

ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ

Списки

Повторение

- Напишите функцию `groupElems` которая группирует подряд идущие одинаковые элементы в списке и возвращает список таких групп.
 1. С использованием рекурсии
 2. С использованием свертки
 3. С использованием стандартной библиотеки (`takeWhile,...`)

```
GHCI> groupElems []
```

```
[]
```

```
GHCI> groupElems [1,2]
```

```
[[1],[2]]
```

```
GHCI> groupElems [1,2,2,2,4]
```

```
[[1],[2,2,2],[4]]
```

```
GHCI> groupElems [1,2,3,2,4]
```

```
[[1],[2],[3],[2],[4]]
```

Упражнение

Напишите функцию, реализующую алгоритм сжатия данных, заменяющий повторяющиеся символы (серии) на один символ и число его повторов.

Функция должна преобразовать строку в список пар:

(<кол-во повторов>, <символ>).

Например:

```
ghci> encode "aaaabccaaadeeee"
```

```
[(4,'a'),(1,'b'),(2,'c'),(2,'a'),(1,'d'),(4,'e')]
```

Напишите функцию, которая будет декодировать строку, закодированную предыдущей функцией. Например:

```
ghci> decode [(4,'a'),(1,'b'),(2,'c'),(2,'a'),(1,'d'),(4,'e')]
```

```
"aaaabccaaadeeee"
```

Упражнение

Реализовать функцию `capitalize`, которая принимает строку и делает заглавной первую букву каждого слова в ней.

Используйте функции:

`toUpper :: Char -> Char` (из модуля `Data.Char`)

`words :: String -> [String]`

`unwords :: [String] -> String`

`capitalize :: String -> String`

`capitalize = undefined`

`>capitalize "goodbye cruel world"`

`"Goodbye Cruel World"`

Упражнение

Реализуйте функцию `delAllUpper`, удаляющую из текста все слова, целиком состоящие из символов в верхнем регистре. Предполагается, что текст состоит только из символов алфавита и пробелов, знаки пунктуации, цифры и т.п. отсутствуют.

```
> delAllUpper "Abc IS not ABC"  
"Abc not"
```

Упражнение

- Реализуйте шифр Цезаря. Функция должна принимать строку, которую нужно зашифровать, и сдвиг.