

# CS1103

## Programación Orientada a Objetos 2

### Unidad 4: Semana 9 - Librería Estándar - Ejercicios

**Estanislao Contreras**  
**Rubén Rivas**

# Objetivos

**Realizar Ejercicios de Contenedores e iteradores de C++.**

# Ejercicio 1

Implementar la función find que retorne el iterador que contenga la primera ocurrencia de x en el rango desde **start** hasta **end** sin incluirlo. si el valor no es ubicado, la función debe retornar el iterador **end**:

```
template <typename Iterator, typename Object>  
Iterator find( Iterator start,  
              Iterator end, const Object & x );
```

## Ejercicio 2

Implementar la función **merge**, que fusione 2 contenedores ordenados en rangos **[first1, last1)** y **[first2, last2)** y generen un rango ordenado que empiece en **d\_first**.

```
template<class InputIt1, class InputIt2, class OutputIt>  
OutputIt merge(InputIt1 first1, InputIt1 last1,  
               InputIt2 first2, InputIt2 last2,  
               OutputIt d_first);
```

## Ejercicio 3

Dado 2 listas (**`std::list`**) ordenadas, L1 y L2, escribir una función que genere una nueva lista  $L1 \cap L2$  usando solo operaciones básicas de listas.

# Ejercicio 4

Dado 2 listas (**`std::list`**) ordenadas, L1 y L2, escribir una función que genere una nueva lista  $L1 \cup L2$  usando solo operaciones básicas de listas.

# Ejercicio 5

Escribir la función **insert\_sorted** que permita agregar un valor en forma ordenada:

```
template <typename T, typename Container>  
void insert_sorted(Container<T> cnt, T value);
```

## Ejercicio 6

Implementar la función **remove** que remueva todos los elementos que satisfaga un criterio especificado por **value** en el rango **[first, last)** y que retorne el último iterador para el nuevo rango.

```
template< class ForwardIt, class T >  
ForwardIt remove(  
    ForwardIt first, ForwardIt last,  
    const T& value );
```



# Ejercicio 7

Implementar la función **copy** que agregue todos los elementos en el rango [**first**, **last**) hacia **d\_first** y retorna el último elemento copiado en el rango de destino.

```
template< class InputIt, class OutputIt >  
OutputIt copy(  
    InputIt first, InputIt last,  
    OutputIt d_first );
```

# Ejercicio 8

Implementar la función **for\_each** que recorra todos los elementos en el rango **[first, last)** y que ejecute una operación a las **dereferencias** de cada uno de los iteradores que hayan sido recorridos.

```
template< class InputIt, class UnaryFunction >  
void for_each(  
    InputIt first, InputIt last,  
    UnaryFunction fun );
```

## Ejercicio 9

Implementar la función **transform** que recorra todos los elementos en el rango **[first, last)** y que ejecute una operación a las **dereferencias** de cada uno de los iteradores que hayan sido recorridos.

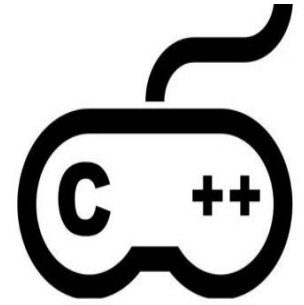
```
template< class InputIt, class OutputIt,  
          class UnaryFunction >  
OutputIt transform(  
    InputIt first, InputIt last,  
    OutputIt d_first,  
    UnaryFunction fun );
```

# Ejercicio 10

El problema de Josefo es el siguiente:  $N$  personas numeradas de 1 a  $N$ , son ubicadas en un círculo, Empezando en la persona 1, una papa caliente es pasada. Luego de  $M$  pases, la persona que mantiene la papa caliente es eliminada, el círculo es cerrado y el juego continúa con las personas que van quedando en el círculo, la última persona que queda es la que gana, Asi si  $M = 0$  y  $N = 5$ , los jugadores son eliminados en orden y el jugador 5 gana, si  $M = 1$  y  $N = 5$  el orden de eliminación es 2, 4, 1, 5.

- Escribir un programa que resuelva el problema de Josefo para valores generales de  $M$  y  $N$ . Tratar de hacer el programa lo más eficiente que sea posible, asegurarse que se dispone de valores.
- ¿Cuál es el tiempo de ejecución del programa?
- Si  $M = 1$ , ¿Cuál es el tiempo de ejecución del programa? ¿Cómo es afectada la velocidad actual para la rutina de borrado de números largos de  $N$  ( $N > 100,000$ )?

# Bibliografía:



- **C++ Standard Library Quick Reference.** Peter Van Weert, Marc Gregoire. Apress, Jun 13, 2016.
- **C++17 STL Cookbook.** Jacek Galowicz. Jun 28, 2017 by Packt Publishing Ltd.
- **Mastering the C++17 STL.** Arthur O'Dwyer. Packt Publishing Ltd, Sep 28, 2017.