

Задание 5. Свёртки и другие операции над списками

Предисловие

При выполнении задания **допускается** (и поощряется) использование функций, определённых в модуле `Data.List` (**кроме функции** `group`).

Определения

- Кодирование длин серий (англ. run-length encoding, RLE) или кодирование повторов — алгоритм сжатия данных, заменяющий повторяющиеся символы (серии) на один символ и число его повторов. Серией называется последовательность, состоящая из нескольких одинаковых символов.

Задание 1

Требуется определить функцию `toSeries :: String -> [String]`, которая разбивает переданную строку на серии таким образом, что в одну серию входит подстрока из одинаковых и идущих подряд символов:

```
GHCI> toSeries "abbbccccbbba"  
["a","bbb","cccc","bbb","a"]
```

Задание 2

Предлагается задать функцию `encodeSeries :: [String] -> [(Int, Char)]`, которая преобразует список серий в список кодирующих пар (число повторений, символ):

```
GHCI> encodeSeries ["a","bbb","cccc","bbb","a"]  
[(1,'a'),(3,'b'),(4,'c'),(3,'b'),(1,'a')]
```

Задание 3

Предлагается написать функцию `writeCode :: [(Int, Char)] -> String`, которая преобразует список кодирующих пар в выходную строку:

```
GHCI> writeCode [(1, 'a'), (3, 'b'), (4, 'c'), (3, 'b'), (1, 'a')]
"1a3b4c3b1a"
```

Задание 4 (*)

Предлагается написать функцию `makeAssoc :: Eq a => [(a, b)] -> a -> b`, которая принимает список пар (a, b) и возвращает функцию типа `Eq a => a -> b`, которая в исходном списке пар находит значение типа `b`, соответствующее указанному ключу типа `a`.

Если в списке имеется несколько пар с одинаковым ключём — функция должна вернуть значение из первой такой пары. Если в списке нет пары с таким ключём — функция должна экстренно завершиться с сообщением “no such key”.

```
GHCI> find = makeAssoc [(1, "one"), (2, "two"), (3, "three"), (1,
    "ten")]
GHCI> find 2
"two"
GHCI> find 1
"one"
GHCI> find 3
"three"
GHCI> find 4
*** Exception: no such key
```

Важно:

Итоговая функция должна быть сконструирована с помощью одной из свёрточных операций и λ -абстракции

Пример:

Для списка `[(1, "foo"), (2, "bar")]` должна быть получена функция, эквивалентная следующей:

```
find = \k ->
  if k == 1 then "foo"
  else ((\k ->
    if k == 2 then "bar"
    else ((\k -> error "no such key") k)) k)
```

Примечания

Рекомендуется обратить внимание на функцию `show`, возвращающую строковое представление переданного значения:

```
GHCI> show 42
```

```
"42"
```

```
GHCI> show [1,2,3]
```

```
"[1,2,3]"
```

```
GHCI> show (1, 'a')
```

```
"(1, 'a')"
```