
Mètode per a moure el primer element d'una cua cap a la última posició X10999_ca

Implementeu un nou mètode de la classe `Queue` que mogui el seu element del front cap a la última posició. En cas que la cua no tingui cap element, la funció no farà res.

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

```
cp queue.old.hpp queue.hpp
```

A continuació, haureu de buscar dins `queue.hpp` la part:

```
// Pre:
// Post: L'element que estava en primera posició de la cua implícita
//       ha estat mogut cap a la última posició.
//       En el cas en que la cua no tingués cap element, res ha canviat.
// Descomenteu les següents dues línies i implementeu la funció:
// void moveFrontToLast() {
// }
```

Haureu de descomentar les dues línies que s'indiquen i implementar aquest 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.

La idea és que la vostra implementació treballi amb punters. De fet, molt possiblement, una implementació basada en usar `front`, `pop` i `push` serà massa lenta i no superarà els jocs de proves privats.

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 queue.hpp
```

Entrada

La entrada del programa és una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre una cua d'strings que se suposa inicialment buida:

```
push x (x és string)
pop
front
size
moveToLast
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop` ni `front` sobre cua buida).

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

Sortida

Per a cada instrucció `front`, s'escriurà el front actual de la cua. Per a cada instrucció `size`, s'escriurà la mida de la cua. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

Exemple d'entrada 1

```
size
moveFrontToLast
size
push a
front
size
push b
front
size
push c
front
size
moveFrontToLast
front
size
push d
front
size
pop
front
size
moveFrontToLast
front
size
pop
front
size
moveFrontToLast
front
size
moveFrontToLast
front
size
pop
size
moveFrontToLast
size
```

Exemple d'entrada 2

```
push l
front
size
push bm
front
size
pop
front
size
```

Exemple de sortida 1

```
0
0
a
1
a
2
a
3
b
3
b
4
c
3
a
3
d
2
c
1
c
1
c
1
0
0
```

```
moveFrontToLast
front
size
moveFrontToLast
front
size
push ar
front
size
push w
front
```

size
push y
front
size
push d
front
size
moveFrontToLast
front
size
push c
front
size
push rj
front
size
push w
front
size
moveFrontToLast
front
size
push sj
front
size
push ld
front
size
push f
front
size
push r
front
size
push yn
front
size
pop
front
size
pop
front
size
moveFrontToLast
front
size
push g
front
size
moveFrontToLast
front
size
moveFrontToLast
front
size
moveFrontToLast
front
size
pop
front
size

push r
front
size
push ln
front
size
pop
front
size
push p
front
size
push wk
front
size
push p
front
size
pop
front
size
pop
front
size

Exemple de sortida 2

1
1
1
2
bm
1
bm
1
bm
1
bm
2
bm
3
bm
4
bm
5
ar
5
ar
6
ar
7
ar
8
w
8
w
9
w
10
w
11

w
12
w
13
y
12
d
11
bm
11
bm
12
c
12
rj
12
w
12
ar
11
ar
12
ar
13
sj
12
sj
13
sj
14
sj
15
ld
14
f
13

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T20:36:01.123Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>