

# Algoritmos y Estructuras de datos

## Ejercicios previos al segundo examen parcial

El apuntador ini apunta al primer elemento de la secuencia  
El apuntador fin apunta después del último elemento de la secuencia

- Escribe una función que tome una secuencia de enteros y la ordene. No es necesario implementar un algoritmo eficiente, basta con que funcione. Puede usar cualquier utilidad de la biblioteca de C++ excepto `std::sort` y `std::stable_sort`.

```
void ordena(int* ini, int* fin) {  
    //...  
}
```

- Escribe una función que tome dos alumnos y pueda ser usada como predicado de orden para ordenar alumnos por calificación de mayor a menor; si hay alumnos que tienen la misma calificación, éstos deben ordenarse por número de lista de menor a mayor.

```
struct alumno {  
    int lista, calif;  
};  
  
bool predicado(alumno a, alumno b) {  
    //...  
}
```

- Escribe una función que tome un entero positivo N y que devuelva un arreglo que contenga los enteros del 0 al N-1. El arreglo devuelto debe seguir siendo válido aún después de que la función termine. No se permite usar variables globales ni estáticas.

```
int* genera(int n) {  
    //...  
}
```

- Escribe una función que tome una doble cola de enteros y devuelva otra con los elementos de la primera pero en orden contrario. No se permite modificar la primera doble cola ni usar las utilidades de `algorithm` de la biblioteca de C++, pero sí las de `deque`.

```
std::deque<int> invierte(const std::deque<int>& d) {  
    //...  
}
```