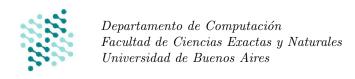
## Algoritmos y Estructuras de Datos I

Primer Cuatrimestre 2018

## Guía Práctica EJERCICIOS DE TALLER



## 1. Introducción a C++

Ejercicio 1. Crear un programa (en cualquier editor de texto) y ejecutarlo como se muestra a continuación.

```
Archivo: labo00.cpp

#include <iostream>

int f(int x){
    return x+1;
}

int main() {
    std::cout << "El resultado es: " << f(10) << std::endl;
    return 0;
}

Para compilar y ejecutar el código en la terminal:
g++ labo00.cpp -o labo00_ejecutable
./labo00_ejecutable</pre>
```

Ejercicio 2. Modificar el programa anterior para que f tome dos parámetros de tipo int y los sume.

**Ejercicio 3.** Modificar el programa anterior para que f tome dos parámetros x e y de tipo int y los sume sólo si x > y, en caso contrario el resultado será el producto.

**Ejercicio 4.** Crear un proyecto nuevo de C++ en **CLion** con el nombre labo00. Escribir el programa del ejercicio anterior y ejecutarlo.

**Ejercicio 5.** Escribir la función que dado  $n \in \mathbb{N}$  devuelve si es primo. Recuerden que un número es primo si los únicos divisores que tiene son 1 y el mismo.

## Iteración vs Recursión

Los siguientes ejercicios deben ser implementados primero en su versión **recursiva**, luego iterativa utilizando **while** y por último iterativa utilizando **for**. Para todos ellos, utilizar el siguiente esqueleto de archivo y modificarlo con los procedimientos que implementen.

**Ejercicio 6.** Escribir la función de Fibonacci que dado un entero n devuelve el n-ésimo número de Fibonacci. Los números de Fibonacci empiezan con  $F_0 = 0$  y  $F_1 = 1$ .  $F_n = F_{n-1} + F_{n-2}$ 

Ejercicio 7. Escribir la función que dado  $n \in \mathbb{N}$  devuelve la suma de todos los números impares menores que n.

**Ejercicio 8.** Escribir la función sumaDivisores que dado  $n \in \mathbb{N}$ , devuelve la suma de todos sus divisores entre [1, n].

• Hint: Recordar que para la versión recursiva es necesario implementar divisoresHasta

**Ejercicio 9.** Escribir una función que dados n, k  $\in \mathbb{N}$  compute el combinatorio:  $\binom{n}{k}$ . Hacerlo usando la igualdad  $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ 

¿Qué pasa si tuvieran que escribir la versión iterativa?

Ejercicio 10. ¿Es mejor programar utilizando algoritmos recursivos ó iterativos? ¿Es mejor usar while o for?