
Mou a l'inici a una llista**V39031_ca**

Implementeu un nou mètode de la classe `List` per a moure el contingut apuntat per un iterador a l'inici de la llista. En cas que l'iterador ja apunti al primer element de la llista, llavors el mètode no farà res. D'entre els fitxers que s'adjunten en aquest exercici, trobareu `list.hh`, a on hi ha una implementació de la classe genèrica `List`.

```
/**
 * @pre  'it' apunta a algún elemento de la lista implícita no vacía.
 *
 * @post 'it' continua apuntando al mismo elemento, el cual ha sido movido
 *        al inicio de la llista. No se ha reservado ni liberado memoria.
 *        En el caso en que el elemento apuntado por 'it' ya fuese el
 *        primero, nada ha cambiado.
 */
void moveToBeginning(iterator &it)
```

(Trobareu la declaració de `moveToBeginning` al final del fitxer `list.hh`, que és bastant llarg.)

No modifiqueu la resta de la implementació de la classe `List`, excepte si, per algun motiu, considereu que necessiteu afegir algun mètode auxiliar a la part privada. D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), que podeu compilar.

Els fitxers públics (icona del gatet) inclouen un `.tar` amb `main.cc`, `list.hh` i un `Makefile`. També s'inclou una còpia dels jocs de prova públics per comoditat. A la carpeta on es descomprimeixin es pot: compilar amb `"make"`; i testar amb `"make test"`.

El `main.cc` ja s'encarrega de llegir l'entrada, processar les comandes i produir la sortida. Per entregar, només cal pujar al Jutge el vostre fitxer `list.hh` modificat.

Entrada

L'entrada del programa és una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre una llista que inicialment està buida i un iterador que està situat inicialment al principi (i final) d'aquesta llista:

```
push_front s      // s és un string
push_back s       // s és un string
pop_front
pop_back
it++
it--
*it
moveToBeginning   // el mètode que heu d'implementar
print_list        // mostra la llista sencera a la sortida
```

Es pot suposar que la seqüència d'entrada serà correcta (és a dir: sense `pop_front` ni `pop_back` sobre llista buida, ni `*it` ni `moveToBeginning` tenint `it` situat a l'end de la

llista; tampoc hi haurà `pop_front` just quan l'iterador estigui apuntant al primer element de la llista, ni hi haurà `pop_back` just quan l'iterador estigui apuntant a l'últim element de la llista; tingueu en compte que l'últim element de la llista no és l'`end` de la llista).

El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe `List`. Només cal que implementeu el mètode abans esmentat.

Sortida

El programa principal ja s'encarrega de produir la sortida: per a cada instrucció `*it`, escriu el contingut apuntat per l'iterador; i per a cada instrucció `print_list` es mostra el contingut de tota la llista.

Exemple d'entrada 1

```
push_back a
it--
*it
print_list
moveToBeginning
print_list
push_back b
it++
*it
print_list
moveToBeginning
print_list
push_front c
it++
*it
print_list
moveToBeginning
print_list
```

Exemple de sortida 1

```
a
a
a
b
a b
b a
a
c b a
a c b
```

Informació del problema

Autoria: PRO2

Generació: 2026-01-27T18:49:30.908Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>