
Clojure — Ús de llistes per comprensió**W54264_ca**

En aquest problema heu d'implementar una sèrie de funcions usant llistes per comprensió.

1. Feu una funció *my-map* que emuli el *map* usant llistes per comprensió.
2. Feu una funció *my-filter* que emuli el *filter* usant llistes per comprensió.
3. Feu una funció *my-zip-with* que que emuli el *map* amb dues llistes usant llistes per comprensió. Podeu utilitzar el *map* a la part del *for* que va entre corxets.
4. Feu una funció *thingify* que, donades dues llistes d'enters, genera la llista que aparella els elements si l'element de la segona llista divideix al de la primera.
5. Feu una funció *factors* que, donat un natural no nul, genera la llista ordenada amb els seus factors (no necessàriament primers).

Puntuació

Cada apartat puntua 20 punts.

Exemple d'entrada

```
(my-map #(* % 2) (range 1 5))  
(my-filter odd? (range 1 5))  
(my-zip-with * (range 1 4) (range 1 4))  
(thingify (range 1 6) (range 1 3))  
(factors 24)
```

Exemple de sortida

```
(2 4 6 8)  
(1 3)  
(1 4 9)  
([1 1] [2 1] [2 2] [3 1] [4 1] [4 2] [5 1])  
(1 2 3 4 6 8 12 24)
```

Informació del problema

Autoria: Albert Rubio / Jordi Petit / Gerard Escudero

Generació: 2026-09-04T10:59:24.118Z

© Jutge.org, 2006–2026.

<https://jutge.org>