
Mètode per a sumar el contingut que ve d'una altra pila X80037_ca

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

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

```
// Pre:
// Post: A l'element del top de la pila implícita se li ha sumat
//       l'element del top de s,
//       al segon element des del top de la pila implícita se li ha sumat
//       el segon element des del top de s, i així successivament.
//       Si la pila implícita era més llarga que s,
//       llavors aquells elements extrems no han canviat.
//       Si s era més llarga que la pila implícita, llavors aquells elements
//       extrems de s han estat ignorats.
// Descomenteu les següents dues línies i implementeu la funció:
// void operator+=(const Stack<T> &s) {
// }
```

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 `stack.hh`. Només cal que pugueu `stack.hh` al jutge.

Entrada

La entrada del programa és una seqüència d'instruccions del següent tipus. En totes elles, 'id' i 'id2' són strings que identifiquen piles sobre les quals s'apliquen el mètodes. S'assumeix que totes les piles 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 pila buida).

El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe pila. Només cal que implementeu el mètode abans esmentat.

Sortida

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

Exemple d'entrada 1

```
s0 += s0
<< s0
s0.push 1
<< s0
s0 += s0
<< s0
s0.push 3
<< s1
s0 += s1
<< s0
<< s1
s1 += s0
<< s0
<< s1
s1.push 3
s1.push 2
s2.push 1
s2.push 2
s2.push 3
<< s0
<< s1
<< s2
s2 += s1
s1 += s2
s0 += s1
s2 += s0
<< s0
<< s1
<< s2
s1.pop
<< s1
s2 += s1
<< s2
s2.pop
<< s2
s2.pop
<< s2
```

Exemple d'entrada 2

```
s0 += s0
<< s0
s1.push 0
<< s1
s1.push 9
<< s1
s1.push -7
<< s1
s0.push 4
<< s0
s2.push -5
<< s2
s1.push -5
<< s1
s0 += s0
<< s0
s3.push 6
<< s3
s1 += s1
```

Exemple de sortida 1

```
0
1 1
1 2
0
2 2 3
0
2 2 3
0
2 2 3
2 3 2
3 1 2 3
2 10 10
2 8 7
3 1 15 15
1 8
3 1 15 23
2 1 15
1 1
```

```
<< s1
s4 += s4
<< s4
s4.push 7
<< s4
s1.pop
<< s1
s5.push -9
<< s5
s6 += s2
<< s6
s5 += s3
<< s5
s4 += s3
<< s4
s3 += s2
<< s3
s2.pop
<< s2
s7.push -7
<< s7
```

```

s2.push -10
<< s2
s8.push 2
<< s8
s8.push -9
<< s8
s0.push 8
<< s0
s2.push 2
<< s2
s9 += s2
<< s9
s2 += s6
<< s2
s6 += s1
<< s6
s10.push 9
<< s10
s11 += s3
<< s11
s12 += s1
<< s12
s13.push 8
<< s13
s5.push 4
<< s5
s1.pop
<< s1
s3 += s4
<< s3

```

Exemple de sortida 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 0 18 -14
1 -9
0
1 -3
1 13
1 1
0
1 -7
1 -10
1 2
2 2 -9
2 8 8
2 -10 2
0
2 -10 2
0
1 9
0
0
1 8
2 -3 4
2 0 18
1 14

```

Informació del problema

Autoria: PRO1

Generació: 2026-01-27T18:55:52.748Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>