

---

## Bocins Ordenats

V12257\_ca

---

Sigui  $V$  un vector d'enters. Un bocí del vector  $V$  és un subvector **màxim** en què tots els elements estan ordenats. Per exemple, el vector  $V = 20\ 22\ 31\ 1\ 3\ 5\ 7\ 2\ 12\ 16$  té 3 bocins. A més, cada bocí té un número d'ordre, que és l'ordre dins del vector  $V$ :

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

Has de fer una **funció** que torni els bocins que té el vector  $V$ . Això vol dir que per a cada bocí cal tornar: l'ordre dins del vector, el primer valor del bocí, i el darrer valor del bocí. A més, els has de tornar (en un vector) segons l'ordre del primer valor del bocí: de més petit a més gran. En cas d'empat en el primer valor del bocí, anirà primer el bocí que tingui l'ordre més petit.

Per exemple, per al vector  $V = 20\ 22\ 31\ 1\ 3\ 5\ 7\ 2\ 12\ 16$  cal tornar un vector que contingui això (i en aquest ordre):

| ordre | 1er valor | últim valor |
|-------|-----------|-------------|
| 2     | 1         | 7           |
| 3     | 2         | 16          |
| 1     | 20        | 31          |

Cada fila té la informació d'un bocí (que estarà continguda a la tupla `Boci` que definim més endavant). En primera posició hi ha el bocí amb el primer valor més petit: el que comença amb un 1 i acaba amb un 7. Aquest bocí es troba en segona posició dins de  $V$ . Després ve el bocí que comença amb un 2 i acaba amb un 16. Aquest bocí es troba en tercera posició dins de  $V$ . Finalment ve el bocí que comença amb un 20 i acaba amb un 31. Aquest bocí es troba en primera posició dins de  $V$ .

Fes una **funció** `bocins` amb la següent declaració:

```
/*
   Torna un vector amb l'ordre, el primer valor i el darrer valor
   de cada bocí del vector v, ordenats pel primer valor, de petit a gran.
   En cas d'empat, el bocí amb l'ordre més petit va primer.
*/

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

Perquè la teva funció compili, hauràs de fer servir la tupla `Boci`. **Cal afegir** el següent codi al fitxer que enviaràs al jutge:

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

```

    int inici;
    int final;
};
#endif

```

Et recomanem que copiïs i enganxis aquest codi del fitxer `main.cc` que et donem.

## Observació

Has d'enviar un fitxer que contingui **únicament**:

1. la funció que et demanem.
2. les funcions auxiliars que hagi declarat (si n'hi ha).
3. els `include` que calguin.
4. el codi per a declarar la tupla `Boci` (que es troba al `main.cc`).

**No has** de posar el `main` al fitxer que enviaràs, perquè si ho fas, el jutge et donarà error.

No es pot fer servir l'ordenació del C++: `std::sort`. Si vols ordenar un vector, has d'implementar-ho tu (pots fer servir qualsevol algorisme).

Si ho creieu convenient, en aquest exercici es poden fer servir: el mètode `push_back()` de la classe `vector`, `min`, `max` o `swap`.

## Entrada

Un vector d'enters. El vector té almenys dos elements.

## Sortida

Torna un vector amb l'ordre, el primer valor i el darrer valor de cada bocí del vector `v`, ordenats pel primer valor, de petit a gran. En cas d'empat, el bocí amb l'ordre més petit va primer.

### Exemple d'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

## Exemple de sortida 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
```

## Informació del problema

Autoria: PRO1

Generació: 2026-01-25T13:05:24.218Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>