Цель задания

Попрактиковаться в комбинировании различных возможностей языка (в данном случае как минимум циклов, массивов, условий и функций) в одной программе. Научиться абстрагироваться от конкретной реализации и отражать это в именовании.

Задание 1

Напишите функцию **filter()**в **filter-black-list.js**, которая создаёт массив email-адресов, не попавших в чёрный список. В качестве аргументов функция должна принимать: массив строк с исходными email адресами, массив строк с email адресами в чёрном списке.

В конце файла с кодом домашнего задания напишите конструкцию export default {название функции}, чтобы была возможность автоматической проверки получившейся функции.

Проверка результата

Для проверки задания вызывайте получившуюся функцию вручную с различными параметрами и сверяйте получившийся результат с тем, что вы считаете правильным. Для этого предварительно произведите самостоятельно несколько вычислений с различными вариантами аргументов.

- Для запуска тестов локально у себя на компьютере в консоли в корне репозитория выполните команду:

npm run test:unit filter-black-list.test

Критерии оценки

Пройдены автоматические тесты. Названия переменных и аргументов достаточно информативные. Название функции информативное, не слишком общее и не слишком узкое.

Рекомендации к выполнению

Не спешите называть функцию в соответствии с формулировкой задания. Подумайте, где ещё можно применить данную функцию, и попробуйте дать ей соответствующее название.

Для проверки наличия в массиве значения можно использовать один из двух методов (array - переменная, в которой необходимо произвести проверку):

* array.indexOf('значение для проверки') вернёт индекс элемента, если он есть в массиве. А если значение не будет найдено, то метод вернёт -1.
* array.includes('значение для проверки') вернёт просто true, если элемент найден, false в противном случае.

Цель задания

Закрепить знания о функциях, их аргументах и возвращаемых значениях.

Задание 2

Напишите функцию **calculate()** в **cart-total.js**, которая вычисляет и возвращает стоимость корзины товаров после применения всех скидок. В качестве аргументов функция принимает 3 параметра:

* Общая сумма корзины
* Количество товаров в корзине
* Промокод (по умолчанию null)

Правила и порядок (порядок важен!) начисления скидок:

1. Если промокод равен 'ДАРИМ300', то из суммы вычитается 300 рублей. При этом если сумма меньше 300 рублей, то итоговая стоимость будет 0.
2. При количестве товаров в корзине ≥10 применяется скидка 5% ко всей сумме
3. При сумме, превышающей 50 000, применяется скидка 20% к сумме превышения (то есть к разнице суммы корзины и 50 000)
4. Если промокод равен 'СКИДКА15', то ко всей сумме применяется скидка 15%, но только если сумма ≥20 000

Каждая следующая скидка должна проверяться и применяться к сумме после применения предыдущих скидок.

В конце файла с кодом домашнего задания напишите конструкцию export default {название функции}, чтобы была возможность автоматической проверки получившейся функции.

- Для запуска тестов локально у себя на компьютере в консоли в корне репозитория выполните команду:

npm run test:unit cart-total.test