

Vectores dinámicos

Tenemos una lista con información sobre el número de objetos disponibles en un cierto almacén. En el almacén pueden aumentar los objetos disponibles, o aumentar el número de unidades disponibles de un objeto ya existente. También existe la posibilidad de disminuir el número de unidades disponibles de un objeto. Si el objeto del que se quieren disminuir las unidades no existe, no se hace nada y si no existen tantas unidades como se piden, su número queda a cero.

El problema consiste en ir leyendo de la entrada las operaciones que se van realizando en el almacén (Aumentar objetos o unidades o Restar unidades) y obtener un listado de los objetos existentes cuando se pida (Listar).

Requisitos de implementación. Se definirá un tipo estructurado con dos campos: el nombre del objeto y su número de unidades. Se declara un vector dinámico, con información sobre su tamaño (memoria reservada) y sobre el número de elementos que tiene (contador). Se reserva memoria para el vector con el operador **new** (tamaño inicial por ejemplo 3). Se procesan las operaciones de la entrada. Cuando el vector se llena, se debe aumentar su tamaño al doble del que está definido. Al aumentar el tamaño del vector y al acabar el programa se debe devolver la memoria dinámica que se ha reservado.

Debe comprobarse que no se pierde memoria. Para ello colocar al comienzo del fichero las directivas al compilador:

```
#ifdef _DEBUG
#define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
#define new DBG_NEW
```

Y como última instrucción del programa principal, antes del return 0; la instrucción:

```
_CrtDumpMemoryLeaks();
```

Después del programa principal debe finalizarse la directiva al compilador con un

```
#endif
```

Entrada

La entrada consta de una serie de operaciones que se realizan en el almacén. Cada operación se describe en una línea. El primer carácter es el código de la operación: A para aumentar el número de objetos o las unidades de un objeto ya existente; R para restar un cierto número de unidades a un objeto ya existente; L para obtener un listado de los objetos del almacén con sus unidades; F para acabar de procesar los datos. En los casos en que se pide aumentar o restar objetos del almacén, a continuación del código separado por un carácter blanco está el identificador del objeto. Este identificador es una cadena alfanumérica sin blancos. A continuación se muestra el número de unidades existentes en el almacén.

Salida

Cada vez que se realice la operación L se deben mostrar todos los objetos del almacén en el orden en que han ido apareciendo en la entrada, seguidos de su número de unidades. Cada objeto se muestra en una línea y al acabar el listado se muestran tres guiones.

Entrada de ejemplo

```
A obj1 3
A obj2 2
R obj1 2
L
A obj1 4
A obj7 1
A obj3 5
R obj7 4
R obj3 2
L
F
```

Salida de ejemplo

```
obj1 1
obj2 2
---
obj1 5
obj2 2
obj7 0
obj3 3
---
```

Autor: Isabel Pita.