**CO 2 PROGRAMS**

**1 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**



**2 :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 = " ")

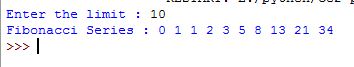
count += 1

a = b

b = sum

sum = a + b

**Output**



**3 Find the sum of all items in a list**

list1 = [10, 15, 20, 25, 30]

total = sum(list1)

print("Sum of list : ",total)

**Output**



**4 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**



# CO 2 :Pgrm 5

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

rows = int(input("Enter the number of rows: "))

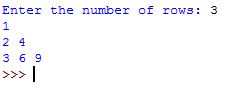
for i in range(1, rows+1):

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

print(i \* j, end=' ')

print()

**Output**



# CO 2 :Pgrm 7

**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")):

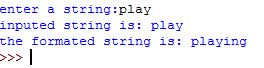
str=str+'ly'

else:

str=str+'ing'

print("the formated string is:",str)

**Output**



# CO 2 :Pgrm 8

**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):

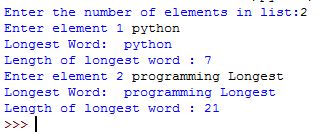
max1=len(i)

temp=i

print("Longest Word:",temp)

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

**Output**



# CO 2 :Pgrm 9

**Construct following pattern using nested loop**

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**

