

1. [1 балл] Написать функцию слияния двух отсортированных списков. На вход функция получает два списка, **отсортированных** в порядке неубывания, возвращает список, который также отсортирован по неубыванию (вспомогательными функциями пользоваться нельзя).
2. [1 балл] Написать функцию

```
selectiveMap :: (a -> Bool) -> (a -> a) -> [a] -> [a]
```

которая принимает предикат p , функцию f и список lst , и возвращает список, в котором каждый элемент x , удовлетворяющий p , заменяется элементом $f\ x$ (остальные остаются неизменными).

Например:

```
GHCI> selectiveMap even (\x -> x*x) [1,2,3,4,5]
[1,4,3,16,5]
```

3. [1 балл] Рассмотрим следующую последовательность. Берём любое натуральное число n :
 - a. если оно чётное, то делим его на 2
 - b. если нечётное, то умножаем на 3 и прибавляем 1 (получаем $3n + 1$).

Над полученным числом выполняем те же самые действия, и так далее. **Гипотеза Коллатца** заключается в том, что какое бы начальное число n мы ни взяли, рано или поздно мы получим **единицу**.

Напишите функцию, которая для любого натурального числа вернет список элементов, где каждый следующий получен с помощью правил a-b.

Например:

```
ghci> collatz 3
[3, 10, 5, 16, 8, 4, 2, 1]
```

4. [2 балла] Реализовать функцию `permutations`, которая для данного конечного списка возвращает список всех возможных перестановок.

```
permutations :: [a] -> [[a]]
> permutations [1,2,3]
[[1,2,3],[2,1,3],[3,2,1],[2,3,1],[3,1,2],[1,3,2]]
```

5. [2 балла] Реализовать функцию `subsequences`, которая для данного конечного списка возвращает список всех возможных подпоследовательностей .

```
subsequences :: [a] -> [[a]]
> subsequences [1,2,2,3]
[[],[1],[2],[1,2],[2],[1,2],[2,2],[1,2,2],[3],[1,3],[2,3],
[1,2,3],[2,3],[1,2,3],[2,2,3],[1,2,2,3]]
```