

## **Practical No. 1**

### **Title:**

Program to Calculate Fibonacci Numbers and Find Step Count

### **Aim:**

To develop a program that calculates the nth Fibonacci number using the iterative method and computes the number of steps (iterations) required for the computation.

### **Objective:**

To understand and implement an iterative approach to generate Fibonacci numbers and evaluate the computational steps involved.

### **Hardware Requirements:**

- Processor: Intel Core i3 or higher
- RAM: Minimum 2 GB
- Hard Disk: Minimum 100 MB free space
- Keyboard and Mouse
- Monitor

### **Software Requirements:**

- Operating System: Ubuntu
- Python 3.x (or C/C++/Java compiler if using a different language)
- Text Editor or IDE (e.g., VS Code, PyCharm, Notepad++)
- Terminal

### **Theory:**

The Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding ones, starting from 0 and 1.

Mathematically:

- $F(0) = 0$
- $F(1) = 1$
- $F(n) = F(n-1) + F(n-2)$  for  $n > 1$

This sequence appears in many areas of mathematics and computer science. An iterative approach is used here for efficient computation and easy step counting.

### Algorithm:

1. Start
2. Input the value of  $n$
3. Initialize:
  - $a = 0, b = 1$
  - $step = 0$
4. If  $n == 0$ , return 0
5. If  $n == 1$ , return 1
6. Loop from 2 to  $n$ :
  - $c = a + b$
  - $a = b$
  - $b = c$
  - Increment  $step$
7. Return  $b$  and  $step$
8. Stop

### Result:

The program correctly calculates the  $n$ th Fibonacci number using the iterative method and also displays the number of steps required for the computation.

### Conclusion:

The iterative method for calculating Fibonacci numbers is efficient and suitable for large values of  $n$ . It also makes it easy to track the number of steps taken, which is  $(n - 1)$  for  $n \geq 2$ .