
Subasta

X85340_es

En una subasta, los postores van haciendo ofertas por un objeto. El subastador anuncia cada vez que alguien supera la mejor oferta hasta el momento. Los postores también pueden retirarse de la subasta.

Escribe un programa que simule una subasta. La entrada empieza con un entero, el *precio mínimo* de salida. Después viene una secuencia de eventos de dos tipos:

- `oferta <nom> <quantitat>`: el postor `<nom>` hace una oferta de `<quantitat>` euros.
- `retira <nom>`: el postor `<nom>` se retira de la subasta.

Después de cada evento, si la mejor oferta ha cambiado, hay que escribir:

```
millor oferta: <quantitat> (<nom>)
```

Si no ha cambiado, no se escribe nada.

La mejor oferta inicial es el precio mínimo de salida. Todas las ofertas se registran, pero solo se anuncian si superan la mejor oferta.

Hay que ignorar también las situaciones siguientes (no producen ninguna salida ni efecto):

- Una oferta de alguien que ya se ha retirado.
- Una retirada de alguien que ya se había retirado.

Una vez un postor se retira, no puede volver a participar.

Al final de todos los eventos, hay que escribir el ganador:

```
guanyador: <nom> (<quantitat>)
```

Si no queda ningún postor (todos se han retirado o nadie ha ofrecido suficiente), hay que escribir:

```
no hi ha guanyador
```

Entrada

La primera línea contiene un entero, el precio mínimo de salida. Las líneas siguientes contienen eventos de tipo `oferta` o `retira`, como se ha descrito. Los nombres de los postores no contienen espacios.

Salida

Cada vez que la mejor oferta cambia, una línea con el formato indicado. La mejor oferta puede cambiar tanto por una nueva oferta que supere la anterior como por la retirada del postor que tenía la mejor oferta. Al final, una línea con el ganador o indicando que no hay.

Observación

El centro de interés en este problema es la **eficiencia**.

En los archivos públicos (icono del gatito) encontrarás los contenedores de PRO2 (`stack.hh`, `queue.hh`, `heap.hh`, y su dependencia `assert.hh`), un `program.cc` vacío para empezar, un `Makefile` y el directorio `.vscode` con la configuración para compilar y depurar con VS-Code.

Implementa el programa en el archivo `program.cc` y envíalo al Jutge. Puedes usar cualquier contenedor de la STL y los de PRO2 que necesites.

Ejemplo de entrada 1

```
100
oferta anna 150
oferta biel 200
oferta carla 300
```

Ejemplo de entrada 2

```
100
oferta anna 150
oferta biel 200
oferta carla 180
retira biel
```

Ejemplo de entrada 3

```
500
oferta anna 100
oferta biel 200
```

Ejemplo de entrada 4

```
100
oferta anna 110
oferta biel 120
oferta carles 130
oferta daniel 140
retira carles
retira daniel
retira anna
```

Ejemplo de salida 1

```
millor oferta: 150 (anna)
millor oferta: 200 (biel)
millor oferta: 300 (carla)
guanyador: carla (300)
```

Ejemplo de salida 2

```
millor oferta: 150 (anna)
millor oferta: 200 (biel)
millor oferta: 180 (carla)
guanyador: carla (180)
```

Ejemplo de salida 3

```
no hi ha guanyador
```

Ejemplo de salida 4

```
millor oferta: 110 (anna)
millor oferta: 120 (biel)
millor oferta: 130 (carles)
millor oferta: 140 (daniel)
millor oferta: 120 (biel)
guanyador: biel (120)
```

Información del problema

Autoría: Pau Fernández

Traducción: Pau Fernández

Generación: 2026-04-03T21:10:49.427Z

© Jutge.org, 2006–2026.

<https://jutge.org>