Haskell — Usage of comprehension lists

In this problem you should implement a series of functions using comprehension lists.

1. Implement a function \( \text{myMap} :: (a \to b) \to [a] \to [b] \) that emulates \textit{map} using comprehension lists.

2. Implement a function \( \text{myFilter} :: (a \to \text{Bool}) \to [a] \to [a] \) that emulates \textit{filter} using comprehension lists.

3. Implement a function \( \text{myZipWith} :: (a \to b \to c) \to [a] \to [b] \to [c] \) that emulates \textit{zipWith} using comprehension lists and \textit{zip}.

4. Implement a function \( \text{thingify} :: [\text{Int}] \to [\text{Int}] \to [([\text{Int}, \text{Int}])] \) that, given two lists of integers, returns the list that pairs the elements if the element of the second list divides the first in the list.

5. Implement a function \( \text{factors} :: \text{Int} \to [\text{Int}] \) that, given a non-null natural number, generates the ordered list with all its factors (non necessarily primes).

Scoring

Each function scores 20 points.

Sample input

\[ \text{myMap} \ (\ast \ 2) \ [1..5] \]
\[ \text{myFilter} \ \text{odd} \ [1..5] \]
\[ \text{myZipWith} \ (\ast) \ [1..4] \ [1..4] \]
\[ \text{thingify} \ [1..6] \ [1..3] \]
\[ \text{factors} \ 24 \]

Sample output

\[ [2,4,6,8,10] \]
\[ [1,3,5] \]
\[ [1,4,9,16] \]
\[ [(1,1), (2,1), (2,2), (3,1), (3,3), (4,1), (4,2), (5,1), (6,1), (6,2), (6,3)] \]
\[ [1,2,3,4,6,8,12,24] \]

Problem information

Author: Albert Rubio / Jordi Petit
Translator: Jordi Petit
Generation: 2016-01-19 11:29:57

http://www.jutge.org