Clojure — Ús de funcions d'ordre superior (vanilla draft)

Feu les funcions següents utilitzant funcions d'ordre superior (i altres funcions predefinides) de Clojure i sense utilitzar recursivitat.

1. Feu una funció *eql* que indiqui si dues llistes d'enters són iguals.
2. Feu una funció *prod–of–evens* que multiplica tots el nombres parells d'una llista d'enters.
3. Feu una funció *my–reverse* que inverteix els elements d'una llista d'enters.
4. Feu una funció *scalar –product* que calculi el producte escalar de dues llistes de reals de la mateixa mida.
5. Feu una funció *count–in* que, donada una llista de llistes d'elements ℓ i un element x ens torna la llista que indica quants cops apareix x en cada llista de ℓ.
6. Feu una funció *first –word* que, donat un string amb blancs i caràcacters alfabètics), en retorna la primera paraula.

Puntuació

Cada funció puntua 16 o 17 punts.

Exemple d'entrada	Exemple de sortida
`(eql '(1 2 3) '(1 2 3))`	`true`
`(eql '(1 2 3) '(3 2 1))`	`false`
`(eql '(1 2 3) '(1 2 3 4))`	`false`
`(prod-of-evens '(2 10 5))`	`20`
`(scalar-product '(2.0 1.0) '(3.0 2.0))`	`8.0`
`(count-in '((3 2 3) (3) () (2 2)) 3)`	`(2 1 0 0)`
`(first-word "  Volem pa amb oli  ")`	`Volem`

Metadata

```
language: ca
source: gerard@logamer:/home/gerard/docencia/cap/jutge/first-class.pbm
generation-time: 2024-08-30 18:04:10
```

problem.ca.yml:

```
email: gerard.escudero@upc.edu
author: Albert Rubio / Jordi Petit / Gerard Escudero
title: Clojure - Ús de funcions d'ordre superior
```

handler.yml:

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

scores.yml:

```
-   part: test-1
    prefix: test-1
    points: 17
-   part: test-2
    prefix: test-2
    points: 17
-   part: test-3
    prefix: test-3
    points: 17
-   part: test-4
    prefix: test-4
    points: 17
-   part: test-5
    prefix: test-5
    points: 16
-   part: test-6
    prefix: test-6
    points: 16
```