**Experiment 3**

**Functions & Lambda Functions**

1. Demonstrate use of round(), floor()  and ceil () in python.
2. Write a python program to display Fibonacci series using recursion.
3. Create a lambda function that takes a list of numbers as input and returns back a list of even numbers. (use filter ())
4. Given three sequences of numbers, write a lambda expression using map and zip functions, that will calculate element-wise maximum among these 3 sequences.

**1.**

import math

num = float(input("Enter number: "))

# round

num\_round = round(num, 2)

print(num\_round)

# floor

num\_floor = math.floor(num)

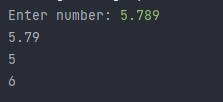
print(num\_floor)

# ceil

num\_ceil = math.ceil(num)

print(num\_ceil)

**OUTPUT:**

****

**2.**

def fibo(n):

if n <= 0:

return 0

elif n == 1:

return 1

else:

return fibo(n - 1) + fibo(n - 2)

terms = int(input("Enter no. of terms: "))

if terms <= 0:

print("Enter positive integer!")

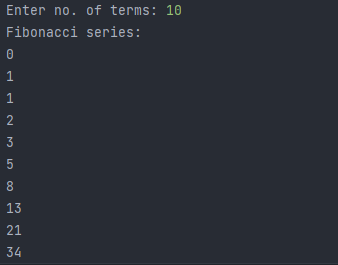
else:

print("Fibonacci series: ")

for i in range(terms):

print(fibo(i))

**OUTPUT:**



**3.**

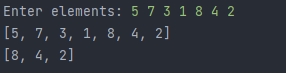
lst = list(map(int,input("Enter elements: ").split()))

print(lst)

even = list(filter(lambda x : x%2 == 0,lst))

print(even)

**OUTPUT:**

****

**4.**

seq1 = [1, 2, 3, 4, 5]

seq2 = [6, 7, 8, 9, 10]

seq3 = [11, 12, 13, 14, 15]

max\_seq = list(map(lambda x, y, z: max(x, y, z), seq1, seq2, seq3))

print(max\_seq)

**OUTPUT:**

