Given a number n, return the index value of the Fibonacci sequence, where the sequence is:

- 0 1 1 2 3 5 8 13 21 34 55 89 144 . . .

Fibonacci(5) should return 5 as the 5th index of the Fibonacci sequence.

As on the previous exercise, this operation can be done with both the iterative and recursive functions.

Starting with the interative function:

In [1]:

```
def iterative(index):
    x = 0
    y = 1
    if index == 0:
        return x
    if index == 1:
        return y
    for i in range(2, index + 1):
        a = x + y
        x = y
        y = a
    return a
```

In [2]:

```
print(iterative(5))
```

```
5
```

In [12]:

```
print(iterative(7))
```

```
13
```

In [4]:

```
print(iterative(12))
```

```
144
```

The time complexity for this function is O(n).

Finally, the recursive function:

In [8]:

```
def recursive(index):
    if index == 0:
        return 0
    if index == 1:
        return 1
    return recursive(index-1) + recursive(index-2)
```

When building our recursive function, we started by covering our base cases. Then, we moved on to build the recursive case. Because, on a Fibonacci sequence, each number is the sum of the two preceding numbers, we call the function itself and deduct 1 from the given index, then we call the function itself once more and deduct 2 from the given index. The result of both function calls are added to obtain the index's value and output at the end of the function.

In [9]:

```
print(recursive(5))
```

```
5
```

In [11]:

```
print(recursive(7))
```

```
13
```

In [13]:

```
print(recursive(12))
```

```
144
```

The time complexity of this function is O(2^n).