

Invertir una lista enlazada simple

Invertir una lista consiste en colocar sus elementos en el orden contrario en el que estaban inicialmente, de manera que el primero se convierta en el último, el segundo en el penúltimo, etc. Por ejemplo, el resultado de invertir la lista [1, 2, 3, 4] es [4, 3, 2, 1].

Partimos de la clase `ListLinkedListSingle`, que implementa el TAD de las listas de números enteros mediante listas enlazadas simples y queremos añadir un nuevo método, llamado `reverse()`:

```
class ListLinkedListSingle {
private:
    struct Node {
        int value;
        Node *next;
    };
    Node *head;

public:
    ...
    void reverse();
};
```

La operación `reverse` puede especificarse informalmente del siguiente modo:

```
{ lista = [x0, x1, ..., xn-1] }
lista.reverse();
{ lista = [xn-1, ..., x1, x0] }
```

Se pide:

1. Implementar el método `reverse()`.
2. Indicar su coste con respecto al tamaño de la lista.

Importante: Para la implementación del método no pueden crearse, directa o indirectamente, nuevos nodos mediante `new`. Tampoco se permite copiar valores de un nodo a otro.

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 entrada tras llamar al método `reverse()`. Puedes utilizar el método `display()` de esta clase.

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]
```

Autores

Isabel Pita y Alberto Verdejo