1. Создание приветствия. Поприветствуйте пользователя в игре. Спросите у пользователя его имя и поприветствуйте его по имени: Welcome to the Brain Games! May I have your name? John Hello, John!

Код, реализующий этот функционал, должен быть в отдельном файле src/cli.\* относительно директории проекта, а в исполняемом файле bin/brain-games.\* сделайте вызов. Для этого код нужно поместить в функцию и импортировать ее в исполняемый файл.

1. Настройка линтера. (Доп.задание лучше сделать, описание приведено только для JS, если у вас другой язык, посмотрите, как это сделать)
2. Установите следующие пакеты: eslint, eslint-config-airbnb-base, eslint-plugin-import
3. Настройте правильно .eslintrc.yml. Документация: <https://eslint.org/docs/user-guide/configuring>. Сверяйтесь с nodejs-package
4. Создайте задачу make lint, которая должна запускать npx eslint .
5. Подключите eslint к своему редактору, убедитесь что все работает. Скорее всего, понадобится поставить eslint глобально
6. Исправьте все ошибки, которые выдает линтер. Отключите правила no-console (оно не актуально для nodejs разработки) и import/extensions (чтобы получить возможность указывать при импорте файлы с расширением)
7. Игра "НОК" Необходимо реализовать игру "наименьшее общее кратное (НОК)". Суть игры в следующем: пользователю показывается три случайных числа, например, 5 7 15. Пользователь должен вычислить и ввести наименьшее общее кратное этих чисел.

Вывод должен получиться следующий:

Welcome to the Brain Games!

May I have your name? Sam

Hello, Sam!

Find the smallest common multiple of given numbers.

Question: 5 7 15

Your answer: 105

Correct!

Question: 100 50 1

Your answer: 100

Correct!

Question: 3 9 27

Your answer: 27

Correct!

Congratulations, Sam!

В случае, если пользователь даст неверный ответ, необходимо вывести:

Question: 5 10 25

Your answer: 15

'15' is wrong answer ;(. Correct answer was '25'.

Let's try again, Sam!

1. Игра "Геометическая прогрессия" Показываем игроку ряд чисел, образующий геометрическую прогрессию, заменив любое из чисел двумя точками. Игрок должен определить это число. • Рекомендуемая длина прогрессии – 10 чисел. Длина может генерироваться случайным образом, но должна содержать не менее 5 чисел • Позиция спрятанного элемента каждый раз изменяется (выбирается случайным образом)

Пример:

Welcome to the Brain Games!

May I have your name? Sam

Hello, Sam!

What number is missing in the progression?

Question: 1 2 4 8 .. 32 64 128

Your answer: 16

Correct!

Question: 1 3 9 27 ..

Your answer: 81

Correct!

Question: 5 25 .. 625 3125

Your answer: 125

Correct!

Congratulations, Sam!

В случае, если пользователь даст неверный ответ, необходимо вывести:

Question: 4 16 64 .. 1024 4096

Your answer: 1

'1' is wrong answer ;(. Correct answer was '256'.

Let's try again, Sam!

Задачи

1. Реализуйте необходимую логику для работы игры.
2. Проверьте работоспособность новой игры.
3. Добавьте в README.md аскинему (<https://asciinema.org/>) с запуском и демонстрацией различных исходов игры (дополнительное задание за доп.баллы).
4. Подключите к проекту сервис по проверке качества кода: [www.CodeClimate.com](http://www.codeclimate.com/).

Разместите значок Maintainability в файле readme вашего репозитория.

1. На данном этапе можно заметить, что игры имеют общий порядок выполнения (задать вопрос, получить ответ, сравнить ответ и т.д.). Эту логику можно вынести в одно место и переиспользовать внутри конкретных игр для устранения дублирования кода. С введением второй игры у вас появляется общая для всех игр логика (Эту логику можно поместить в отдельный файл). Главная задача этого шага – построить архитектуру запуска игр так, чтобы эта логика была в одном месте и управляла играми. Правильная организация файловой структуры хорошо помогает понимать структуру проекта. Хорошим решением будет положить все игры (и только игры) в директорию games. Максимальное количество раундов одинаково для всех игр и должно быть равно трём.

Главное в этом задании выполнить рефакторинг, в результате которого общая логика будет находиться в одном файле, очень хорошо, если реализуете движок для игр, в который будете передавать функции, с уникальным кодом для каждой игры, запуская их в движке. Все функции с побочными эффектами лучше оставить в файлах с играми и не писать их в движке.