

## Splice en listas

V38718\_es

En este ejercicio implementarás el método `splice` para la clase `List`. Este método recibe un iterador `it` y una lista `L` como parámetros, y tiene el siguiente comportamiento: todos los elementos de `L` se insertan en la lista parámetro implícito justo antes de la posición indicada por `it`, y después de la inserción la lista `L` queda vacía. La declaración es la siguiente:

```
/**
 * @brief introduce una lista en el p.i. justo delante del iterador.
 *
 * @pre   `L` es otra lista. it apunta a un elemento de la lista
 *        p.i. o a la posición end().
 *
 * @post  En la lista (p.i.) se ha introducido la lista L justo
 *        delante de donde apunta el iterador. L queda vacía. it no se mueve.
 */

void splice(iterator& it, List& L);
```

Los ficheros públicos (icono del gatito) contienen:

|                       |  |
|-----------------------|--|
| <code>list.hh</code>  | la clase <code>List&lt;T&gt;</code>                  |
| <code>main.cc</code>  | el programa principal (gestiona la entrada y salida) |
| <code>Makefile</code> | para compilar con <code>make</code> en la terminal   |
| <code>.vscode</code>  | para compilar y depurar con F5                       |

Para entregar solo hace falta **enviar el fichero `list.hh` modificado.**

### Entrada

De la entrada ya se encarga el programa principal. La entrada está formada por distintos casos seguidos. Cada caso comienza con un entero que indica la posición donde insertar, seguido de dos listas (cada lista acaba con `#`).

### Salida

De la salida también se encarga el programa principal. La salida muestra la primera posición del iterador, seguida de la primera lista tras hacer el `splice`, el valor apuntado por el iterador una vez hecho el `splice`, y finalmente la segunda lista (que habrá quedado vacía), toda esta información en 4 líneas separadas y con una línea en blanco entre casos.

### Ejemplo de entrada

|         |            |
|---------|------------|
| 0       | #          |
| #       | 0          |
| #       | #          |
|         | 10 20 30 # |
| 1       | 0          |
| 1 2 3 # | 5 6 7 #    |

```
10 20 #
```

```
1
1 2 3 #
10 20 #
```

```
3
1 2 3 #
10 20 #
```

```
0
10 #
20 #
```

```
2
10 20 #
30 #
```

## Ejemplo de salida

```
<end>
{}
<end>
{}

2
{1, 2, 3}
2
{}

<end>
{10, 20, 30}
<end>
{}

5
{10, 20, 5, 6, 7}
5
{}

2
{1, 10, 20, 2, 3}
2
{}

<end>
{1, 2, 3, 10, 20}
<end>
{}

10
{20, 10}
10
{}

<end>
{10, 20, 30}
<end>
{}

```

## Información del problema

Autoría: M<sup>a</sup> Lluïsa Bonet i Pau Fernández

Generación: 2026-03-25T18:20:58.673Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>