

---

## Clojure — funció until

W94600\_ca

---

En Haskell tenim una funció *until* que, donat un predicat  $p$ , una funció  $f$  i un valor inicial  $v$ , va aplicant la funció  $f$  tal que  $v, f(v), f(f(v)) \dots$  fins que es satisfi el predicat. Per exemple:

```
(until #( > % 100) #(* 2 %) 1)
```

té com a resultat 128.

1. Implementeu la funció *until1* en Clojure utilitzant el **recur**.
2. Implementeu la funció *until2* en Clojure amb **funcions d'ordre superior**.
3. Implementeu l'algorisme d'Euclides en forma de funció *mcd* per calcular el màxim comú divisor utilitzant una de les funcions *until*.

Entrada:  $a, b$

1. Si  $a=b$ ,  $mcd=a$ , fi.
2. Si  $a>b$ , canvi de  $a$  per  $a-b$ , anar a 1.
3. Si  $a<b$ , canvi de  $b$  per  $b-a$ , anar a 1.

### Exemple d'entrada 1

```
(until1 #( > % 100) #(* 2 %) 1)
```

### Exemple de sortida 1

```
128
```

### Exemple d'entrada 2

```
(until2 #( > % 100) #(* 2 %) 1)
```

### Exemple de sortida 2

```
128
```

### Exemple d'entrada 3

```
(mcd 14 8)
```

### Exemple de sortida 3

```
2
```

## Informació del problema

Autoria: Jordi Delgado / Gerard Escudero

Generació: 2026-01-25T13:21:30.469Z

© Jutge.org, 2006–2026.

<https://jutge.org>