

Taller de Programación C++

Ejercicio 1. *Nuevo proyecto CLION* Crear un proyecto nuevo de C++ en **CLion** con el nombre labo01. En el archivo *labo.cpp* escribir el siguiente programa y guardarlo en el directorio del proyecto labo01. Editar el archivo CMAKELIST.TXT, reemplazando el archivo *main.cpp* por el archivo *labo.cpp*. Ejecutar el proyecto.

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;
}
```

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. Escribir la función que dado $n \in \mathbb{N}$ devuelve una variable booleana con el valor **true** si es primo. Recuerden que un número es primo si los únicos divisores que tiene son 1 y él mismo.

1. Recursión e Iteración

Los siguientes ejercicios deben ser implementados en versión **resursiva**. Luego, generar una nueva función que utilice la versión iterativa con **while** y con **for**.

Ejercicio 5. 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 6. Escribir la función que dado $n \in \mathbb{N}$ devuelve la suma de todos los números impares menores que n .

Ejercicio 7. Escribir la función `sumaDivisores` que dado $n \in \mathbb{N}$, devuelve la suma de todos sus divisores entre $[1, n]$. Para la versión recursiva, es conveniente utilizar una función **divisoresHasta**.

Ejercicio 8. 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?