
Haskell — Usage of comprehension lists

P93588_en

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

1. Implement a function `myMap :: (a → b) → [a] → [b]` that emulates *map* using comprehension lists.
2. Implement a function `myFilter :: (a → Bool) → [a] → [a]` that emulates *filter* using comprehension lists.
3. Implement a function `myZipWith :: (a → b → c) → [a] → [b] → [c]` that emulates *zipWith* using comprehension lists and *zip*.
4. Implement a function `thingify :: [Int] → [Int] → [(Int, Int)]` that, given two lists of integers, returns the list that pairs the elements if the element of the second list divides the one in the first list.
5. Implement a function `factors :: Int → [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

```
myMap (*2) [1..5]
myFilter odd [1..5]
myZipWith (*) [1..4] [1..4]
thingify [1..6] [1..3]
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

© Jutge.org, 2006–2016.

<http://www.jutge.org>