

# Experiment No.:-1

**Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.**

## 1) Non - Recursive Program

**Source Code:-**

```
In [1]: nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid if
nterms <= 0:
    print("Please enter a positive integer")
# if there is only one term, return n1 elif
nterms == 1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
# generate fibonacci sequence
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1 + n2      #
        # update values
        n1 = n2            n2 =
        nth                count += 1
```

```
How many terms? 7
Fibonacci sequence:
0
1
1
2
3
5
8
```

## 1. Recursive Program

**Source Code:-**

In [2]:

```
def fibonacci(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    elif n == 2:
        return [0, 1]
    else:
        seq = fibonacci(n - 1)
        seq.append(seq[-1] + seq[-2])
    return seq
nterms = int(input("How many terms? "))

# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    fib_sequence = fibonacci(nterms)
    for num in fib_sequence:
        print(num)
```

How many terms? 7  
Fibonacci sequence:  
0  
1  
1  
2  
3  
5  
8

In [ ]: