Clojure — Llistes infinites (vanilla draft)

L'objectiu d'aquest problema és treballar la definició de llistes infinites. Concretament, es demana que implementeu funcions que generin llistes infinites per a:

1. *ones*: genera la seqüència dels uns $1, 1, 1, 1, 1, 1, 1, 1, \ldots$.
2. *naturals*: genera la seqüència dels naturals $0, 1, 2, 3, 4, 5, 6, 7, \ldots$.
3. *enters*: genera la seqüència dels enters $0, 1, -1, 2, -2, 3, -3, 4, \ldots$.
4. *powers*−**of**−2: genera la seqüència de les potències de dos $1, 2, 4, 8, 16, 32, \ldots$.
5. *triangulars*: genera la seqüència dels nombres triangulars: $0, 1, 3, 6, 10, 15, 21, 28, \ldots$.
6. *factorials*: genera la seqüència dels nombres factorials: $1, 1, 2, 6, 24, 120, 720, 5040, \ldots$.
7. *fibs*: genera la seqüència dels nombres de Fibonacci: $0, 1, 1, 2, 3, 5, 8, 13, \ldots$.
8. *primers*: genera la seqüència dels nombres primers: $2, 3, 5, 7, 11, 13, 17, 19, \ldots$.
9. *hammings*: genera la seqüència ordenada dels nombres de Hamming: $1, 2, 3, 4, 5, 6, 8, 9, \ldots$. Els nombres de Hamming són aquells que només tenen 2, 3 i 5 com a divisors primers.

Puntuació

Cada funció puntua 11 punts.

Exemple d'entrada

```
(take 8 ones)
(take 8 naturals)
(take 8 enters)
(take 8 powers-of-2)
(take 8 triangulars)
(take 8 factorials)
(take 8 fibs)
(take 8 primers)
(take 8 hammings)
```

Exemple de sortida

```
(1N 1N 1N 1N 1N 1N 1N 1N)
(0N 1N 2N 3N 4N 5N 6N 7N)
(0N 1N -1N 2N -2N 3N -3N 4N)
(1N 2N 4N 8N 16N 32N 64N 128N)
(0N 1N 3N 6N 10N 15N 21N 28N)
(1N 2N 6N 24N 120N 720N 5040N 40320N)
(0N 1N 1N 2N 3N 5N 8N 13N)
(2N 3N 5N 7N 11N 13N 17N 19N)
(1N 2N 3N 4N 5N 6N 8N 9N)
```

Metadata

```
language: ca
source: gerard@logamer:/home/gerard/docencia/cap/jutge/llistes-infinites.pbm
generation-time: 2024-11-21 19:31:44
```

problem.ca.yml:

```
email: gerard.escudero@upc.edu
author: Albert Rubio / Jordi Petit / Gerard Escudero
title: Clojure - Llistes infinites
```

handler.yml:

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

scores.yml:

```
-   part: ones
    prefix: test-ones
    points: 11
-   part: nats
    prefix: test-nats
    points: 11
-   part: ints
    prefix: test-ints
    points: 11
-   part: powers-of-2
    prefix: test-p2
    points: 11
-   part: triangulars
    prefix: test-triangulars
    points: 11
-   part: factorials
    prefix: test-factorials
    points: 11
-   part: fibs
    prefix: test-fibs
    points: 11
-   part: primes
    prefix: test-primes
    points: 11
-   part: hammings
    prefix: test-hammings
    points: 12
```