

1. [2 балла] Напишите функцию `pairSwap :: [a] -> [a]`, которая попарно меняет местами подряд идущие элементы: первый со вторым, третий с четвертым и т.д. Последний нечетный элемент (если он есть) остается на своем месте.

```
ghci>pairSwap "abcde"
"badce"
ghci>pairSwap "abcdef"
"badcfe"
```

2. [3 балла] Напишите функцию `zipN :: [[a]] -> [[a]]`, которая похожа на `zip`, но работает с произвольным количеством списков, помещенных в один входной список. Первым элементом результирующего списка является список первых элементов каждого из списков, вторым элементом - список вторых и так далее, пока какой-либо из списков не закончится:

```
ghci>zipN ["abcdef", "12345678"]
["a1", "b2", "c3", "d4", "e5", "f6"]
ghci>zipN ["abc", "banana", "crocodile"]
["abc", "bar", "cno"]
ghci>zipN [[1..3], [2..], [3..]]
[[1,2,3], [2,3,4], [3,4,5]]
ghci>zipN []
[]
```

3. [3 балла] Напишите функцию `capitalize :: String -> [String] -> String`, которая переводит первую букву каждого слова, кроме слов исключений, входной строки в верхний регистр. Все остальные буквы переведите в

нижний регистр. Список слов исключений функция принимает в качестве второго аргумента.

```
ghci>capitalize "this is a sentence." ["a", "is"]  
"This is a Sentence."
```

4. [2 балла] Напишите функцию `sublist :: (Eq a) => [a] -> [a] -> Bool`, которая проверяет является ли один список подпоследовательностью в другом. Т.е. каждый элемент первого списка входит во второй список в *том же порядке*:

```
ghci> sublist [1, 2, 3] [100, 1, 101, 2, 102, 3]  
True  
ghci> sublist [1, 2, 3] [100, 1, 101, 2, 102, 4]  
False
```

5. [3 балла] Используйте `foldl` или `foldr`, чтобы реализовать:

1. функцию, находящую наибольший общий делитель непустого списка значений `gcdl :: (Integral a) => [a] -> a`
2. конкатенацию двух списков `append :: [a] -> [a] -> [a]`
3. функцию, добавляющую элемент в конец списка `snoc :: a -> [a] -> [a]`

6. **Теория множеств**. В этом задании будем представлять множества в виде списков без повторов (т.е. все значения в списке уникальные).

1. [2 балла] Реализуйте функцию `toSet :: Eq a => [a] -> [a]`, которая превращает любой список в

множество. Не используйте `nub`, напишите свою реализацию!

```
ghci>toSet [1,2,2,3,1]
[1,2,3]
ghci>toSet [[2], [1], [2]]
[[2], [1]]
```

2. [2 балла] Реализуйте функцию `power :: Eq a => [a] -> [[a]]`, которая возвращает множество всех подмножеств заданного множества

```
ghci>power [1, 2]
[], [1], [2], [1,2]
```

3. [3 балла] Два множества A и B называются сравнимыми, если $A \subseteq B$ или $B \subseteq A$. В противном случае они несравнимы. Антицепью называется подмножество частично упорядоченного множества, в котором любые два элемента несравнимы. Для заданного множества множеств проверьте, является ли оно антицепью `isAntichain :: Eq a => [[a]] -> Bool` (т.е. каждая пара множеств несравнима)

```
ghci>isAntichain []
True
ghci>isAntichain [[]]
True
ghci>isAntichain [[1],[2]]
True
ghci>isAntichain [[1],[2,3],[1,4,5]]
False
```

4. [бонус] Сгенерируйте все антицепи множества всех подмножеств множества $\{1, 2, 3, \dots, N\}$. Например, для $N = 2$ антицепи: $\{\}$, $\{\{\}\}$, $\{\{1\}\}$, $\{\{2\}\}$, $\{\{1\}, \{2\}\}$, $\{\{1, 2\}\}$