

Navigation

Haskell

Wiki community

Recent changes

Random page

Tools

What links here

Related changes

Special pages

Printable version

Permanent link

Page information

Cite this page

99 questions/11 to 20

[< 99 questions](#)

This is part of [Ninety-Nine Haskell Problems](#), based on [Ninety-Nine Prolog Problems](#) and [Ninety-Nine Lisp Problems](#).

Problem 11

(*) 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.

Example:

```
* (encode-modified '(a a a a b c c a a d e e e e))
((4 A) B (2 C) (2 A) D (4 E))
```

Example in Haskell:

```
λ> encodeModified "aaaabccaadeeee"
[Multiple 4 'a',Single 'b',Multiple 2 'c',
 Multiple 2 'a',Single 'd',Multiple 4 'e']
```

[Solutions](#)

Problem 12

(**) Decode a run-length encoded list.

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

Example in Haskell:

```
λ> decodeModified
      [Multiple 4 'a',Single 'b',Multiple 2 'c',
        Multiple 2 'a',Single 'd',Multiple 4 'e']
"aaaabccaadeeee"
```

[Solutions](#)

Problem 13

(**) Run-length encoding of a list (direct solution).

Implement the so-called run-length encoding data compression method directly. I.e. don't explicitly create the sublists containing the duplicates, as in problem 9, but only count them. As in problem P11, simplify the result list by replacing the singleton lists (1 X) by X.

Example:

```
* (encode-direct '(a a a a b c c a a d e e e e))
((4 A) B (2 C) (2 A) D (4 E))
```

Example in Haskell:

```
λ> encodeDirect "aaaabccaadeeee"
[Multiple 4 'a',Single 'b',Multiple 2 'c',
 Multiple 2 'a',Single 'd',Multiple 4 'e']
```

[Solutions](#)

Problem 14

(*) Duplicate the elements of a list.

Example:

```
* (dupli '(a b c c d))
(A A B B C C C D D)
```

Example in Haskell:

```
λ> dupli [1, 2, 3]
[1,1,2,2,3,3]
```

[Solutions](#)

Problem 15

(**) Replicate the elements of a list a given number of times.

Example:

```
* (repli '(a b c) 3)
(A A A B B B C C C)
```

Example in Haskell:

```
λ> repli "abc" 3
"aaabbbccc"
```

[Solutions](#)

Problem 16

(**) Drop every N'th element from a list.

Example:

```
* (drop '(a b c d e f g h i k) 3)
(A B D E G H K)
```

Example in Haskell:

```
λ> dropEvery "abcdefghik" 3
"abdeghk"
```

[Solutions](#)

Problem 17

(*) Split a list into two parts; the length of the first part is given.

Do not use any predefined predicates.

Example:

```
* (split '(a b c d e f g h i k) 3)
((A B C) (D E F G H I K))
```

Example in Haskell:

```
λ> split "abcdefghik" 3
("abc", "defghik")
```

[Solutions](#)

Problem 18

(**) 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.

Example:

```
* (slice '(a b c d e f g h i k) 3 7)
(C D E F G)
```

Example in Haskell:

```
λ> slice ['a','b','c','d','e','f','g','h','i','k'] 3 7
"cdefg"
```

[Solutions](#)

Problem 19

(**) Rotate a list N places to the left.

Hint: Use the predefined functions length and (++).

Examples:

```
* (rotate '(a b c d e f g h) 3)
(D E F G H A B C)
```

```
* (rotate '(a b c d e f g h) -2)
(G H A B C D E F)
```

Examples in Haskell:

```
λ> rotate ['a','b','c','d','e','f','g','h'] 3
"defghabc"
```

```
λ> rotate ['a','b','c','d','e','f','g','h'] (-2)
"ghabcdef"
```

[Solutions](#)

Problem 20

(*) Remove the K'th element from a list.

Example in Prolog:

```
?- remove_at(X,[a,b,c,d],2,R).
X = b
R = [a,c,d]
```

Example in Lisp:

```
* (remove-at '(a b c d) 2)
(A C D)
```

(Note that this only returns the residue list, while the Prolog version also returns the deleted element.)

Example in Haskell:

```
λ> removeAt 2 "abcd"
('b',"acd")
```

[Solutions](#)

Category: [Tutorials](#)