

# Softcopy – Practical-1 (Fibonacci)

## # Code

# Problem Statement: Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.

# Non-recursion

```
def fibonacci(n):
    fib_series = []
    a = 0
    b = 1

    for i in range(n):
        fib_series.append(a)
        a, b = b, a + b

    return fib_series
```

# Recursion

```
def fibonacci_recursive(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    elif n == 2:
        return [0, 1]
    else:
        fib_series = fibonacci_recursive(n - 1) # Get the series up to
n-1
        fib_series.append(fib_series[-1] + fib_series[-2]) # Append the
next Fibonacci number
        return fib_series
```

# Non-recursion

```
n = int(input("Enter total numbers to print in fibonacci series:\t"))
print("Fibonacci Series (non-recusive):\t", fibonacci(n))
```

# Recursion

```
print("Fibonacci Series (recusive):\t\t", fibonacci_recursive(n)
)
```

## # Output

```
$ python3 Code-A1.py
Enter total numbers to print in fibonacci series:      5
Fibonacci Series (non-recusive):                      [0, 1, 1, 2, 3]
Fibonacci Series (recusive):                          [0, 1, 1, 2, 3]
```