99 problems in Haskell - Part 2 (Lists, continued) P36399_en

11. **data** *ListItem a* = *Single a* | *Multiple* **Int** *a* **deriving** (**Show**)
encodeModified :: (**Eq** *a*) ⇒ [*a*] → [*ListItem a*]

    Modified run-length encoding. Modify the result of item 10 in such a way that if an element has no duplicates it is simply copied into the result list. Only elements with duplicates are transferred as (n, e) lists.

12. *decodeModified* :: [*ListItem a*] → [*a*]

    Decode a run-length encoded list. Given a run-length code list generated as specified in item 11. Construct its uncompressed version.

13. *encodeDirect* :: (**Eq** *a*) ⇒ [*a*] → [*ListItem a*]

    Run-length encoding of a list. Implement the so-called run-length encoding data compression method directly. I.e. don't explicitly create the sublists containing the duplicates, as in item 9, but only count them. As in item 11, simplify the result list by replacing the singleton lists $(1, x)$ by x.

14. *dupli* :: [*a*] → [*a*]

    Duplicate the elements of a list.

15. *repli* :: [*a*] → **Int** → [*a*]

    Replicate the elements of a list a given number of times.

16. *dropEvery* :: [*a*] → **Int** → [*a*]

    Drop every n-th element from a list.

17. **split** :: [*a*] → **Int** → ([*a*], [*a*])

    Split a list into two parts; the length of the first part is given. Do not use any predefined predicates.

18. *slice* :: [*a*] → **Int** → **Int** → [*a*]

    Extract a slice from a list. Given two indices, i and k, the slice is the list containing the elements between the i-th and k-th element of the original list (both limits included). Start counting the elements with 1.

19. *rotate* :: [*a*] → **Int** → [*a*]

    Rotate a list n places to the left. Hint: Use the predefined functions length and (++).

20. *removeAt* :: **Int** → [*a*] → (*a*, [*a*])

    Remove the k-th element from a list.

Scoring

Each item scores 10 points.

Sample input

```
encodeModified "aaaabccaadeeee"
decodeModified [Multiple 4 'a',Single 'b',Multiple 2 'c',Multiple 2 'a',Single 'd',Multiple 4 'e
encodeDirect "aaaabccaadeeee"
dupli [1, 2, 3]
repli "abc" 3
dropEvery "abcdefghik" 3
split "abcdefghik" 3
slice "abcdefghik" 3 7
rotate "abcdefgh" 3
rotate [1..10] (-2)
removeAt 2 "abcd"
```

Sample output

```
[Multiple 4 'a',Single 'b',Multiple 2 'c',Multiple 2 'a',Single 'd',Multiple 4 'e']
"aaaabccaadeeee"
[Multiple 4 'a',Single 'b',Multiple 2 'c',Multiple 2 'a',Single 'd',Multiple 4 'e']
[1,1,2,2,3,3]
"aaabbbccc"
"abdeghk"
("abc","defghik")
"cdefg"
"defghabc"
[9,10,1,2,3,4,5,6,7,8]
('b',"acd")
```

Problem information

Author : Jordi Petit
Generation : 2014-02-04 17:25:50