**1. Write a Python Program to Display Fibonacci Sequence using Recursion?**

I/P:

**def** genFibonacci(n,a,b):

**if** n **==** 0:

**return** 1

**else**:

result **=** a**+**b

print(result, end**=**', ')

genFibonacci(n**-**1,b,result)

in\_num **=** int(input('Enter the length of Series: '))

print('0, 1',end**=**', ')

genFibonacci(in\_num,1,2)

O/P:

Enter the length of Series: 20

0, 1, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657,

**2. Write a Python Program to Find Factorial of a Number using Recursion?**

I/P:

**def** factorial(num):

**if** (num **<** 1):

**return** 1

**else**:

**return** num**\***factorial(num**-**1)

num **=** int(input('Enter a number: '))

value **=** factorial(num)

print(f'The Factorial of {num} is {value}')

O/P:

Enter a number: 5

The Factorial of 5 is 120

**3. Write a Python Program to Calculate your Body Mass Index?**

I/P:

**def** calculateBMI():

in\_weight **=** eval(input('Enter your Weight(kgs): '))

in\_height **=** eval(input('Enter your Height(mts): '))

calc\_bmi **=** in\_weight**/**pow(in\_height,2)

**if** (calc\_bmi **<** 18.5):

status **=** 'Underweight'

**elif** (calc\_bmi **>=** 18.5 **and** calc\_bmi **<** 24.9):

status **=** 'Healthy'

**elif** (calc\_bmi **>=** 24.9 **and** calc\_bmi **<** 30):

status **=** 'Overweight'

**elif** (calc\_bmi **>=**30):

status **=** 'Suffering from Obesity'

print(f'Your BMI is {calc\_bmi} and status is {status} ')

calculateBMI()

O/P:

Enter your Weight(kgs): 70

Enter your Height(mts): 1.8

Your BMI is 21.604938271604937 and status is Healthy

**4. Write a Python Program to Calculate the Natural Logarithm of any Number?**

I/P:

**import** math

**def** genNatLog():

in\_num **=** eval(input("Enter a Number: "))

print(math**.**log(in\_num))

genNatLog()

O/P:

Enter a Number: 32

3.4657359027997265

**5. Write a Python Program for Cube sum of first n Natural Numbers?**

I/P:

**def** cubeOfNaturalNumbers():

in\_num **=** int(input("Enter the no of Natural Numbers: "))

result **=** pow(((in\_num **\*** (in\_num **+**1))**/**2),2)

print(f'The Cube Sum of First {in\_num} Natural Numbers is {result}')

cubeOfNaturalNumbers()

O/P:

Enter the no of Natural Numbers: 3

The Cube Sum of First 3 Natural Numbers is 36.0