

---

## Pila sense duplicats (II)

X94265\_ca

---

Implementeu una classe per a piles *sense duplicats*. En fer un `p.push(element)`, no apilarem res a la pila `p` si l'`element` ja hi és.

La vostra implementació ha de ser senzilla i eficient (tant en espai com en temps). Comproveu els possibles errors amb assercions. Documenteu el vostre codi adequadament.

### Entrada

L'entrada serà una seqüència d'instruccions `push` (amb allò que calgui apilar, en aquests exemples farem servir enters), `pop`, `cim`, `mida` i `buida`. Vegeu els jocs de proves públics.

### Sortida

Escrivim al `stdout` el resultat de les instruccions `cim`, `mida` i `buida`.

### Observacions

Heu de baixar-vos el fitxer `code.py` (consell: renombreu-lo, ja que en alguns editors el nom `code` us pot portar problemes), l'heu de completar amb el codi que falta, i això, tot, és el que heu d'enviar al Jutge com a solució.

Penseu bé com feu que la comprovació de la presència o no d'un element a la pila sigui eficient.

L'eficiència i la qualitat de la solució es tindran en compte a la correcció manual.

#### Exemple d'entrada 1

```
mida
push 3
push 5
push 2
push 3
mida
cim
pop
cim
pop
cim
pop
buida
```

#### Exemple de sortida 1

```
0
3
2
5
3
True
```

#### Exemple d'entrada 2

```
push 2
push 2
push 2
mida
pop
buida
push 2
buida
```

#### Exemple de sortida 2

```
1
True
False
```

## **Informació del problema**

Autor : Jordi Delgado (adaptació de P91890, de Jordi Petit)

Generació : 2024-05-20 17:39:19

© *Jutge.org*, 2006–2024.

<https://jutge.org>