Clojure — Ús de llistes per comprensió (vanilla draft)

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.
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)
```

Metadata

```
language: ca
source: gerard@logamer:/home/gerard/docencia/cap/jutge/llistes-comprensio.pbm
generation-time: 2024-09-17 16:47:04
```

problem.ca.yml:

```
email: gerard.escudero@upc.edu
author: Albert Rubio / Jordi Petit / Gerard Escudero
title: Haskell - Ús de llistes per comprensió
```

handler.yml:

```
handler: std
compilers: RunClojure
```

scores.yml:

```
-   part: test-1
    prefix: test-1
    points: 20
-   part: test-2
    prefix: test-2
```

```
    points: 20
-   part: test-3
    prefix: test-3
    points: 20
-   part: test-4
    prefix: test-4
    points: 20
-   part: test-5
    prefix: test-5
    points: 20
```