
Modificar operadors ++ i – dels iteradors de la classe List per a que siguin circulars X90450_ca

Modifiqueu la classe List per a fer que els iteradors es comportin de manera circular, és a dir, que si un iterador `it` està situat al end de la llista i fem `++it` o `it++`, llavors es posicionarà al begin de la llista, i si un iterador `it` és al begin de la llista i fem `--it` o `it--`, llavors es posicionarà al end de la llista.

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` els mètodes d'increment i decrement d'iteradors i modificar-los convenientment. 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. 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 x (x és un string)
push_back x (x és un string)
pop_front
pop_back
it++
++it
it--
--it
*it
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop_front` ni `pop_back` sobre llista buida, ni `*it` 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 el mètode abans esmentat.

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

```
it--
--it
it++
++it
push_front a
it++
*it
++it
++it
*it
it++
it++
*it
--it
--it
*it
it--
it--
*it
++it
pop_back
push_back b
--it
*it
push_back c
*it
--it
it--
*it
--it
pop_back
*it
push_back d
*it
--it
push_back e
it--
*it
push_back f
++it
*it
push_front g
it++
it++
*it
++it
*it
it++
*it
```

Exemple d'entrada 2

```
it--
push_front d
pop_front
++it
it++
it++
--it
```

Exemple de sortida 1

```
a
a
a
a
a
b
b
c
b
b
e
f
g
b
d
```

```
push_back a
--it
*it
++it
pop_front
it++
push_front b
pop_front
push_back d
```

```
pop_back
push_front b
pop_back
--it
push_front a
push_back c
it++
*it
pop_back
it++
push_front b
++it
*it
pop_back
++it
pop_back
push_back c
pop_back
++it
it++
it++
push_back c
push_back d
pop_front
it++
*it
```

Exemple de sortida 2

```
a
a
b
d
```

Informació del problema

Autor : PRO1

Generació : 2023-05-07 09:09:55

© *Jutge.org*, 2006–2023.

<https://jutge.org>