Задачи I

1) Реализовать функцию-аналог map, которая принимает два аргумента – массив и функцию.
Пример работы:

```
const numbers = [1, 4, 9];
const roots = map(numbers, Math.sqrt);
// expected: [1, 2, 3]
```

Задачи II

2) Реализовать функцию partition, которая принимает два аргумента – массив и функцию. Функция возвращает кортеж из 2 массивов. Для первого массива предикат выполняется, для второго — нет.

```
var users = [
  { 'customer': 'john',  'age': 26, 'active': false },
  { 'customer': 'jonny',   'age': 34, 'active': true },
  { 'customer': 'johnson', 'age': 12,  'active': false }
];

const isActive =  (user) => { return  user.active; };
const paritionedUsers = partition(users, isActive);
const [activeUsers, otherUsers] = paritionedUsers;
```

Задачи III

3) Реализовать функцию partition, которая принимает два аргумента – массив и функцию. Функция возвращает кортеж из 2 массивов. Для первого массива предикат выполняется, для второго — нет.

```
var users = [
  { 'customer': 'john',  'age': 26, 'active': false },
  { 'customer': 'jonny',  'age': 34, 'active': true },
  { 'customer': 'johnson', 'age': 12,  'active': false }
];

const isActive = (user) => { return  user.active; };
const paritionedUsers = partition(users, isActive);
const [activeUsers, otherUsers] = paritionedUsers;
```

Реализовать с использованием разных функций:
1) forEach 2) reduce 3)* rest-оператор и рекурсия

Задачи IV

4) Дано n чисел. Известно, что в массиве содержится число, которое встречается только 1 раз. Все остальные числа встречаются 2 раза. Найти это число.
Например, для массива [2, **1**,2,3,4,3,4] ответ – 1.

Реализовать несколькими способами:
1) с использованием доп. памяти
2) без использования доп. памяти – за O(nlogn)
2*) без использования доп. памяти за O(n)

Задачи V

5) ЗАДАЧА "ИТЕРАТОР"
Перестановка - это строка-комбинация из N разных символов
Реализуйте класс согласно описанию:

- Конструктор принимает символы английского языка и длину перестановки (len)
- Доступен метод hasNext, который указывает на наличие следующей перестановки.
- Доступен метод next, который возвращает следующую перестановку длины len

```
const test = new MyIterator("bcd", 2);
test.next(); // returns "bc"
test.hasNext(); // returns true
test.next(); // returns "bd"
test.hasNext(); // returns true
test.next(); // returns "cd"
test.hasNext(); // returns false
```

Как сдавать

GITHUB GIST

1) Переходите по ссылке gist.github.com
2) Вводите любое описание -> Вставляете код
3) Нажимаете кнопку "Create secret gist" (используйте расширение *.js)

Ссылку отправляете мне в личку в tg (@vs9lh). Я изучаю код, пишу комментарии и т.д.

Если задача не будет получаться, то заранее напишите в беседу – разберём на ближайшем занятии/после занятия.