
Mètode de llistes per a moure l'element apuntat per un iterador al final de la llista X86753_ca

Implementeu un nou mètode de la classe `List` per a moure el contingut apuntat per un iterador al final de la llista. En cas que l'iterador ja apunti a l'últim element de la llista, llavors el mètode no farà res.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `list.old.hpp`, a on hi ha una implementació de la classe genèrica `List`. En primer lloc, haureu de fer:

```
cp list.old.hpp list.hpp
```

A continuació, haureu de buscar dins `list.hpp` les següents línies:

```
// Pre: it apunta a algun element de la llista implícita.
// Post: it continua apuntant al mateix element, el qual ha estat mogut al final
//       de la llista. En el cas en que l'element apuntat per it ja era l'últim
//       res ha canviat.
// Descomenteu les següents dues línies i implementeu el mètode:
// void moveToEnd(iterator &it) {
// }
```

Descomenteu les dues línies que s'indiquen i implementeu el mètode. No toqueu la resta de la implementació de la classe, excepte si, per algun motiu, considereu que necessiteu afegir algun mètode auxiliar a la part privada.

Preferiblement, la implementació d'aquest mètode hauria de consistir en modificar punters. De fet, possiblement qualsevol implementació alternativa serà massa lenta o produirà error d'execució.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `program.cpp` (programa principal) i `Makefile` per a compilar. Per a pujar la vostra solució, heu de crear el fitxer `solution.tar` així:

```
tar cf solution.tar list.hpp
```

Entrada

La entrada del programa és una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre una llista que se suposa inicialment buida i un iterador que se suposa 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
moveToEnd
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop_front` ni `pop_back` sobre llista buida, ni `*it` ni `moveToEnd` tenint `it` situat al 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 el `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 els mètodes abans esmentats.

Sortida

Per a cada instrucció `*it`, s'escriurà el contingut apuntat per l'iterador. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

Exemple d'entrada 1

```
push_front a
it--
*it
moveToEnd
*it
push_back b
it++
*it
moveToEnd
*it
it--
*it
moveToEnd
*it
it--
*it
push_front c
*it
it--
*it
moveToEnd
*it
it--
*it
moveToEnd
*it
it--
*it
moveToEnd
*it
```

Exemple de sortida 1

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

Exemple d'entrada 2

```
push_back rb
push_back b
push_back ar
push_back k
pop_front
pop_front
push_back dq
it--
it--
```

```
push_front jm
moveToEnd
it--
moveToEnd
pop_front
push_back yb
it++
it--
*it
push_front a
pop_front
```

```
pop_back
moveToEnd
it--
*it
pop_back
push_front g
push_front pk
it--
moveToEnd
it--
it--
it++
moveToEnd
*it
it--
moveToEnd
pop_front
moveToEnd
moveToEnd
pop_front
*it
```

Exemple de sortida 2

```
dq
k
k
ar
```

Informació del problema

Autor : PRO1

Generació : 2023-05-07 09:08:42

© *Jutge.org*, 2006–2023.

<https://jutge.org>