

---

**Bocins Iguals****W75823\_ca**

---

Sigui  $V$  un vector d'enters. Un **Bocí** del vector  $V$  és un subvector **màxim** en què tots els elements són iguals. Per exemple, el vector  $V = 5\ 5\ 5\ 5\ 6\ 4\ 4\ 5\ 5\ 5\ 7\ 7$  té 5 bocins:

5 5 5 5      6      4 4      5 5 5      7 7

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

Per exemple, per al vector  $V = 5\ 5\ 5\ 5\ 6\ 4\ 4\ 5\ 5\ 5\ 7\ 7$  cal tornar un vector que contingui això (i en aquest ordre):

6   4   4  
4   5   6  
7   10   11  
5   7   9  
5   0   3

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 més petit: el que té un 6 i va de la posició 4 a la posició 4 (té mida 1). El següent és el que té un 4 i va de la posició 5 a la posició 6 (té mida 2), etcètera fins a l'últim, que és el que té un 5 i va de la posició 0 a la posició 3 (té mida 4). Fixa't que el segon bocí més petit (4 5 6) i el tercer (7 10 11) tenen la mateixa mida, però (4 5 6) va abans de (7 10 11) perquè  $4 \leq 7$ .

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

```
/*  
    Torna un vector amb el valor, inici i final dels bocins  
    del vector v, ordenats per mida, de menor a més gran.  
    En cas d'empat en la mida, va primer el bocí amb el valor més petit.  
    En cas d'empat en la mida i en el valor,  
    anirà primer el bocí que comenci a la posició més petita.  
*/  
  
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 valor;
    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 hakis 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

El valor, inici i final dels bocins del vector `v`, ordenats per mida, de menor a més gran. En cas d'empat, el valor més petit va primer. En cas d'empat en la mida i en el valor, anirà primer el bocí que comenci a la posició més petita.

### Exemple d'entrada 1

```

12
5 5 5 5 6 4 4 5 5 5 7 7
10
1 2 3 4 5 6 7 8 9 10
5
1 1 1 1 1
4
4 3 2 1

```

### Exemple de sortida 1

```

6 4 4
4 5 6
7 10 11
5 7 9
5 0 3

1 0 0
2 1 1
3 2 2
4 3 3
5 4 4
6 5 5
7 6 6
8 7 7
9 8 8
10 9 9

```

1 0 4

1 3 3

2	2	2
3	1	1
4	0	0

## Informació del problema

Autoria: PRO1

Generació: 2026-01-25T13:20:03.231Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>