

# Lab 08 - Using Recursion

---

200pts pts total.

## Problem - Fibonacci Numbers

---

200pts - Due Nov 1, 2021

An example of a recursive definition is:

$$\text{sum}(n) = \begin{cases} 0 & \text{if } n \leq 0 \\ n + \text{sum}(n-1) & \text{if } n > 0 \end{cases}$$

Then we can build a function that matches this.

```
1:
2: def recursive_sum ( n ):
3:     if n <= 0:
4:         return 0
5:     return n + recursive_sum(n-1)
6:
7:
8:
9: # Automated Test
10: if __name__ == "__main__":
11:     n_err = 0
12:     x = recursive_sum ( 5 )
13:     if x != 15:
14:         n_err = n_err + 1
15:         print ( "Error: Test 1: sum not working, expected {} got {}".format ( 15,
16: x = recursive_sum ( 0 )
17: if x != 0:
18:     n_err = n_err + 1
19:     print ( "Error: Test 2: sum conversion not working, expected {} got {}".fc
20:
21: if n_err == 0 :
22:     print ( "PASS" )
23: else:
24:     print ( "FAILED" )
25:
```

Given our definition for a Fibonacci number

$$\begin{aligned} \text{fib}(n) = & \{ 0 : n = 0 \\ & \{ 1 : n = 1 \\ & \{ \text{fib}(n-1) + \text{fib}(n-2) \end{aligned}$$

Implement a recursive function that calculates this in python. Supply an automated test.

Turn in your code.