

---

## Troceos Ordenados

**V12257\_es**

Sea  $v$  un vector de enteros. Un trozo del vector  $v$  es un subvector **máximo** en el que todos los elementos están ordenados. Por ejemplo, el vector  $v = [20, 22, 31, 1, 3, 5, 7, 2, 12, 16]$  tiene 3 trozos. Además, cada trozo tiene un número de orden, que es el orden dentro del vector  $v$ :

orden:	20 22 31 1	1 3 5 7 2	2 12 16 3
--------	---------------	--------------	--------------

Debes hacer una **función** que devuelva los trozos que tiene el vector  $v$ . Esto quiere decir que para cada trozo hay que devolver: el orden dentro del vector, el primer valor del trozo y el último valor del trozo. Además, debes devolverlos (en un vector) según el orden del primer valor del trozo: de menor a mayor. En caso de empate en el primer valor del trozo, irá primero el trozo que tenga el orden más pequeño.

Por ejemplo, para el vector  $v = [20, 22, 31, 1, 3, 5, 7, 2, 12, 16]$  hay que devolver un vector que contenga esto (y en este orden):

orden	1er valor	último valor
2	1	7
3	2	16
1	20	31

Cada fila corresponde a un trozo. En primera posición está el trozo con el primer valor más pequeño: el que empieza con un 1 y termina con un 7. Este trozo se encuentra en segunda posición dentro de  $v$ . Después viene el trozo que empieza con un 2 y termina con un 16. Este trozo se encuentra en tercera posición dentro de  $v$ . Finalmente viene el trozo que empieza con un 20 y termina con un 31. Este trozo se encuentra en primera posición dentro de  $v$ .

Haz una **función** `bocins` con la siguiente declaración:

```
/*
Devuelve un vector con el orden, el primer valor y el último valor
de cada trozo del vector v, ordenados por el primer valor,
de menor a mayor.

En caso de empate, el trozo con el orden más pequeño va primero.
*/
```

  

```
vector<Boci> bocins(const vector<int>& v)
```

Para que tu función compile, deberás usar la tupla `Boci`. **Es necesario añadir** el siguiente código al fichero que enviarás al juez:

```
#ifndef BOCI
#define BOCI
struct Boci
{
    int ordre;
```

```

    int inici;
    int final;
};

#endif

```

Te recomendamos que copies y pegues este código del fichero `main.cc` que te damos.

## Observación

Debes enviar un fichero que contenga **únicamente**:

1. la función que te pedimos.
2. las funciones auxiliares que hayas declarado (si las hay).
3. los `include` necesarios.
4. el código para declarar la tupla `Boci` (que se encuentra en el `main.cc`).

**No debes** poner el `main` en el fichero que enviarás, porque si lo haces, el juez te dará error. No se puede usar la ordenación de C++: `std::sort`. Si quieres ordenar un vector, debes implementarlo tú (puedes usar cualquier algoritmo).

Si lo consideras conveniente, en este ejercicio se pueden usar: el método `push_back()` de la clase `vector`, `min`, `max` o `swap`.

## Entrada

Un vector de enteros. El vector tiene al menos dos elementos.

## Salida

Devuelve un vector con el orden, el primer valor y el último valor de cada trozo del vector `v`, ordenados por el primer valor, de menor a mayor. En caso de empate, el trozo con el orden más pequeño va primero.

### Ejemplo de entrada 1

```

10
20 22 31 1 3 5 7 2 12 16
6
1 2 1 3 1 4
6
1 4 1 3 1 2
15
14 15 13 10 11 12 8 9 4 5 6 7 1 2 3
5

```

1 2 3 4 5	
5	
4 5 1 2 3	

## Ejemplo de salida 1

2 1 7  
3 2 16  
1 20 31

1 1 2  
2 1 3  
3 1 4

1 1 4  
2 1 3  
3 1 2

6 1 3  
5 4 7  
4 8 9  
3 10 12  
2 13 13  
1 14 15

1 1 5

2 1 3  
1 4 5

## Información del problema

Autoría: PRO1

Generación: 2026-01-25T13:05:28.863Z

© *Jutge.org*, 2006–2026.  
<https://jutge.org>