**Experiment No : 6**

**Name : Mohammad Sohail Shaikh A 65**

**Code: Write a Python program to apply decorators to a factorial function.**

# Define a decorator to log function execution

def log\_execution(func):

    def wrapper(n):

        result = func(n)  # Call the original factorial function

        print(f"Factorial of {n} is {result}")

        return result

    return wrapper

# Apply the decorator to the factorial function

@log\_execution

def factorial(n):

    """Calculates factorial of a number recursively."""

    return 1 if n == 0 else n \* factorial(n - 1)

# Get user input

num = int(input("Enter a number: "))

# Call the decorated function

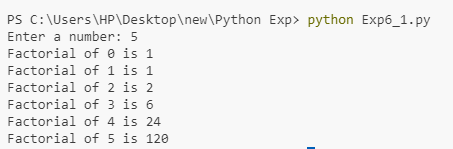
if num < 0:

    print("Factorial is not defined for negative numbers.")

else:

    factorial(num)

**Output:**

****

**Code: Write a Python generator for the Fibonacci series.**

def fibonacci\_recursive(n):

    """

    Generates the Fibonacci series up to n terms using recursion.

    :param n: Number of terms

    :return: List of Fibonacci numbers

    """

    if n <= 0:

        return []

    elif n == 1:

        return [0]

    elif n == 2:

        return [0, 1]

    else:

        series = fibonacci\_recursive(n - 1)  # Recursive call

        series.append(series[-1] + series[-2])  # Add next term

        return series

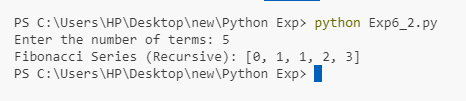
# Get user input

num = int(input("Enter the number of terms: "))

# Print Fibonacci series

print("Fibonacci Series (Recursive):", fibonacci\_recursive(num))

**Output:**

****