

---

**Afegir el mètode `push_front` a la classe `Queue`****X94369\_ca**

---

Implementeu un nou mètode de la classe `Queue`, anomenat `push_front`, que permeti afegir un element al principi de la cua, enlloc de al final com passa amb el mètode `push` ja existent. En cas de dubte, mireu el següent programa i el seu comportament descrit en els seus comentaris.

```
Queue<int> q;                // q:
q.push(1);                  // q: 1
q.push(2);                  // q: 1, 2
q.push(3);                  // q: 1, 2, 3
cout << q.front() << endl; // escriu 1
q.push_front(4);            // q: 4, 1, 2, 3
cout << q.front() << endl; // escriu 4
q.pop();                    // q: 1, 2, 3
cout << q.front() << endl; // escriu 1
```

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `queue.hh`, a on hi ha una implementació de la classe genèrica `Queue`. Busqueu dins `queue.hh` les següents línies i implementeu el mètode:

```
// Pre:
// Post: value ha estat afegit al principi de la cua representada pel paràmet
//void push_front(T value) {
//}
```

D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), i el podeu compilar directament, doncs inclou `queue.hh`. Només cal que pugueu `queue.hh` al jutge.

**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
push_front x (x és string)
```

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.

```
size
size
push_front a
front
size
push_front b
front
size
pop
front
size
pop
size
push c
front
size
push d
front
size
push_front e
front
size
front
size
push f
front
size
pop
front
size
front
size
pop
size
pop
front
size
front
size
front
size
pop
size
size
size
```

```
push rb
front
size
push b
front
size
push ar
front
size
push k
front
```

0  
0  
a  
1  
b  
2  
a  
1  
0  
c  
1  
c  
2  
e  
3  
e  
3  
e  
4  
c  
3  
c  
3  
d  
2  
f  
1  
f  
1  
f  
1  
0  
0

```
size
pop
front
size
pop
front
size
push dq
front
size
push xr
front
size
```

push w  
front  
size  
push\_front sj  
front  
size  
push\_front db  
front  
size  
pop  
front  
size  
push r  
front  
size  
push\_front n  
front  
size  
pop  
front  
size  
pop  
front  
size  
push g  
front  
size  
push\_front kl  
front  
size  
pop  
front  
size  
push ln  
front  
size  
pop  
front  
size  
push qf  
front  
size  
push op  
front  
size  
pop  
front  
size  
pop  
front  
size  
push h  
front  
size  
push ku  
front  
size  
pop  
front  
size  
pop

front  
size  
push m  
front  
size  
push\_front uq  
front  
size  
push\_front ji  
front  
size  
push m  
front  
size  
push t  
front  
size  
push\_front x  
front  
size

## Exemple de sortida 2

rb	k1
1	8
rb	ar
2	7
rb	ar
3	8
rb	k
4	7
b	k
3	8
ar	k
2	9
ar	dq
3	8
ar	xr
4	7
ar	xr
5	8
sj	xr
6	9
db	w
7	8
sj	r
6	7
sj	r
7	8
n	uq
8	9
sj	ji
7	10
ar	ji
6	11
ar	ji
7	12
	x
	13

## Observació

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, on cada operació té cost **CONSTANT**, i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

## Informació del problema

Autoria: PRO2

Generació: 2026-01-27T18:57:29.544Z

© Jutge.org, 2006–2026.

<https://jutge.org>