Списки

1. Есть два класса: Address с полями улица и номер дома и Person с полями имя и Address. Нужно написать функцию:

   List<Address> getAddresses(List<Person> persons)

   то есть по списку persons возвращать список их адресов.
2. Есть список с именами: Ivan, Maria, Stephan, John, Amalia. Написать функцию, которая вернет список, в котором не содержатся имена исходного списка, длиной 4.
3. Есть два списка одинаковой длины с числами. Написать функцию, которая вернет список с элементами Yes или No, где значение на i-том месте зависит от того, равны ли элементы двух списков под номером i. Например, {1, 2, 3, 4} и {5, 2, 3, 8} вернет {No, Yes, Yes, No}
4. Есть два списка с буквами. Определить, является ли один список циклической версией другого. Например, для списков {aa, bb, cc, dd} и {cc, dd, aa, bb} результат будет true, а для {dd, ee, ff} и {dd, ff, ee} - false.
5. Написать функцию, которая реверсирует список, возвращает список элементов в обратном порядке. {1, 2, 3, 4, 5} -> {5, 4, 3, 2, 1}
6. Объединить два списка в один.
7. Есть список с целыми числами. Написать функцию, которая вернет список без элементов, больше заданного.

Множества

1. С использованием множеств (Set) реализовать функцию, которая вернет список без повторяющихся элементов:
   List<String> removeDuplicates(List<String> input)

   Пример: {“Ivan”, “Maria”, “Piotr”, “Anna”, “Maria”, “Ivan”} -> {“Ivan”, “Maria”, “Piotr”, “Anna”}
2. С использованием множеств реализовать функцию, которая вернет повторяющиеся элементы из данного списка:

   List<String> getDuplicates(List<String> input)

   Пример: {“Ivan”, “Maria”, “Piotr”, “Anna”, “Maria”, “Ivan”} -> {“Ivan”, “Maria”}

Мапы

1. Дана Map<String, String> map, написать функцию, которая вернет Мап, такую, если в есходной map есть ключ ‘a’ и ‘b’, то нужно создать новый ключ ‘ab’ с суммой значений от ключей a и b.

   Примеры:

   mapAB({"a": "Hi", "b": "There"}) → {"a": "Hi", "ab": "HiThere", "b": "There"}
   mapAB({"a": "Hi"}) → {"a": "Hi"}
   mapAB({"b": "There"}) → {"b": "There"}

2. Дан массив букв, вернуть мапу Map<String, Boolean> где каждая строка является ключем, а значением true, если строка в массиве больше одного раза и false, если только один раз.

   Примеры:

   wordMultiple(["a", "b", "a", "c", "b"]) → {"a": true, "b": true, "c": false}
   wordMultiple(["c", "b", "a"]) → {"a": false, "b": false, "c": false}
   wordMultiple(["c", "c", "c", "c"]) → {"c": true}

3. Анаграмма слова x - слово, по длине равное x и состоящее из таких же букв, что и x. Например, слово “vani” является анаграммой слова “ivan”, а слова “naan” и “anan” являются анаграммами слова “anna”. Дан словарь, сожержащий анаграммы, например {“anna”, “ivan”, “naan”, “vani”, “piotr”, “nana”, “navi”}. Написать функцию, которая по слову из словаря вернет все анаграммы этого слова, которые есть в словаре. Например, по слову “ivan” функция вернет {“navi”, “vani”}
4. Дан список имен, где некоторые имена повторяются. Написать функцию, которая по имени вернет количество вхождений этого имени в список.
5. Дан текст, подсчитать, какое слово встречается больше остальных. Слова в предложениях могут разделяться запятыми, точками, пробелами. Предлоги [в, на, из, под] и остальные не учитывать при подсчете.