Name: Darshan Bele
Roll No: 14110
BE – A – A1

```
# Fibonacci Calculations with Step Count

# 1. Recursive Fibonacci
step_count_rec = 0

def fib_recursive(n):
    global step_count_rec     step_count_rec += 1
if n <= 1:         return n     return fib_recursive(n - 1)
+ fib_recursive(n - 2)

# 2. Iterative Fibonacci
def fib_iterative(n):
step_count = 0     a, b =
0, 1     for _ in range(n):
a, b = b, a + b
step_count += 1
    return a, step_count

# 3. Dynamic Programming (Memoization)
Fibonacci step_count_memo = 0 memo = {}

def fib_memo(n):
    global    step_count_memo
step_count_memo += 1      if
n in memo:
        return memo[n]     if
n <= 1:
        memo[n] = n     else:
        memo[n] = fib_memo(n - 1) + fib_memo(n - 2)
return memo[n]

# ----- Main ----- if
__name__ == "__main__":
    n = int(input("Enter value of n: "))

    # Recursive
step_count_rec = 0
```

```
    result_rec = fib_recursive(n)     print(f"\nRecursive
Fibonacci of {n} = {result_rec}")     print(f"Recursive
Step Count = {step_count_rec}")

    # Iterative
    result_iter, step_count_iter = fib_iterative(n)
print(f"\nIterative Fibonacci of {n} = {result_iter}")
print(f"Iterative Step Count = {step_count_iter}")

    # Memoization
step_count_memo = 0
memo.clear()
    result_memo  =  fib_memo(n)           print(f"\nMemoization
Fibonacci  of  {n}  =  {result_memo}")      print(f"Memoization
Step Count = {step_count_memo}")
```

OUTPUT:

```
Enter value of n:  10

Recursive Fibonacci of 10 = 55
Recursive Step Count = 177

Iterative Fibonacci of 10 = 55
Iterative Step Count = 10

Memoization Fibonacci of 10 = 55
Memoization Step Count = 19
```