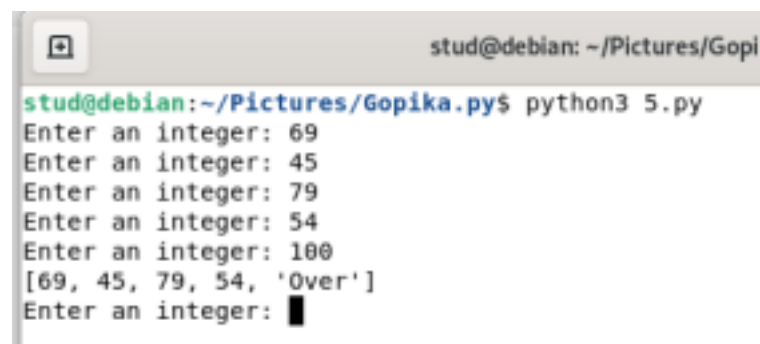
```
Enter a string:hello good night hello good
hello      2
good       2
night      1
```

- 4) Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

Source code

```
list=[]
while True:
    n=int(input('Enter an integer: '))
    if(n<=100):
        list.append(n)
    else:
        list.append('over')
print(list)
```

Output



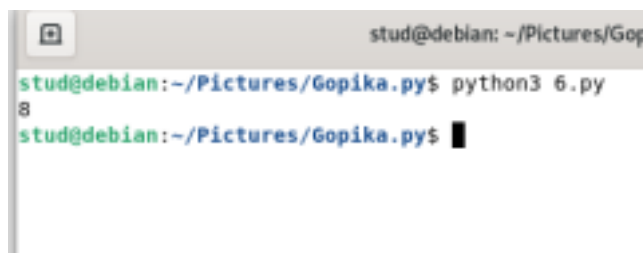
```
stud@debian: ~/Pictures/Gopi
stud@debian:~/Pictures/Gopika.py$ python3 5.py
Enter an integer: 69
Enter an integer: 45
Enter an integer: 79
Enter an integer: 54
Enter an integer: 100
[69, 45, 79, 54, 'Over']
Enter an integer: █
```

- 5) Store a list of first names. Count the occurrences of 'a' within the list.

Source code

```
list=['anjana','aparna','anna'] print("Elements in the list are:")
print(list)
count=0
for word in list:
    for i in word:
        if i=='a':
            count+=1
print("count of 'a' is:", count)
```

Output



```
stud@debian: ~/Pictures/Gopika.py
stud@debian:~/Pictures/Gopika.py$ python3 6.py
8
stud@debian:~/Pictures/Gopika.py$
```

6) Enter 2 lists of integers.Check

a. whether list are of same length

```
l1=[2,4,8]
l2=[3,5,7,9]
x=len(l1)
y=len(l2)
if(x==y):
    print('same')
else:
    print('not same')
```

Output



```
stud@debian: ~/Pictures/Gopika.py
stud@debian:~/Pictures/Gopika.py$ python3 7a.py
not same
stud@debian:~/Pictures/Gopika.py$
```

b. whetherlist sums of same value

```
l1=[2,4,8]
l2=[3,5,7,9]
s=0
for i in l1:
    s=s+i
for i in l2:
    t=0
    t=t+i
if(s==t):
    print('same')
else:
    print('not same')
```

Output

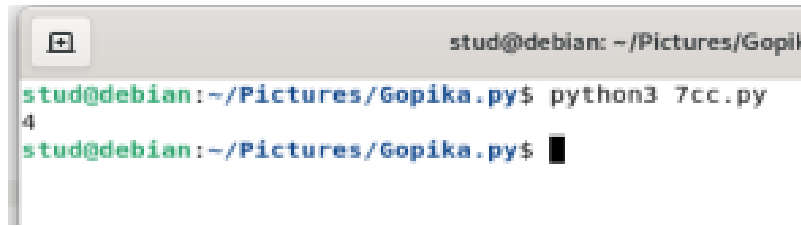


```
stud@debian: ~/Pictures/G
stud@debian:~/Pictures/Gopika.py$ python3 7b.py
not same
stud@debian:~/Pictures/Gopika.py$
```

c. whether any value occur in both.

```
l1=[8,4,9]
l2=[1,2,3,4,5]
f=0
for i in l1:
    if(i in l2):
        print(i)
        f=f+1
if(f==0):
    print('no common elements')
```

Output



```
stud@debian: ~/Pictures/Gopika.py$ python3 7cc.py
stud@debian:~/Pictures/Gopika.py$
```

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

Source code

```
str=input("Enter a string: ")
print("Original string is: ",str)
char=str[0]
str=str.replace(char,'$')
str=char+str[1:]
print("String: ",str)
```

Output



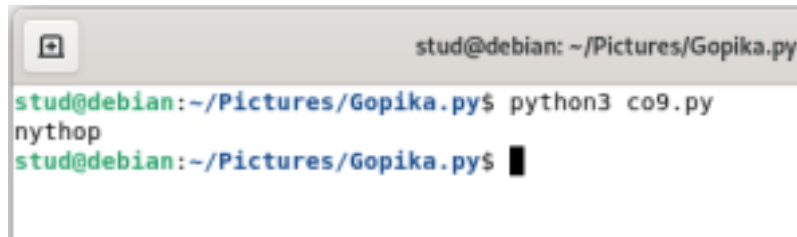
```
stud@debian:~/Pictures/Gopika.py$ python3 co8.py
enter the string:ONION
original string: ONION
replaced string: ONI$n
stud@debian:~/Pictures/Gopika.py$
```

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

Source code

```
s=input("Enter a string: ")
t=s[0]
t1=s[-1]
n=len(s)
ns=t1+s[1:n-1]+t
print(ns)
```

Output



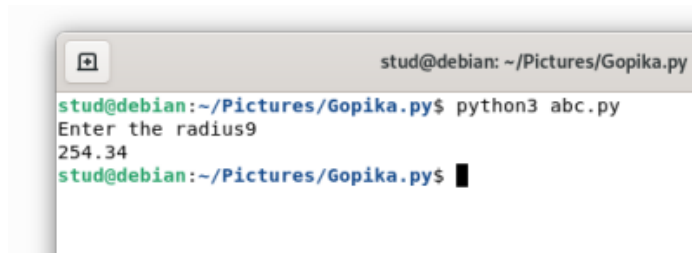
```
stud@debian: ~/Pictures/Gopika.py
stud@debian:~/Pictures/Gopika.py$ python3 co9.py
nythop
stud@debian:~/Pictures/Gopika.py$
```

9) Accept the radius from the user and find the area of the circle.

Source code

```
r=int(input('Enter the radius: '))
A=3.14*r*r
print(A)
```

Output



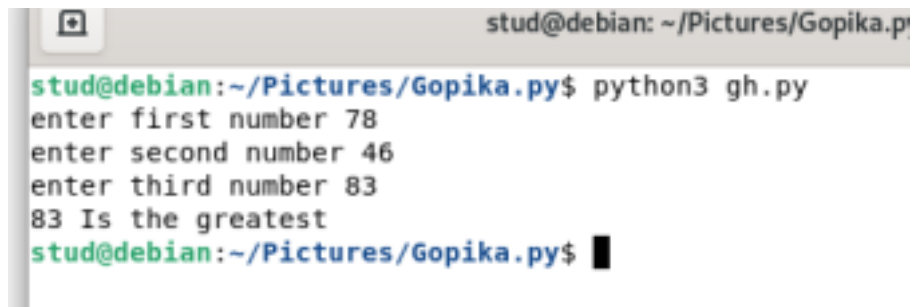
```
stud@debian: ~/Pictures/Gopika.py
stud@debian:~/Pictures/Gopika.py$ python3 abc.py
Enter the radius9
254.34
stud@debian:~/Pictures/Gopika.py$
```

10) Find the biggest of 3 numbers

Source code

```
a=int(input('Enter first number:'))
b=int(input('Enter second number:'))
c=int(input('Enter third number:'))
if a>b and a>c:
    print(a)
if b>a and b>c:
    print(b)
if c>a and c>b:
    print(c)
```

Output



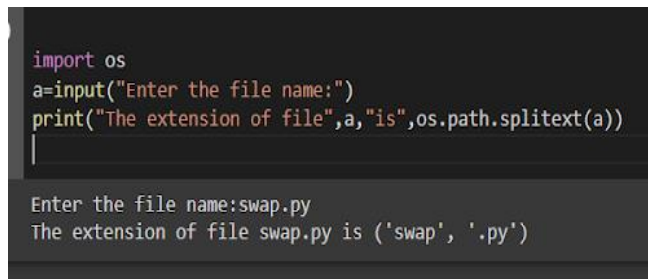
```
stud@debian: ~/Pictures/Gopika.py$ python3 gh.py
enter first number 78
enter second number 46
enter third number 83
83 Is the greatest
stud@debian:~/Pictures/Gopika.py$
```

11) Accept a file name from user and print extension of that.

Source code

```
import os
a=input("Enter file name:")
print("The extension of file",a,"is",os.path.splitext(a))
```

Output



```
import os
a=input("Enter the file name:")
print("The extension of file",a,"is",os.path.splitext(a))
|
Enter the file name:swap.py
The extension of file swap.py is ('swap', '.py')
```

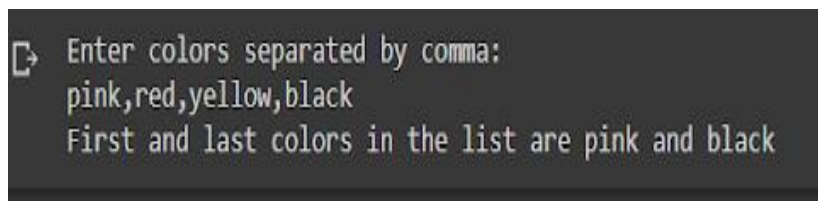
12) Create a list of colors from comma-separated color names entered by user.

Display first and last colors.

Source code

```
colors=[]
str=(input("Enter color names:"))
for i in str.split(','):
    colors.append(i)
print(colors)
print("first color:",colors[0],"Last color:",colors[-1])
```

Output



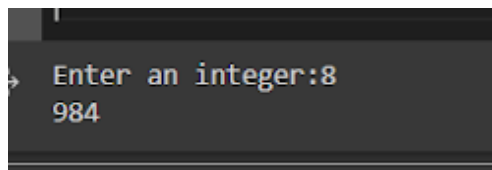
```
Enter colors separated by comma:
pink,red,yellow,black
First and last colors in the list are pink and black
```

13) Accept an integer n and compute n+nn+nnn.

Source code

```
n=int(input("Enter the number:"))
a=n*1
b=n*11
c=n*111
s=a+b+c
print(n,"+",n,"*",n,"+",n,"*",n,"*",n,"=",s)
```

Output



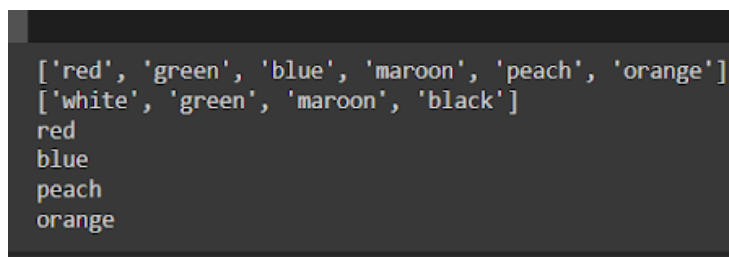
```
Enter an integer:8
984
```

14) Print out all color from color-list1 not contained in color-list2

Source code

```
l1=["red","green","blue","maroon","peach","orange"]
l2=["white","green","maroon","black"]
print(l1)
print(l2)
for i in l1:
    if i not in l2:
        print(i)
```

Output



```
['red', 'green', 'blue', 'maroon', 'peach', 'orange']
['white', 'green', 'maroon', 'black']
red
blue
peach
orange
```

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

Source code

```
str1=input("Enter first string:")
str2=input("Enter second string:")
str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]
print(str3)
```

Output


```
> Enter first string:Computer
Enter second string:Architecture
Aomputer Crchitecture
```

16) Merge two dictionaries.

Source code

```
dict1={"Name":"Rose","Age":25}
dict2={"Gender":"F","Qualification":"PG"}
dict1.update(dict2)
print(dict1)
```

Output

```
dict1={"Name":"Rose","Age":25}
dict2={"Gender":"F","Qualification":"PG"}
dict1.update(dict2)
print(dict1)

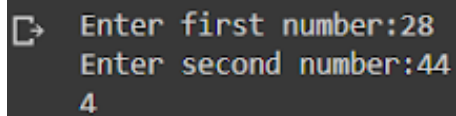
{'Name': 'Rose', 'Age': 25, 'Gender': 'F', 'Qualification': 'PG'}
```

17) Find gcd of 2 numbers

Source code

```
a=int(input("Enter first number: "))
b=int(input("Enter first number: "))
x=min(a,b)
gcd=0
for i in range (1,x+1):
    if((a%x==0) and (b%x==0)):
        gcd=i
print("GCD is",i)
```

Output



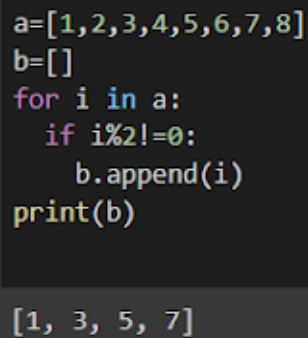
```
Enter first number:28
Enter second number:44
4
```

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

Source code

```
a=[1,2,3,4,5,6,7,8]
b=[]
for i in a:
    if i%2!=0:
        b.append(i)
print(b)
```

Output



```
a=[1,2,3,4,5,6,7,8]
b=[]
for i in a:
    if i%2!=0:
        b.append(i)
print(b)

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

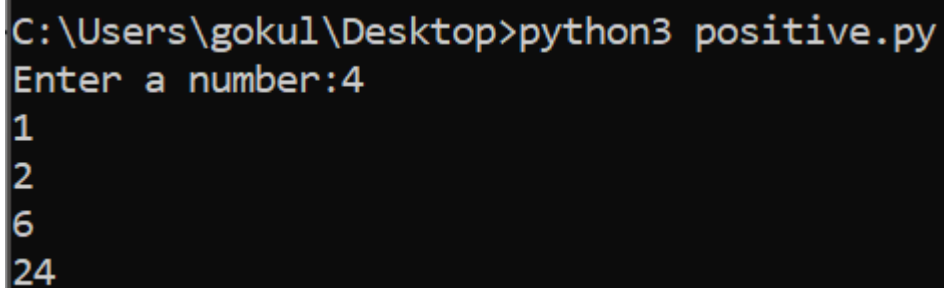
COURSE OUTCOME 2

1) Program to find the factorial of a number.

Source code

```
n=int(input('Enter a number:'))  
fact=1  
for i in range (1,n+1):  
    fact=fact*i  
print(fact)
```

Output



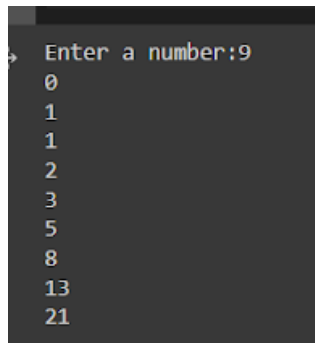
```
C:\Users\goku1\Desktop>python3 positive.py  
Enter a number:4  
1  
2  
6  
24
```

2) Generate fibonacci series of N terms.

Source code

```
n=int(input("Enter a number:"))
f1=0
f2=1
print(f1)
print(f2)
for i in range(0,n-2):
    f3=f1+f2
    print(f3)
    f1=f2
    f2=f3
```

Output



```
Enter a number:9
0
1
1
2
3
5
8
13
21
```

3) Find the sum of all items in a list.

Source code

```
list1=[1,2,3,4,5]
sum=0
for i in list1:
    sum=sum+i
print(sum)
```

Output

```
list1=[1,2,3,4,5]
sum=0
for i in list1:
    sum=sum+i
print(sum)

15
```

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

Source 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]
```

5) Display the given pyramid with step number accepted from user.

Source code

```
n=int(input("Enter a number:"))
for j in range(0,n+1):
    for i in range(1,j+1):
        i=j*i
        print(i,end=" ")
    print("\n")
```

Output

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

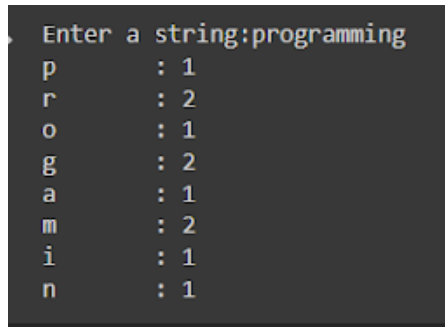
6) Count the number of characters (character frequency) in a string.

Source 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



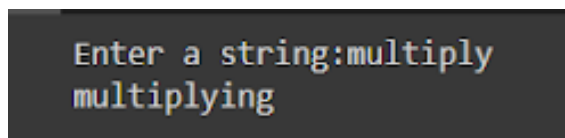
```
Enter a string:programming
p      : 1
r      : 2
o      : 1
g      : 2
a      : 1
m      : 2
i      : 1
n      : 1
```

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

Source code

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

Output



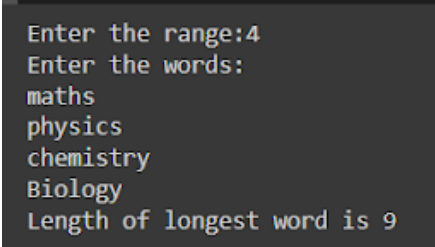
```
Enter a string:multiply
multiplying
```

- 8) Accept a list of words and return length of longest word.

Source 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

A screenshot of a terminal window showing the output of the Python program. The text is as follows:

```
Enter the range:4
Enter the words:
maths
physics
chemistry
Biology
Length of longest word is 9
```

9) Construct following pattern using nested loop.

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

Source code

```
for i in range(1,6):
```



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

Output



10) Generate all factors of a number.

Source 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:46
Factors are
1
2
23
46
```

COURSE OUTCOME 3

- 1) 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)

Source code

Graphice\circle.py

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

Graphics\rectangle.py

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

Graphics\tdgraphics\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
```

Graphics\tdgraphics\sphere.py

```
from math import pi
def area_sphere(radius):
    return 4*(pi*radius*radius)
def perimeter_sphere(radius):
    return 2*pi*radius
```

graphics.py (driver code)

```
import Graphics
from Graphics import circle,rectangle
from Graphics.tdgraphics import cuboid,sphere
from Graphics.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

```
Area of rectangle :144
Area of circle :36
Area of sphere :1808.6399999999999
Area of cuboid :95551488
Perimeter of rectangle :48
Perimeter of circle :37.68
Diameter of sphere :24
Periameter of cuboid :184
```

COURSE OUTCOME 4

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

Source 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)

l=int(input("Enter length of rectangle1: "))

b=int(input("Enter breadth of rectangle1: "))

rect1 = Rectangle(l,b)

a1=rect1.area()

p1=rect1.perimeter()

print("Area:",a1)

print("Perimeter:",p1)

l=int(input("Enter length of rectangle2: "))

b=int(input("Enter breadth of rectangle2: "))

rect2 = Rectangle(l,b)

a2=rect2.area()

p2=rect2.perimeter()

print("Area:",a2)

print("Perimeter:",p2)

if (a1>a2):

    print("First rectangle is larger")
```

```
elif a1==a2:

    print("Rectangles are of same area")

else:

    print("Second rectangle is larger")
```

Output

```
Enter length of rectangle1: 5
Enter breadth of rectangle1: 7
Area: 35
Perimeter: 24
Enter length of rectangle2: 8
Enter breadth of rectangle2: 2
Area: 16
Perimeter: 20
First rectangle is larger
```

- 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.**

Source code

```
class bank:

    def __init__(self,acc_no,name,acc_type,bal):

        self.acc_no=acc_no

        self.name=name

        self.acc_type=acc_type

        self.bal=bal

    def deposit(self):

        self.bal=self.bal+y

        return self.bal

    def withdraw(self):

        return self.bal-y
```

```
def display_balance(self):  
    return self.bal  
  
acc1=bank("b11","Ann","Savings",50000)  
  
while(1):  
    print("1.Deposit\n2.Withdraw\n3.Display balance\n4.Exit\n")  
    ch=int(input("Enter your choice:"))  
    if ch==1:  
        amt=int(input("Enter the amount:"))  
        b=acc1.deposit(amt)  
        print("Current balance:",b)  
    elif ch==2:  
        amt=int(input("Enter the amount:"))  
        b=acc1.withdraw(amt)  
        print("Current balance:",b)  
    elif ch==3:  
        cb=acc1.display_balance()  
        print("Current balance:",cb)  
    elif ch==4:  
        exit(1)  
    else:  
        print("Invalid choice")
```

Output

```
2435 anju sbi 21000  
5436 aju federal 19500
```

3) Create a class Rectangle with private attributes length and width.

Overload '<' operator to compare the area of 2 rectangles.

Source code

```
class Rectangle:
    def __init__(self,length,breadth):
        self.__length = length
        self.__breadth = breadth
    def __lt__(self,rect2):
        if self.__length*self.__breadth < rect2.__length*rect2.__breadth:
            return True
        else:
            return False

l=int(input("Enter length of rectangle1: "))
b=int(input("Enter breadth of rectangle1: "))
rect1 = Rectangle(l,b)

l=int(input("Enter length of rectangle2: "))
b=int(input("Enter breadth of rectangle2: "))
rect2 = Rectangle(l,b)

if rect1 < rect2:
    print("Second rectangle is larger")
else:
    print("First rectangle is larger")
```

output


```
) Enter length of the first rectangle:12
Enter breadth of the first rectangle:4
Enter length of the second rectangle:8
Enter breadth of the second rectangle:6
Perimeter of first rectangle= 32
Perimeter of second rectangle= 28
Least one is: 48
```

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

Source code

```
class Time:
    def __init__(self,h,m,s):
        self.__hour=h
        self.__minute=m
        self.__second=s
    def __add__(self,ob):
        hour=self.__hour+ob.__hour
        minute=self.__minute+ob.__minute
        second=self.__second+ob.__second
        t=Time(hour,minute,second)
        return t

    def print_it(self):
        print("Hour :",self.__hour)
        print("Minute :",self.__minute)
        print("Second :",self.__second)

t1=Time(10,10,10)
t2=Time(20,20,20)
t3=t1+t2
t3.print_it()
```

Output

- 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.

Source 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("ABC Publications","Gaming Python","Gokul",100,500)
p.display3()
q=Python("XYZ Publications","Java programming","E
Balagurusami",500,1200)
q.display3()
```

Output

```
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 100
Number of Pages : 500
This Fuction is a member fuction of class Publisher
```

COURSE OUTCOME 5

- 1) Write a Python program to read a file line by line and store it into a list.

Source code

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

Output

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn1.py
["Cats, also called domestic cats are small, carnivorous mammals, of the family Felidae.", "Domestic cat
s are often called 'house cats' when kept as indoor pets.", 'Cats have been domesticated for nearly 10,00
0 years.', 'They are one of the most popular pets in the world."']
PS C:\Users\HP\OneDrive\Desktop\python\co5> █
```

- 2) Write a Python program to read each row from a given csv file and print a list of strings.

Source code

```
import csv

with open('people.csv', 'r') as file:

    reader = csv.reader(file)

    for row in reader:

        print(row)
```

Output

```
PS C:\Users\HP\OneDrive\Desktop\python\co5> python qn2.py
['Name', 'Designation', 'Salary']
['Jessy', 'Manager', '90000']
['Tom', 'Clerk', '40000']
['Alfred', 'Assistant Manager', '70000']
PS C:\Users\HP\OneDrive\Desktop\python\co5> █
```