
Mètode per a sumar el contingut que ve d'una altra cua X98788_en

Implementeu un nou mètode de la classe `Queue` que rebi una altra cua del mateix tipus com a paràmetre i modifiqui tots els elements de la cua implícita a base de sumar els elements corresponents de l'altra cua. Òbviament, aquest mètode només serà útil quan instanciem la classe `Queue` amb un tipus per al qual estigui definida la operació suma.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `queue.hh`, a on hi ha una implementació de la classe genèrica `Queue`. Haureu de buscar dins `queue.hh` la part:

```
// Pre:
// Post: Al primer element de la cua implícita se li ha sumat
//       el primer element de q,
//       al segon element de la cua implícita se li ha sumat
//       el segon element de q, i així successivament.
//       Si la cua implícita era més llarga que q,
//       llavors aquells elements extrems no han canviat.
//       Si q era més llarga que la cua implícita, llavors aquells elements
//       extrems de q han estat afegits al final de la cua implícita
//       en el mateix ordre.
// Descomenteu les següents dues línies i implementeu la funció:
// void operator+=(const Queue<T> &q) {
// }
```

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 `main.cc` (programa principal), i el podeu compilar directament, doncs inclou `queue.hh`. Només cal que pugueu `queue.hh` al jutge.

Input

La entrada del programa és una seqüència d'instruccions del següent tipus. En totes elles, 'id' i 'id2' són strings que identifiquen cues sobre les quals s'apliquen el mètodes. S'assumeix que totes les cues usades estaran buides inicialment:

```
id.push x (x és un enter)
id.pop
<< id
id += id2
```

Se suposa que la seqüència d'entrada serà correcta (sense pop 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.

Output

Per a cada instrucció `<< id`, s'escriurà el contingut actual de la cua identificada per `id`. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

Sample input 1

```
q0 += q0
<< q0
q0.push 1
<< q0
q0 += q0
<< q0
q0.push 3
<< q1
q0 += q1
<< q0
<< q1
q1 += q0
<< q0
<< q1
q2 += q1
<< q2
q1.pop
<< q1
q2 += q1
<< q2
q2.pop
<< q2
q2.pop
<< q2
<< q1
<< q0
```

Sample output 1

```
0
1 1
1 2
0
2 2 3
0
2 2 3
2 2 3
2 2 3
1 3
2 5 3
1 3
0
1 3
2 2 3
```

Sample input 2

```
q0 += q0
<< q0
q1.push 0
<< q1
q1.push 9
<< q1
q1.push -7
<< q1
q0.push 4
<< q0
q2.push -5
<< q2
q1.push -5
<< q1
q0 += q0
<< q0
q3.push 6
<< q3
q1 += q1
<< q1
q4 += q4
<< q4
q4.push 7
<< q4
q1.pop
```

```
<< q1
q5.push -9
<< q5
q6 += q2
<< q6
q5 += q3
<< q5
q4 += q3
<< q4
q3 += q2
<< q3
q2.pop
<< q2
q7.push -7
<< q7
q2.push -10
<< q2
q8.push 2
<< q8
q8.push -9
<< q8
q0.push 8
<< q0
q2.push 2
<< q2
q9 += q2
```

```
<< q9
q2 += q6
<< q2
q6 += q1
<< q6
q10.push 9
<< q10
q11 += q3
<< q11
q12 += q1
<< q12
q13.push 8
<< q13
q5.push 4
<< q5
q1.pop
<< q1
q3 += q4
<< q3
```

Sample output 2

```
0
1 0
2 0 9
3 0 9 -7
1 4
1 -5
4 0 9 -7 -5
1 8
1 6
4 0 18 -14 -10
0
1 7
3 18 -14 -10
1 -9
1 -5
1 -3
1 13
1 1
0
1 -7
1 -10
1 2
2 2 -9
2 8 8
2 -10 2
2 -10 2
2 -15 2
3 13 -14 -10
1 9
1 1
3 18 -14 -10
1 8
2 -3 4
2 -14 -10
1 14
```

Problem information

Author: PRO2

Generation: 2026-01-27T18:58:16.999Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>