

Этот список проблем взят [https://wiki.haskell.org/99\\_questions](https://wiki.haskell.org/99_questions). Убедительная просьба делать задания самостоятельно. Авторам, которые обнаружат чрезмерное сходство друг с другом, либо с предлагаемыми на сайте решениями, задания засчитаны не будут.

Все задания оцениваются в 1 балл. Совокупное количество баллов за эту работу: 10.

Для корректной оценки вашего задания автоматической системой, называйте реализуемую вами функцию *в точности так*, как она названа в примере.

**Проблема 1.** Найдите последний элемент в списке.

```
|| Prelude> myLast [1,2,3,4]
|| 4
|| Prelude> myLast ['x','y','z']
|| 'z'
```

**Проблема 2.** Найдите предпоследний элемент в списке.

```
|| Prelude> myButLast [1,2,3,4]
|| 3
|| Prelude> myButLast ['a'..'z']
|| 'y'
```

**Проблема 3.** Найдите k-й элемент списка, элементы нумеруются с 1.

```
|| Prelude> elementAt [1,2,3] 2
|| 2
|| Prelude> elementAt "haskell" 5
|| 'e'
```

**Проблема 4.** Найдите длину списка.

```
|| Prelude> myLength [123, 456, 789]
|| 3
|| Prelude> myLength "Hello , world!"
|| 13
```

**Проблема 5.** Переверните список.

```
|| Prelude> myReverse "A man, a plan, a canal, panama!"
|| "!amanap ,lanac a ,nalp a ,nam A"
|| Prelude> myReverse [1,2,3,4]
|| [4,3,2,1]
```

**Проблема 6.** Определите, является ли список палиндромом, т.е. читается ли он одинаково слева направо и справа налево, например “хатах”.

```
|| Prelude> isPalindrome [1,2,3]
|| False
|| Prelude> isPalindrome "madamimadam"
|| True
|| Prelude> isPalindrome [1,2,4,8,16,8,4,2,1]
|| True
```

**Проблема 7.** Превратите вложенный список в плоский. Вложенный список описывается типом данных *NestedList*.

```

|| data NestedList a = Elem a | List [NestedList a]
|| Prelude> flatten (Elem 5)
|| [5]
|| Prelude> flatten (List [Elem 1, List [Elem 2, List [Elem 3, Elem 4], Elem 5]
|| [1,2,3,4,5]
|| Prelude> flatten (List [])
|| []

```

**Проблема 8.** Удалите последовательно идущие дубликаты из списка, заменяя последовательные вхождения одинаковых элементов единственным вхождением. Порядок элементов должен остаться неизменным.

```

|| Prelude> compress "aaaabcsaadeeee"
|| "abcade"

```

**Проблема 9.** Упакуйте последовательно идущие дубликаты элементов в отдельные подсписки.

```

|| Prelude> pack ['a', 'a', 'a', 'a', 'b', 'c', 'c', 'a',
||               'a', 'd', 'e', 'e', 'e', 'e']
|| [ 'aaaa', 'b', 'cc', 'aa', 'd', 'eeee' ]

```

**Проблема 10.** По аналогии с предыдущим заданием, упакуйте список, заменяя последовательно идущие дубликаты парами (N E), где N - количество последовательно идущих дубликатов элемента E.

```

|| Prelude> encode "aaaabcsaadeeee"
|| [(4, 'a'), (1, 'b'), (2, 'c'), (2, 'a'), (1, 'd'), (4, 'e')]

```