

---

## Mètode de Stack per a esborrar el segon element des del topX54683\_ca

---

Implementeu un nou mètode de la classe Stack que esborri el segon element des del top. En el cas especial en que hi hagi només un element, llavors eliminarà aquest element. En el cas especial en que no hi hagi cap element, llavors no farà res.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `stack.old.hpp`, a on hi ha una implementació de la classe genèrica `Stack`. En primer lloc, haureu de fer:

```
cp stack.old.hpp stack.hpp
```

A continuació, haureu de buscar dins `stack.hpp` la part:

```
// Pre:
// Post: Elimina l'element de la pila implícita que es troba en segona posició
//       Si la pila té només un element, llavors elimina aquest element.
//       Si la pila no té cap element, llavors no canvia res.
// Descomenteu les següents dues línies i implementeu el mètode:
// void pop2() {
// }
```

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.

Preferiblement, hauríeu d'aconseguir implementar `pop2` a base de treballar amb els punters de l'objecte. De fet, una implementació a base d'usar `push` i `pop` us permetrà passar els jocs de proves públics (i així obtenir una part de la nota), però no els privats. Recordeu que és important alliberar la memòria que ja no s'utilitzarà més.

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 stack.hpp
```

### Entrada

La entrada del programa és una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre una pila que se suposa inicialment buida:

```
push x (x és un string)
pop
top
size
pop2
```

Se suposa que la seqüència d'entrada serà correcta (sense `pop` ni `top` sobre pila buida). Si que hi pot haver `pop2` sobre pila buida, cas en el qual la instrucció no té cap efecte.

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 `top`, s'escriurà el `top` actual de la pila. Per a cada instrucció `size`, s'escriurà el nombre actual d'elements de la pila. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

### Exemple d'entrada 1

```
size
pop2
size
push a
top
size
pop2
size
push b
top
push c
top
pop2
top
size
pop2
size
push a
push b
push c
push d
pop2
top
pop
top
pop2
top
```

### Exemple de sortida 1

```
0
0
a
1
0
b
c
c
1
0
d
b
b
```

### Exemple d'entrada 2

```
push tq
push t
pop
top
push a
push v
push pm
push y
pop2
top
push os
pop2
top
push b
push ex
push d
pop
top
pop2
top
pop2
top
```

```
pop
top
pop
top
push wi
push ux
pop2
top
push sl
pop2
top
size
push hz
top
```

## Exemple de sortida 2

tq  
Y  
os  
ex  
ex

ex  
v  
a  
ux  
sl  
3  
hz

## Observació

Avaluació sobre 10 punts:

- Solució lenta: 6 punts.
- solució ràpida: 10 punts.

## Informació del problema

Autor : PRO1

Generació : 2022-04-24 13:27:03

© *Jutge.org*, 2006–2022.

<https://jutge.org>