**Fundamentos Intermedios**

**Parte 1**

1. **Sigma -**Implementa una función sigma(num) que, dado un número, devuelve la suma de todos los enteros positivos (incluyendo el número dado). Ej: sigma(3) = 6 (1+2+3); sigma(5) = 15 (1+2+3+4+5).
2. **Factorial -**Escribe una función factorial(num) que, dado un número, devuelva el producto (multiplicación) de todos los enteros positivos (incluyendo el número dado). Por ejemplo: factorial(3) = 6 (1\*2\*3); factorial(5) = 120 (1\*2\*3\*4\*5).
3. **Fibonacci -**Crea una función para generar números de Fibonacci. En esta famosa secuencia matemática, cada número es la suma de las dos anteriores, partiendo con los valores 0 y 1. Tu función debería aceptar un argumento, un índice en la secuencia (donde 0 corresponde al valor inicial, 4 corresponden al valor cuatro más tarde, etc). Ejemplos: fibonacci(0) = 0 (dado), fibonacci(1) = 1 (dado), fibonacci(2) = 1 (fib(0)+fib(1), o 0+1), fibonacci(3) = 2 (fib(1) + fib(2)3, o 1+1), fibonacci(4) = 3 (1+2), fibonacci(5) = 5 (2+3), fibonacci(6) = 8 (3+5), fibonacci(7) = 13 (5+8). Haz esto primero sin usar recursión. Si no sabes qué es una recursión, no te preocupes puesto que vamos a introducir este concepto en la Parte 2 de esta actividad.
4. **Array: Penúltimo:** Devuelve el penúltimo elemento del array. Dado [42,true,4,”Liam”, 7] devuelve “Liam”. Si el array es muy pequeño, devuelve null.
5. **Array: “N” último:** Devuelve el elemento “N” último. Dado ([5,2,3,6,4,9,7],3], devuelve 4. Si el array es muy pequeño, devuelve null.
6. **Array: Segundo más grande:**Devuelve el segundo elemento más grande de un array. Dado [42,1,4,3.14,7], devuelve 7.  Si el array es muy pequeño, devuelve null.
7. **Doble Problema Par:**Crea una función que cambie un array dado duplicando cada uno de sus elementos en una posición par, y manteniendo el orden original. Convierte [4, "Ulysses", 42, false]   a    [4,4, "Ulysses", 42, 42, false].