

99 questions/Solutions/11

From HaskellWiki

< 99 questions | Solutions

(*) Modified run-length encoding.

Modify the result of problem 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.

```
data ListItem a = Single a | Multiple Int a
    deriving (Show)

encodeModified :: Eq a => [a] -> [ListItem a]
encodeModified = map encodeHelper . encode
    where
        encodeHelper (1,x) = Single x
        encodeHelper (n,x) = Multiple n x
```

Again, like in problem 7, we need a utility type because lists in haskell are homogeneous. Afterwards we use the

```
encode
function from problem 10 and map single instances of a list item to
Single
and multiple ones to
Multiple
.
```

The ListItem definition contains 'deriving (Show)' so that we can get interactive output.

This problem could also be solved using a list comprehension like so:

```
encodeModified xs = [y | x <- group xs, let y = if (length x) == 1 then Single (head x) else |
```

In this case,

ListItem

type can be used from the above solution and

group

can be found in

Data.List

module.

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/11&oldid=57419"

Category:

- Programming exercise spoilers

-
- This page was last modified on 18 January 2014, at 19:31.
 - Recent content is available under a simple permissive license.