Afegir mètodes += i -= als iteradors de la classe llista X63009_ca

Modifiqueu la classe List per tal d'afegir els operadors += i −= a la subclasse iterador. Aquests mètodes reben un enter, i desplacen l'iterador cap endavant i cap enrere, respectivament, tantes posicions com indica aquest enter. Noteu que, si l'enter és negatiu, += mourà l'iterador cap enrere, i −= mourà l'iterador cap endavant. S'assumeix que els paràmetres no fan que l'iterador se surti del rang 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` les següents línies:

```
// Pre: x és un enter tal que, si desplacem l'iterador implícit x unitats cap e
//      (que implicarà moure'l -x unitats cap enrere si x és negatiu), no se su
// Post: l'iterador implícit s'ha mogut x unitats cap endavant (cosa que haurà
//       moure'l -x unitats cap enrere en el cas que x sigui negatiu, o no mour
// Descomenteu les següents dues linies i implementeu el mètode:
// void operator+=(int x) {
// }

// Pre: x és un enter tal que, si desplacem l'iterador implícit x unitats cap e
//      (que implicarà moure'l -x unitats cap endavant si x és negatiu), no se
// Post: l'iterador implícit s'ha mogut x unitats cap enrere (cosa que haurà im
//       moure'l -x unitats cap endavant en el cas que x sigui negatiu, o no mo
// Descomenteu les següents dues linies i implementeu el mètode:
// void operator-=(int x) {
// }
```

Descomenteu les quatre linies que s'indiquen i implementeu els mètodes. 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 s (s és un string)
push_back s (s és un string)
pop_front
```

```
pop_back
it+= x (x és un enter)
it-= x (x és un enter)
*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). A més a més, els paràmetres x són tals que l'iterador no se surt de rang.
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

```
it+= 0
it-= 0
push_front a
it-= 1
*it
it+= 1
it-= 1
*it
it-= -1
it+= -1
*it
push_back b
it+= 1
*it
it+= -1
*it
it-= -1
*it
it-= 1
*it
push_front c
it-= 1
*it
it+= 3
it-= 1
*it
it+= -2
*it
push_front d
push_back e
it-= -3
*it
it+= -4
*it
```

Exemple de sortida 1

```
a
a
a
b
a
b
a
c
b
c
e
d
```

Exemple d'entrada 2

```
push_front e
it-= 1
it-= 0
push_back g
it-= -2
it+= -2
*it
push_front s
pop_front
push_front di
pop_back
pop_front
push_front il
it-= -1
it-= 1
push_front tq
it+= -2
pop_back
it+= 1
pop_front
it+= 1
it+= -1
it-= -1
it-= 0
it+= -1
it+= 0
*it
push_front ly
push_back j
*it
pop_back
it+= -1
pop_back
push_back pd
pop_back
it+= 0
push_front j
it-= 1
push_back p
it-= -1
push_front k
pop_front
pop_front
push_front f
it+= 1
it+= -1
*it
pop_front
push_back ft
*it
push_back w
it-= -1
*it
*it
*it
pop_front
pop_back
*it
it+= 1
pop_front
*it
```

Exemple de sortida 2

```
e
il
il
ly
```

```
ly
p
p
p
p
ft
```

Exemple d'entrada 3

```
it-= 0
push_back os
pop_front
it-= 0
push_back pe
it+= 0
push_front is
pop_back
push_back di
push_back v
pop_front
it-= 1
push_front vu
push_front tq
push_back v
*it
it-= 1
pop_front
pop_back
*it
it+= -1
push_back yr
pop_back
*it
it+= 1
*it
it+= 2
push_back yf
push_back sj
push_front h
push_front pd
pop_back
push_back k
it+= -2
push_back cg
push_front pb
pop_front
push_front q
pop_front
it-= 5
*it
```

Exemple de sortida 3

```
v
di
vu
di
pd
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:19:30.704Z