



Antes de pasar a la página siguiente...
¿Has iniciado la grabación del screencast?

Concatenar el inverso de una lista

Primer parcial - Estructuras de Datos - Grupo F
Facultad de Informática - UCM
Tiene un peso del 20 % en la nota final de la asignatura

Este ejercicio debe entregarse en el siguiente problema *DOMjudge*:

- URL: <http://ed.fdi.ucm.es/>
- Concurso: ED-C-EV
- Identificador de problema: C17

Debes entregar:

1. El fichero `.cpp` con el código fuente de tu solución. Utiliza la plantilla que se proporciona con este enunciado. El código fuente debe entregarse en *DOMjudge* antes de **1 hora** desde el comienzo del examen. Se evaluará la última entrega realizada antes de la hora límite.
2. Un breve video explicativo (2-5 minutos) de la solución realizada. Este video se entrega a través de la carpeta de *Google Drive* que se ha compartido contigo. Para ello dispones de **30 minutos** una vez finalizado el plazo de entrega del código fuente.
3. El video con el *screencast* de la realización de la prueba. Este video se entrega a través de la carpeta de *Google Drive* que se ha compartido contigo. Para ello dispones de **1 hora** una vez finalizado el plazo de entrega del código fuente.

Recuerda que **no se permite copiar** en la entrega **código fuente externo**, salvo que provenga de la plantilla proporcionada.

Enlace a las instrucciones:

https://drive.google.com/open?id=1y50P6GtroTeru8-f3TDjn6d8fasIW_9sHNLvMaI5aKs.

En este ejercicio partimos de la clase `ListLinkedList<T>`, que implementa el TAD lista mediante listas doblemente enlazadas circulares con nodo fantasma. Queremos añadir un nuevo método, llamado `concat_reverse()`:

```
template <typename T>
class ListLinkedList {
private:
    struct Node {
        T value;
        Node *next;
        Node *prev;
    };
    Node *head;
    int num_elems;

public:
    ...
    void concat_reverse(ListLinkedList &other);
};
```

El método `concat_reverse()` recibe una lista `other` como parámetro e inserta al final de `this` los elementos de `other` en orden inverso. Tras la llamada, la lista `other` queda vacía. Por ejemplo, si `xs = [1, 2, 3, 4, 5]` y `zs = [60, 70, 80, 90]`, tras la llamada `xs.concat_reverse(zs)` se tiene que `xs = [1, 2, 3, 4, 5, 90, 80, 70, 60]` y `zs = []`.

Importante: Para la implementación del método no pueden crearse, directa o indirectamente, nuevos nodos mediante `new` ni borrar nodos mediante `delete`; han de reutilizarse los nodos de las listas de entrada. Tampoco se permite copiar valores de un nodo a otro. El coste de la operación ha de ser lineal con respecto a la longitud de ambas listas.

Entrada

La entrada comienza con un número que indica el número de casos de prueba que vienen a continuación. Cada caso de prueba consiste en cuatro líneas. La primera línea contiene un número N indicando cuántos elementos tiene la lista `this`. La segunda línea contiene esos N elementos, separados por espacios. La tercera línea contiene un número M indicando cuántos elementos tiene la lista `other`. La cuarta línea contiene esos M elementos, separados por espacios.

Salida

Para cada caso de prueba se imprimirá una línea con el contenido de la lista `this` tras llamar al método `concat_reverse()`. Para imprimir la lista puedes utilizar el método `display()`, o la sobrecarga del operador `<<` que se proporciona para listas.

Entrada de ejemplo

```
2
5
1 2 3 4 5
4
60 70 80 90
0

3
5 7 6
```

Salida de ejemplo

```
[1, 2, 3, 4, 5, 90, 80, 70, 60]
[6, 7, 5]
```