

Iteradores inversos

Queremos incluir en el TAD Lista una nueva clase de iteradores que nos permitan recorrer la lista desde el final hacia delante, desde el último elemento al primero. Las operaciones de consulta y avance de estos nuevos iteradores podrían especificarse de la siguiente manera (donde el iterador se representa mediante un cursor justo a la derecha del elemento apuntado por él):

El TAD Lista debe tener operaciones para obtener un iterador al primer elemento en el recorrido inverso (último elemento de la lista) y para obtener un iterador detrás del último elemento del recorrido inverso (justo delante del primer elemento). Sus especificaciones son las siguientes:

list del fichero list_eda.h

Extended la clase ~~ListLinkedDouble~~ con una clase interna `reverse_iterator` y métodos `rbegin()` y `rend()` que permitan recorrer los elementos de una lista en orden inverso, desde el último al primero. Para simplificar vamos a suponer que solamente tenemos una clase de estos iteradores, los no constantes, y que estos iteradores no se utilizan para insertar o eliminar elementos, solamente para recorrer la lista, posiblemente modificándola.

Entrada

La entrada comienza con un número que indica el número de casos de prueba que vienen a continuación. Cada caso se muestra en una línea y consiste en la descripción de una lista: una serie de números que pueden estar comprendidos entre 1 y 10.000, finalizando con un 0 que no se considera parte de la lista.

Salida

Para cada caso de prueba se imprimirá el contenido de la lista de manera inversa, desde el final hasta el principio, utilizando para ello un iterador inverso.

Entrada de ejemplo

```
4
1 2 3 4 0
0
8 17 9 0
3 0
```

Salida de ejemplo

```
4 3 2 1
9 17 8
3
```

Autor

Alberto Verdejo