
Mètode de Queue per a multiplicar els elements de la cua per un paràmetre X50664_ca

Implementeu un nou mètode de la classe `Queue` que rebí un element com a paràmetre i modifiqui tots els elements de la cua a base de multiplicar-los per aquest paràmetre. Òbviament, aquest mètode només serà útil quan instanciem la classe `Queue` amb un tipus per al qual estigui definida la operació producte.

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: Tots els elements de la cua implícita han estat multiplicats per 'value'  
// Descomenteu les següents dues línies i implementeu la funció:  
// void operator*=(T value) {  
// }
```

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.

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 de doubles que se suposa inicialment buida:

```
push x (x és un double)  
pop  
front  
size  
*= x (x és un double)
```

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

```
push 2
push 1.5
push 1.2
size
front
*= 2
front
pop
front
push -5
push 0
*= 1.5
front
pop
front
size
pop
front
pop
front
pop
size
```

Exemple d'entrada 2

```
push 8.6
front
size
*= 3.6
*= 0.6
push 19.872
front
size
*= 0.2
push 1.1664
front
size
push 2.5488
front
size
push 1.1232
front
size
pop
front
size
push 3.1104
front
size
pop
front
size
push 2.9376
front
size
push 1.2528
front
size
push 1.296
```

Exemple de sortida 1

```
3
2
4
3
4.5
3.6
3
-7.5
0
0
```

```
front
size
push 0.9936
front
size
push 1.512
front
size
*= 8.3
push 20.7965
front
size
*= 0.8
*= 0.7
push 8.43333
front
size
*= 0.4
*= 4
pop
front
size
*= 0.9
pop
front
size
push 20.24
front
size
*= 0.7
push 16.192
front
size
pop
front
size
```

```
*= 0.3
push 5.82912
front
size
*= 2.6
*= 0.5
push 2.8417
front
size
```

Exemple de sortida 2

```
8.6
1
18.576
2
3.7152
3
3.7152
4
3.7152
5
3.9744
4
3.9744
5
1.1664
4
1.1664
5
1.1664
6
1.1664
7
1.1664
8
1.1664
9
9.68112
10
5.42143
11
18.9549
10
7.51771
9
7.51771
10
5.2624
11
14.5728
10
4.37184
11
5.68339
12
```

Informació del problema

Autor : PRO1

Generació : 2022-05-05 22:35:11

© *Jutge.org*, 2006–2022.

<https://jutge.org>