
Mètode per a moure un element d'una cua cap a la primera posició X53930_ca

Implementeu un nou mètode de la classe `Queue` que mou l'element de la posició `x` a la primera posició de la cua sense alterar la mida. 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: x està entre 0 i la mida de la cua implícita menys 1.
// Post: Mou l'element de la posició x a la primera posició de la cua sense alt
//       Si la cua només té un element o no en té cap la funció no fa res.
// Descomenteu les següents dues línies i implementeu la funció:
// void moveToFront(int x) {
// }
```

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 s (s és string)
pop
size
moveToFront
list
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop` ni `list` 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 `list`, s'escriurà 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

```
push g
pop
push p
pop
size
push z
push s
size
push o
size
moveToFront 1
list
pop
moveToFront 1
list
push q
size
pop
pop
pop
size
size
push z
moveToFront 0
list
push f
pop
size
push u
size
push d
moveToFront 2
list
size
push o
pop
moveToFront 1
list
moveToFront 2
list
size
push a
size
pop
size
moveToFront 2
list
```

Exemple d'entrada 2

```
size
push f
```

Exemple de sortida 1

```
0
2
3
s, z, o
o, z
3
0
0
z
1
2
d, f, u
3
u, f, o
o, u, f
3
4
3
a, u, f
```

```
size
push m
moveToFront 1
list
```

```
push y
moveToFront 0
list
push p
moveToFront 2
list
pop
push f
size
push y
push m
size
moveToFront 3
list
push i
moveToFront 4
list
pop
push m
size
push a
pop
size
size
pop
pop
push o
moveToFront 0
list
moveToFront 2
list
moveToFront 3
list
pop
pop
pop
pop
size
push e
pop
size
moveToFront 1
list
push c
size
push b
size
size
moveToFront 2
list
moveToFront 2
list
size
moveToFront 2
list
push u
push c
moveToFront 2
list
size
size
```

```
pop
size
pop
push p
moveToFront 0
list
size
pop
size
pop
push d
moveToFront 1
list
moveToFront 1
list
pop
push n
push g
size
```

Exemple de sortida 2

```
0
1
m, f
m, f, y
y, m, f, p
4
6
f, m, f, p, y, m
y, f, m, f, p, m, i
7
7
7
p, m, i, m, a, o
i, p, m, m, a, o
m, i, p, m, a, o
2
2
```

```
e, o
3
4
4
c, e, o, b
o, c, e, b
4
e, o, c, b
c, e, o, b, u, c
6
6
5
o, b, u, c, p
5
4
c, u, p, d
u, c, p, d
5
```

Informació del problema

Autoria: María Elizo i Xavi Vila

Generació: 2026-01-25T21:16:10.359Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>