

---

## Nou mètode de la classe Queue per a accedir indexadament als seus elements X67645\_ca

---

Implementeu un nou mètode de la classe Queue que permeti accedir indexadament als seus elements amb l'operador []. Per exemple, amb la declaració `Queue<int> q`, la instrucció `q[0]` ha de retornar el primer element de `q` (és a dir, el mateix que retorna `q.front()`), i `q[1]` ha de retornar el segon element de `q`, i `q[2]` ha de retornar el tercer element de `q`, 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: i està entre 0 i la mida de la cua implícita menys 1.
// Post: Retorna l'i-èsim valor de la cua implícita (indexat començant des de
// Descomenteu les següents dues línies i implementeu el mètode:
// T operator[](int i) const {
// }
```

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 strings que se suposa inicialment buida:

```
push s (s és un string)
pop
front
size
[i] (i és un enter)
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop` ni `front` sobre cua buida, ni `[i]` amb un `i` fora 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 `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. Per a cada instrucció `[i]`, s'escriurà el contingut de la posició `i` de la cua. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

### Exemple d'entrada 1

```
size
push a
size
push b
[0]
[1]
front
pop
size
[0]
push abc
push bb
[0]
[1]
[2]
size
pop
front
[0]
[1]
size
```

### Exemple de sortida 1

```
0
1
a
b
a
1
b
b
abc
bb
3
abc
abc
bb
2
```

### Exemple d'entrada 2

```
push bb
push aa
[1]
size
front
size
front
push aa
push ba
[1]
size
size
size
front
front
front
push a
pop
push b
push ab
[3]
push baa
push b
push aa
[5]
size
push bb
```

```
size
[6]
push bb
push b
pop
[0]
[1]
size
```

## Exemple de sortida 2

aa  
2  
bb  
2  
bb  
aa  
4  
4  
4

bb  
bb  
bb  
a  
ab  
9  
10  
baa  
aa  
ba  
11

## Informació del problema

Autor : PRO1

Generació : 2022-06-29 11:30:49

© *Jutge.org*, 2006–2022.

<https://jutge.org>