

---

## Mètode de la classe cua per a revessar-la

X53561\_ca

---

Implementeu un nou mètode de la classe `Queue` que revessi els seus propis elements. És a dir, el que estava al principi ara apareixerà al final, el que era el segon des del principi ara apareixerà com a segon des del final, i així successivament.

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: Sigui [e1,e2...,en] el contingut inicial de la cua des del principi fins a
// Post: El contingut final de la cua és [en,...,e2,e1], és a dir, la cua ha estat
// Descomenteu les següents dues línies i implementeu la funció:
// void reverse() {
// }
```

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.

Preferiblement, hauríeu d'aconseguir implementar `reverse` a base d'intercanviar els punters de l'objecte. De fet, una implementació a base d'usar `push` i `pop` us permetrà passar els jocs de proves públics (i així obtenir una part de la nota), però no els 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 que se suposa inicialment buida:

```
push x (x és un string)
pop
front
size
reverse
```

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
reverse
size
push a
front
reverse
front
size
push b
front
size
reverse
front
size
pop
front
push c
front
push d
front
reverse
front
push e
front
size
reverse
front
size
pop
front
pop
front
pop
front
pop
front
pop
size
```

### Exemple d'entrada 2

```
push ab
push b
push a
push a
size
pop
push aa
pop
push bc
reverse
size
front
push ca
push aa
push c
pop
push b
push c
push cb
size
```

### Exemple de sortida 1

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

```
push b
reverse
push ba
size
pop
push ac
reverse
front
pop
pop
reverse
reverse
push cb
push cb
size
push ba
push d
front
push a
push b
reverse
push d
```

```
reverse
reverse
pop
push c
size
push a
pop
front
push c
push ca
push cc
size
size
push db
push b
push cc
push ca
push d
front
```

## Exemple de sortida 2

```
4
4
bc
9
11
ac
11
aa
16
d
19
19
d
```

### Informació del problema

Autor : PRO1

Generació : 2022-11-03 13:51:36

© *Jutge.org*, 2006–2022.

<https://jutge.org>