
99 problems in Haskell - Part 1 (Lists)

P65945_en

1. *myLast* :: [a] → a

Find the last element of a list. Assume the list is non empty.

2. *myButLast* :: [a] → a

Find the last but one element of a list. Assume the list has, at least, two elements.

3. *elementAt* :: Int → [a] → a

Find the *k*-th element of a list. The first element in the list is number 1. Assume the list has, at least, *k* elements.

4. *myLength* :: [a] → Int

Find the number of elements of a list.

5. *myReverse* :: [a] → [a]

Reverse a list.

6. *isPalindrome* :: (Eq a) ⇒ [a] → Bool

Find out whether a list is a palindrome.

7. *myFlatten* :: [[a]] → [a]

Flatten a two-level nested list structure.

8. *compress* :: (Eq a) ⇒ [a] → [a]

Eliminate consecutive duplicates of list elements.

9. *pack* :: (Eq a) ⇒ [a] → [[a]]

Pack consecutive duplicates of list elements into sublists. If a list contains repeated elements they should be placed in separate sublists.

10. *encode* :: (Eq a) ⇒ [a] → [(Int, a)]

Run-length encoding of a list. Consecutive duplicates of elements are encoded as lists (*n, e*) where *n* is the number of duplicates of the element *e*.

Scoring

Each item scores 10 points.

Sample input

```
myLast [1..5]
myLast "hello"
myButLast [1..5]
elementAt 3 [1..5]
myLength [1..5]
myReverse [1..5]
```

```
isPalindrome "madam"
myFlatten [[1..5],[3..4],[2..4]]
compress "aaacaabb"
pack "aaacaabb"
encode "aaacaabb"
```

Sample output

```
5
'o'
4
3
5
```

```
[5, 4, 3, 2, 1]
True
[1, 2, 3, 4, 5, 3, 4, 2, 3, 4]
"acab"
["aaa", "c", "aa", "bb"]
[(3, 'a'), (1, 'c'), (2, 'a'), (2, 'b')]
```

Problem information

Author : Jordi Petit

Generation : 2024-05-02 21:30:15

© *Jutge.org*, 2006–2024.

<https://jutge.org>