
Cases i Pobles**S71653_ca**

Tenim un conjunt de cases situades al llarg d'una carretera, en posicions enteres (que són com la distància des de l'inici de la carretera fins la casa). Un **poble** és una subseqüència màxima de 2 cases o més, consecutives a la carretera, que estan a menys de 3 unitats de distància entre elles. Per aquesta definició, els pobles no es poden solapar entre sí. (A vegades també hi ha cases soles, perquè estan a una distància de 3 o més de les seves cases veïnes.)

Per exemple, considerem les següents cases a la carretera:

Nom	Can_Sola	Can_Verdet	Cal_Quer	Ca_n_Aurelia	Can_Roca
Posició	0	1	4	8	10

En aquest cas, hi ha 2 pobles: Can_Sola i Can_Verdet per una banda; i Ca_n_Aurelia i Can_Roca per l'altra. Cal_Quer no forma part de cap poble.

Has de fer una **funció** `detecta_pobles` amb la següent declaració:

```
struct Casa {
    string nom;
    int pos;
};

struct Poble {
    int inici, fi;
    int num_cases;
};

/**
 * @brief Detecta els pobles de la carretera.
 *
 * @pre   El vector `cases` està ordenat per nom.
 * @post  El resultat està ordenat per posició.
 */
vector<Poble> detecta_pobles(vector<Casa>& cases);
```

El vector `cases` conté les cases, però aquestes venen ordenades per *nom*. El resultat ha de ser un vector de pobles (`vector<Poble>`), amb tots els pobles (usant la definició) en l'ordre en què es troben en la carretera (és a dir, ordenats per posició).

Observació

La icona de nom ".CPP" conté el programa principal per fer proves.

Només has d'enviar un fitxer que contingui la funció requerida, amb els `include` necessaris i les funcions auxiliars que hauràs declarat (si n'hi ha), i **res més**.

No es pot fer servir l'ordenació `sort` de C++. Si vols ordenar un vector, has d'implementar la ordenació tu mateix (pots fer servir qualsevol algorisme).

Per poder utilitzar les estructures a la solució, cal copiar i enganxar el següent codi (que evita que surti un error que diu que la declaració de `Casa` i `Poble` està duplicada):

```

#ifndef TIPUS
#define TIPUS

struct Casa {
    string nom;
    int pos;
};

struct Poble {
    int inici, fi;
    int num_cases;
};

#endif

```

Aquest codi també es troba al fitxer `main.cc` que et donem.

Entrada

L'entrada ja la processa el programa principal que es dona (icona `.CPP`). L'entrada consisteix en diverses seqüències de cases, a on cadascuna comença amb un enter que indica quantes cases hi ha a la carretera, i després segueixen les cases ordenades per nom.

Sortida

La sortida ja la produeix el programa principal que es dona (icona `.CPP`). Per a cada seqüència de cases, es mostren els pobles detectats, cadascun amb la posició inicial, la final, i el nombre de cases.

Exemple d'entrada 1

```

10
Ca_n_Aurelia 8
Cal_Mestre 14
Cal_Quer 7
Can_Pla 13
Can_Roca 20
Can_Sola 0
Can_Torrent 3
Can_Verdet 2
Can_Vila 25
Can_Xic 30

```

Exemple de sortida 1

```

____
0 3 3
7 8 2
13 14 2

```

Informació del problema

Autoria: Pau Fernández

Generació: 2026-01-25T12:47:55.228Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>