

Practical- 1 DAA

Name- Pratham Mahabare
Roll No.- 39

Code-

```
# Iterative Fibonacci with step count
def fibonacci_iterative(n: int):
    step_count = 0 # To count number of steps
    if n == 1:
        return 0, step_count + 1
    elif n == 2:
        return 1, step_count + 1
    else:
        dp = [0] * n
        dp[0], dp[1] = 0, 1
        step_count += 2
        for i in range(2, n):
            dp[i] = dp[i - 1] + dp[i - 2]
            step_count += 1 # Counting each addition as one step
        return dp[n - 1], step_count

# Recursive Fibonacci with memoization and step count
def fibonacci_recursive(n: int, cache=None, steps=None):
    if cache is None:
        cache = {1: 0, 2: 1}
    if steps is None:
        steps = {'count': 0}

    steps['count'] += 1 # Count every call
    if n in cache:
        return cache[n], steps['count']

    cache[n], _ = fibonacci_recursive(n - 1, cache, steps)
    cache[n], _ = fibonacci_recursive(n - 2, cache, steps)
    cache[n] = cache[n - 1] + cache[n - 2]

    return cache[n], steps['count']

# --- Main Program ---
n = int(input("Enter value of n (nth Fibonacci number): "))

fib_iter, steps_iter = fibonacci_iterative(n)
fib_rec, steps_rec = fibonacci_recursive(n)

print(f"\nIterative Fibonacci({n}) = {fib_iter}, Steps = {steps_iter}")
print(f"Recursive Fibonacci({n}) = {fib_rec}, Steps = {steps_rec}")
```

```
print(f"Recursive Fibonacci({n}) = {fib_rec}, Steps = {steps_rec}")
```

Output-

```
> ~/Downloads/LP-3/DAA
pratham_mahabare@fedora:~/Downloads/LP-3/DAA$ python3 1_fibonacci.py
Enter value of n (nth Fibonacci number): 5

Iterative Fibonacci(5) = 3, Steps = 5
Recursive Fibonacci(5) = 3, Steps = 7
pratham_mahabare@fedora:~/Downloads/LP-3/DAA$ 
```