
Intersección de listas ordenadas

T58713_es

Escribe la siguiente función, que calcula la intersección de dos listas de personas en **orden ascendente** de DNI:

```
/**
 * @brief Calcula la intersección de dos listas ordenadas.
 *
 * @param a Lista en orden ascendente de DNI.
 * @param b Lista en orden ascendente de DNI.
 *
 * @return Lista en orden ascendente de DNI con las personas comunes de a y b.
 */
list<Persona> interseccio_ordenada(const list<Persona>& a, const list<Persona>& b)
```

La intersección contiene las personas que aparecen en ambas listas. Dos personas se consideran iguales si tienen el mismo DNI.

Las listas están ordenadas ascendentemente por DNI (comparación lexicográfica de strings).

La declaración de la clase `Persona` es la siguiente:

```
class Persona {
public:
    Persona(string dni, string nom); // Construye una persona
    string dni() const; // devuelve el DNI
    string nom() const; // devuelve el nombre
};
```

Observación

Debes implementar la función `interseccio_ordenada` en el fichero `interseccio.cc`, compilar (está preparado para poder compilar y depurar con VSCode), y finalmente **enviar únicamente el fichero `interseccio.cc`**.

Es necesario que la función sea eficiente y que no visite elementos de forma innecesaria.

Los ficheros públicos (icono del gatito) contienen:

<code>main.cc</code>	el programa principal, con la entrada/salida hecha
<code>persona.hh</code>	la clase <code>Persona</code> , con los métodos mencionados
<code>Makefile</code>	para compilar con <code>make</code> cómodamente
<code>.vscode</code>	carpeta para compilar y depurar con VSCode

Entrada

La función recibe dos listas de personas no modificables, ordenadas ascendentemente por DNI. (De la lectura de las listas, que están separadas por una línea vacía, ya se encarga el programa principal.)

Salida

La función devuelve la lista con las personas comunes de las dos listas, preservando el orden ascendente original. (De la escritura de la lista resultante también se encarga el programa principal.)

Ejemplo de entrada

```
12345678Z Anna
34567890K Clara
45678901X David
56789012T Elena
78901234B Gerard
```

```
23456789D Bernat
34567890K Clara
56789012T Elena
67890123L Ferran
89012345C Helena
```

Ejemplo de salida

```
34567890K Clara
56789012T Elena
```

Información del problema

Autoría: Pau Fernández

Traducción: Pau Fernández

Generación: 2026-02-25T09:47:31.538Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>