

Supon la declaración de la tupla Player:

```
struct Player {
    string first, last; // nombre, apellido
    int points;        // puntos
};
```

que almacena la información de un jugador de la ATP.

Haz una función ordena con la cabecera:

```
void ordena(list<Player>& P, bool by_points);
```

que permite ordenar los jugadores por puntos (cuando `by_points` es cierto) o por apellido (cuando `by_points` es falso).

## Entrada

La acción recibe una lista de jugadores, posiblemente vacía.

## Salida

La acción devuelve la lista ordenada por puntos, si `by_points` es `true`, y ordenada por apellido, si `by_points` es `false`.

## Observación

La solución a este problema puede tener más de una función.

El programa principal que utiliza el jutge es el siguiente:

```
Player p;
list<Player> L;
while (cin >> p.first >> p.last >> p.points) {
    L.push_back(p);
}
ordena(L, true);
mostra(L);
ordena(L, false);
mostra(L);
```

## Ejemplo de entrada 1

```
Roger Federer 8710
Rafael Nadal 10415
Novak Djokovic 13310
David Ferrer 4665
Nicolas Almagro 2070
Andy Murray 7450
```

## Ejemplo de salida 1

```
Novak Djokovic 13310
Rafael Nadal 10415
Roger Federer 8710
Andy Murray 7450
David Ferrer 4665
Nicolas Almagro 2070

Nicolas Almagro 2070
```

Novak Djokovic 13310  
Roger Federer 8710  
David Ferrer 4665

### Ejemplo de entrada 2

X C 2000  
X B 1000  
X A 500

Andy Murray 7450  
Rafael Nadal 10415

### Ejemplo de salida 2

X C 2000  
X B 1000  
X A 500

X A 500  
X B 1000  
X C 2000

### Información del problema

Autor : Pau Fernández  
Traductor : Pau Fernández  
Generación : 2013-09-02 14:08:08

© *Jutge.org*, 2006–2013.  
<http://www.jutge.org>