

---

## Màxim comú divisor de piles

Y64728\_ca

---

Implementa una funció **RECURSIVA** que, donades dues piles d'enters, retorna una altra pila que conté el màxim comú divisor de cada parell d'elements entre les dues piles, començant des del fons de cada pila. Si una de les piles s'acaba abans que l'altra, els elements sobrants s'afegeixen tal qual a la pila resultant.

La capçalera de la funció és la següent:

```
// Pre: cert  
// Post: Retorna una pila amb el MCD de cada parell d'elements de p1 i p2  
//   començant des del fons de les piles.  
stack<int> mcd_piles(const stack<int> &p1, const stack<int> &p2);
```

En cas que les dues piles tinguin un nombre diferent d'elements, els elements sense parella s'inclouen tal qual en la pila resultant.

Per exemple, donades les piles

```
p1 = [18, 24, 36, 5] (5 seria el cim de la pila)  
p2 = [12, 6] (6 seria el cim de la pila)
```

```
mcd_piles(p1, p2) = [6, 6, 36, 5]
```

### Observació

Només cal enviar el procediment demanat; el programa principal serà ignorat.

### Observació

La funció i les subfuncions que creïs han de treballar només amb piles (la classe `stack` de la biblioteca STL). Has de trobar una solució **RECURSIVA** i eficient del problema. En particular, no hi hauria d'haver cap bucle en cap de les funcions que implementis. Si crees funcions auxiliars, afegeix-hi les corresponents **Pre** i **Post**. En les crides recursives, inclou tant la **Hipòtesi d'inducció** com la **funció de fita/decreixement** de cada crida recursiva.

### Informació del problema

Autoria: Bernardino Casas

Generació: 2026-01-25T19:56:40.632Z

© Jutge.org, 2006–2026.

<https://jutge.org>