

## Вариант 2

### Задача 1

Напишите функцию `getUniqueValues`, которая принимает массив и возвращает новый массив, состоящий только из уникальных значений.

*Корректное выделение уникальных значений — 10 баллов.*

*Правильная работа с массивом и возврат результата — 5 баллов.*

### Задача 2

Напишите функцию `filterByProperty`, которая принимает массив объектов и имя свойства. Функция должна возвращать массив объектов, где данное свойство имеет значение `true`.

Например:

```
const items = [ { name: 'item1', available: true }, { name: 'item2', available: false }, { name: 'item3', available: true }, ];
```

*Корректная фильтрация объектов по значению свойства — 15 баллов.*

*Обработка массива объектов и возврат правильного результата — 5 баллов.*

### Задача 3

Создайте функцию `fetchWithLimit`, которая принимает URL и максимальное количество попыток запроса. Если запрос завершился ошибкой, функция должна повторить его указанное количество раз. Если все попытки завершились неудачей, возвращается ошибка "Запрос не удался".

*Корректное выполнение повторных попыток — 15 баллов.*

*Обработка ошибок и возврат сообщения после всех попыток — 10 баллов.*

### Задача 4

Напишите функцию `loadAllData`, которая принимает массив URL и загружает данные по каждому URL параллельно, используя `Promise.all`. Если один из запросов завершился ошибкой, функция должна вернуть сообщение "Ошибка загрузки данных".

*Корректное выполнение `Promise.all` для параллельной загрузки данных — 10 баллов.*

*Обработка ошибок и возврат правильного сообщения — 10 баллов.*

### Задача 5

Напишите асинхронный генератор `fetchPages`, который принимает базовый URL и максимальное количество страниц. Генератор должен возвращать данные для каждой страницы (`?page=1`, `?page=2` и т.д.) до достижения `maxPages`.

*Правильная реализация постраничной загрузки — 10 баллов.*

*Умение завершить генератор при достижении `maxPages` — 10 баллов.*