Haskell — Definition of higher-order functions (1) P90677_en

This problem explores the definition of high-order functions on lists. Implement the following functions that work as the original Haskell functions without using the original function eachself (i.e., you cannot use *foldl* to implement *myFoldl* but you can use it to implement *myAll*). Additionally. you can only use recursion to implement *myFoldl, myFoldr, myIterate, myUntil* and *myZip*.

1. *myFoldl* :: $(a \to b \to a) \to a \to [b] \to a$
2. *myFoldr* :: $(a \to b \to b) \to b \to [a] \to b$
3. *myIterate* :: $(a \to a) \to a \to [a]$
4. *myUntil* :: $(a \to \textbf{Bool}) \to (a \to a) \to a \to a$
5. *myMap* :: $(a \to b) \to [a] \to [b]$
6. *myFilter* :: $(a \to \textbf{Bool}) \to [a] \to [a]$
7. *myAll* :: $(a \to \textbf{Bool}) \to [a] \to \textbf{Bool}$
8. *myAny* :: $(a \to \textbf{Bool}) \to [a] \to \textbf{Bool}$
9. *myZip* :: $[a] \to [b] \to [(a, b)]$
10. *myZipWith* :: $(a \to b \to c) \to [a] \to [b] \to [c]$

Scoring

Each function scores 10 points.

Sample input 1

```
myFoldl (+) 1 [1..5]
myFoldr (+) 1 [1..5]
take 10 $ myIterate (*2) 1
myUntil (>100) (*2) 1
myMap ("la "++) ["joana", "mireia"]
myFilter odd [1..10]
myAll odd [1,3,5,3,1]
myAny odd [2,4,6,8,10]
myZip [1..4] [1..3]
myZipWith (+) [1..4] [1..3]
```

Sample output 1

```
16
16
[1,2,4,8,16,32,64,128,256,512]
128
["la joana","la mireia"]
[1,3,5,7,9]
```

```
True
False
[(1,1),(2,2),(3,3)]
[2,4,6]
```

Problem information

Author: Albert Rubio / Jordi Petit
Translator: Jordi Petit

Generation: 2026-02-12T08:18:03.581Z

© *Jutge.org*, 2006–2026.
https://jutge.org