Муниципальное Автономное Общеобразовательное Учреждение “Миасская Средняя Образовательная Школа №16”

Проект

По информатике на тему:

“Реализация генератора задач по выполнению и анализу простых алгоритмов ЕГЭ по информатике на языке JavaScript.”

Выполнили ученики 11-В класса

Зверев Денис

Дорожкин Сергей

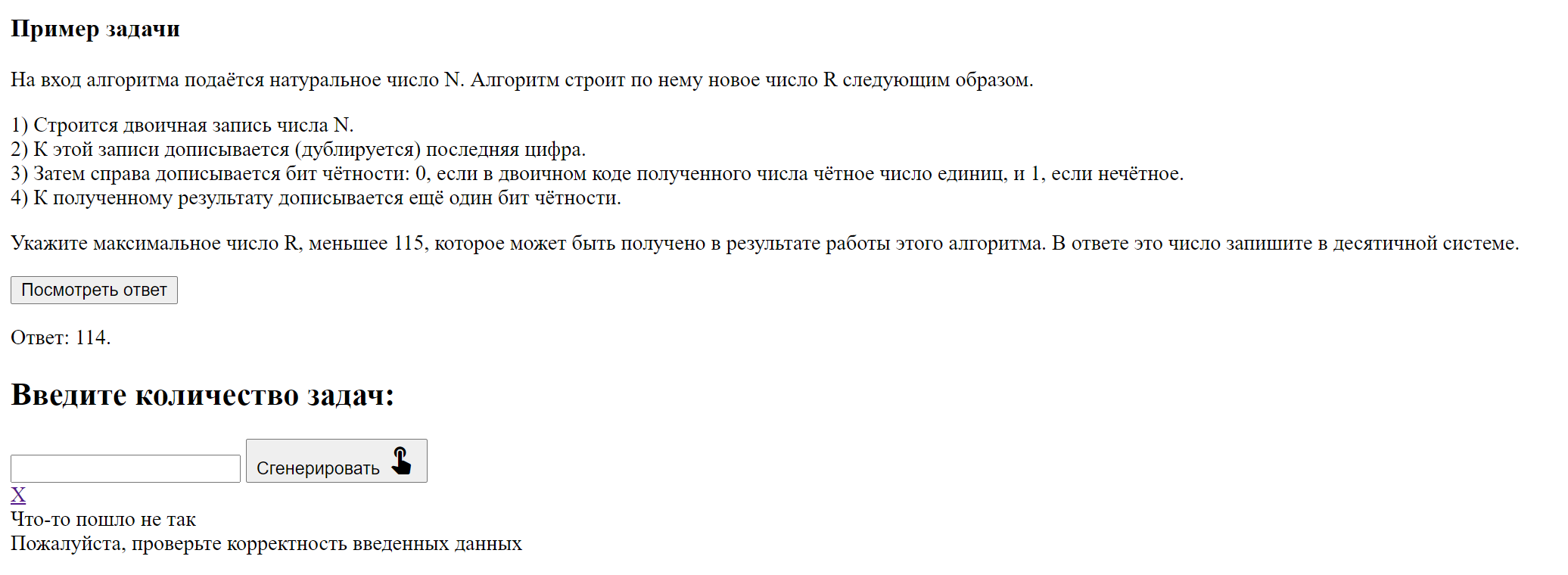
Миасс 2021.

Тема проекта: создать генератор задач по выполнению и анализу простых алгоритмов ЕГЭ по информатике на языке JavaScript. Выбранная тема является актуальной, потому что с каждым годом количество сдающих информатику увеличивается, и данный генератор поможет сдающим ЕГЭ отработать навыки в решении задач с простыми алгоритмами. Таким образом, целью проекта была разработка этого генератора. Для достижения цели потребовалось решить следующие задачи:

* Изучить HTML и CSS.
* Выделить различные подвиды задач.
* Научиться решать данные задачи.
* Ознакомиться с языком программирования JavaScript.
* Написать генератор задач с использованием скриптов JavaScript.

Проект разрабатывается с использованием web-технологий. Это вызвало необходимость изучения HTML, CSS и JavaScript. HTML – это язык, предоставляющий возможность сделать разметку веб-страниц. Он предназначен для структурирования информации на web-странице, чтобы браузер мог определить, как нужно отображать загруженный сайт. После изучения языка была реализована основная структура веб-страницы, основной макет генератора задач. Было принято решение разделить его на несколько страниц. Первая страница является главной. На ней расположен пример задачи, строка, куда нужно вписывать количество выводимых задач, а также кнопка «Сгенерировать». Эта страница позволяет указать нужное количество задач, нажать на кнопку “Сгенерировать”, после чего произойдёт переход на следующую страницу. На второй странице находятся сгенерированные в нужном количестве задачи, кнопки “Наверх” и “Показать ответы”. Таким образом, эта страница предоставляет разные задачи, а также показывает ответы на каждую сгенерированную задачу. За счёт кнопки “Наверх” можно быстро перейти к началу страницы.

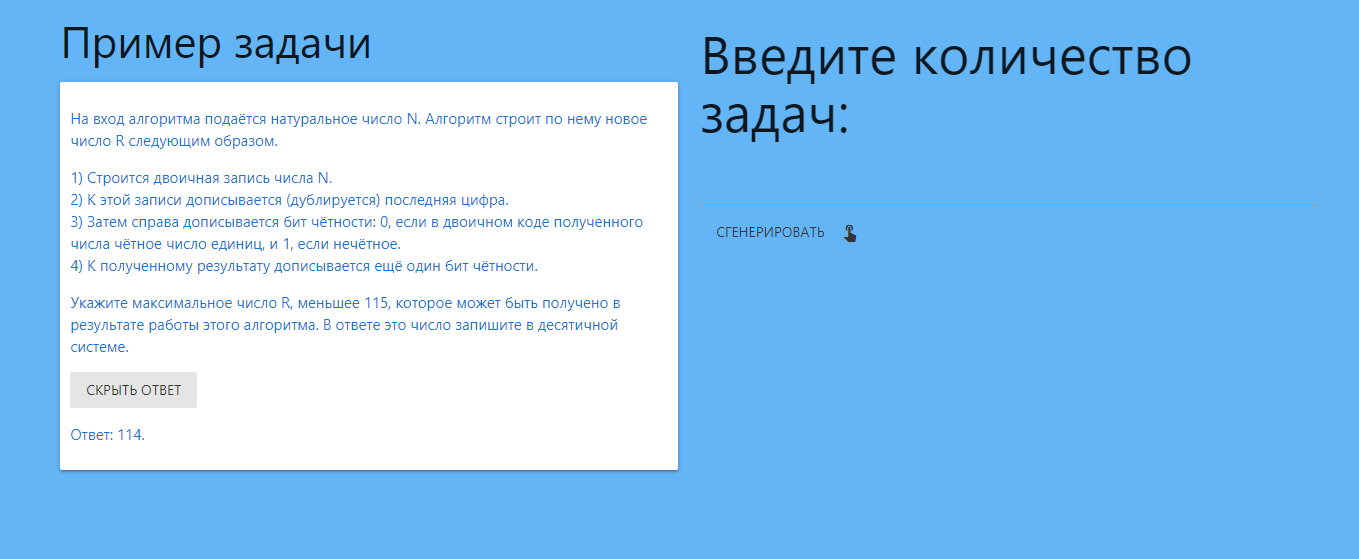
Сайт без использования CSS и Materialize:



Для придания веб-странице нужного дизайна потребовалось изучение языка CSS. CSS – это язык, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML). Он предназначен для реализации внешнего вида веб-страницы. Так как разработка дизайна, согласование стилей страницы и создание основных элементов страницы, например, кнопок, сетки для расположения элементов достаточно трудоёмкий процесс была использована библиотека Materialize.

Materialize – это новый фреймворк (полезная коллекция компонентов CSS, HTML и JavaScript, которые могут быть использованы в проектах), который использует большое количество стилей, элементов, компонентов, анимации и принципы Материального дизайна от Google. После изучения данной библиотеки и CSS было выполнено оформление элементов структуры сайта. С помощью Materialize, был сделан внешний вид сайта, расположение всех объектов и были добавлены кнопки.

Сайт с подключёнными CSS и Materialize:



По окончании реализации части сайта, было принято решение перейти к изучению задач с алгоритмами. Было проанализировано около полусотни алгоритмов и были выделены разные типы пунктов. Было выяснено, что могут быть задачи трёх типов: задачи с двумя пунктами, задачи с тремя пунктами, задачи с четырьмя пунктам. Независимо от пунктов и типа задач имеются разные условия, которые подходят к любой сгенерированной задаче. Например, были задачи с тремя пунктами действий, первым без изменения оставался перевод в двоичную систему счисления, но далее задачи становились разными:

2) К этой записи дописывается справа бит чётности: 0, если в двоичном коде числа N было чётное число единиц, и 1, если нечётное.

3) К полученному результату дописывается ещё один бит чётности.

// ИЛИ //

2) Складываются все цифры двоичной записи числа. Если сумма четная, то в конец числа (справа) дописывается 0, а если нечетная, то дописывается 1. Например, запись числа 10 преобразуется в запись 101;

3) К полученному результату применяется еще раз пункт 2 этого алгоритма.

// ИЛИ //

2) К этой записи дописывается (дублируется) последняя цифра.

3) Затем справа дописывается бит чётности: 0, если в двоичном коде полученного числа чётное число единиц, и 1, если нечётное.

Но и условия тоже могли изменяться, независимо от представленных действий. Например:

Укажите максимальное число R, меньшее Х, которое может быть получено в результате работы этого алгоритма. В ответе это число запишите в десятичной системе.

// ИЛИ //

Укажите минимальное число R, большее Х, которое может быть получено в результате работы этого алгоритма. В ответе это число запишите в десятичной системе.

// ИЛИ //

Укажите минимальное число N, после обработки которого автомат получает число, большее X. В ответе это число запишите в десятичной системе.

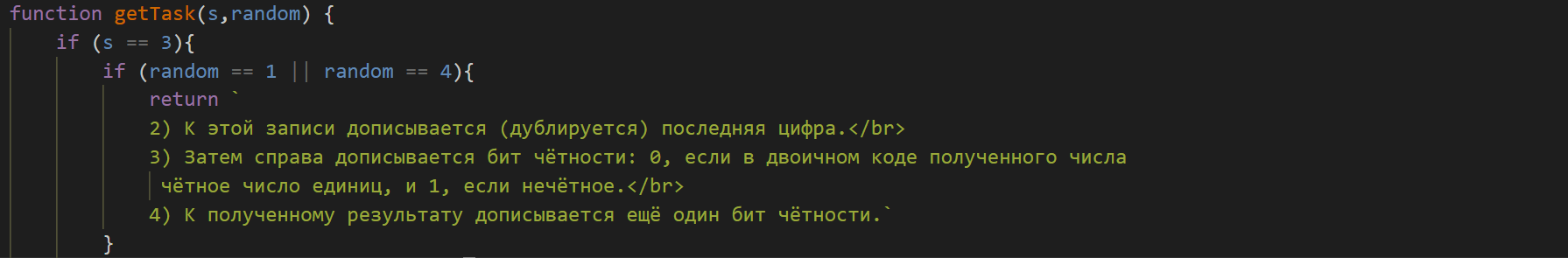
Вся совокупность, без учёта переменной Х, даёт 9 уникальных вариантов составления задачи и это только для трёх пунктов.

После чего приступили к отработке навыков по решению этих задач.

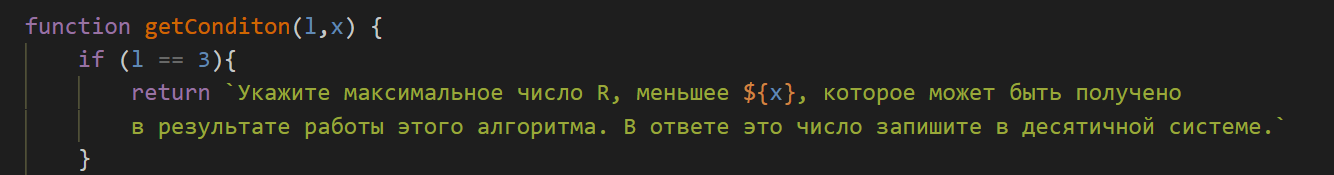
Для написания скриптов, которые генерировали бы задачи и ответы, потребовалось изучение языка JavaScript. JavaScript – это язык программирования, с помощью которого появляется возможность взаимодействовать с разными объектами на сайте. Изучив основы данного языка, а также пользуясь информацией про задачи с алгоритмом, приступили к написанию скрипта, генерирующего разные задачи. Для генерации конкретной задачи используется ряд функций, разработанных на языке JavaScript.



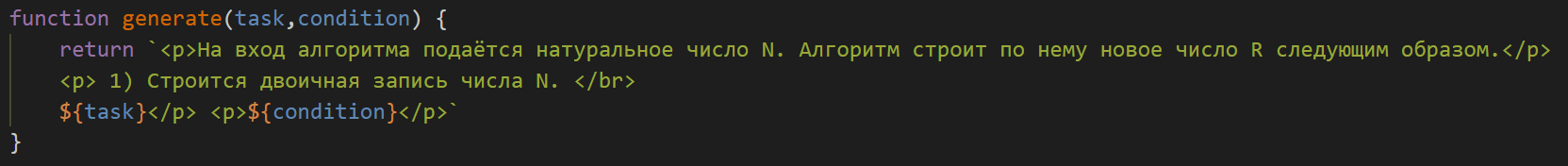
Здесь переменным присваиваются значения вызванных функций, чтобы облегчить их использование далее.



При вызове этой функции, программа определяет какое задание пойдёт в составление задачи.



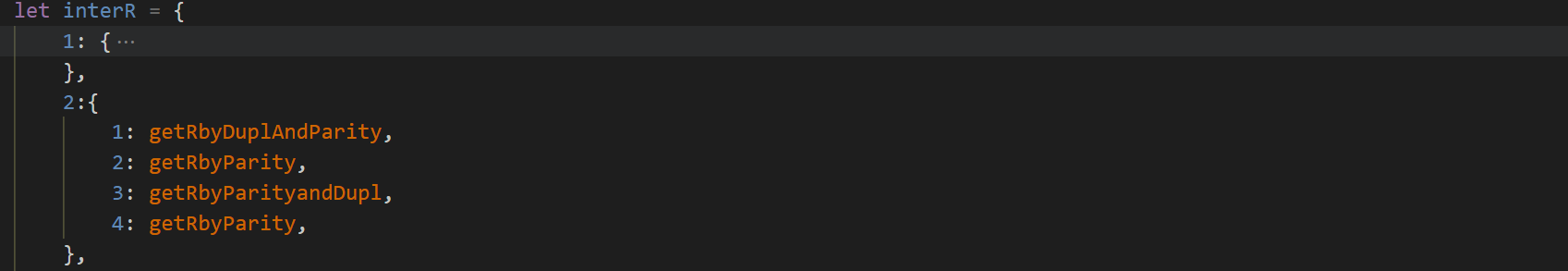
Затем избирается условие, по которому, в том числе, будет строиться ответ.



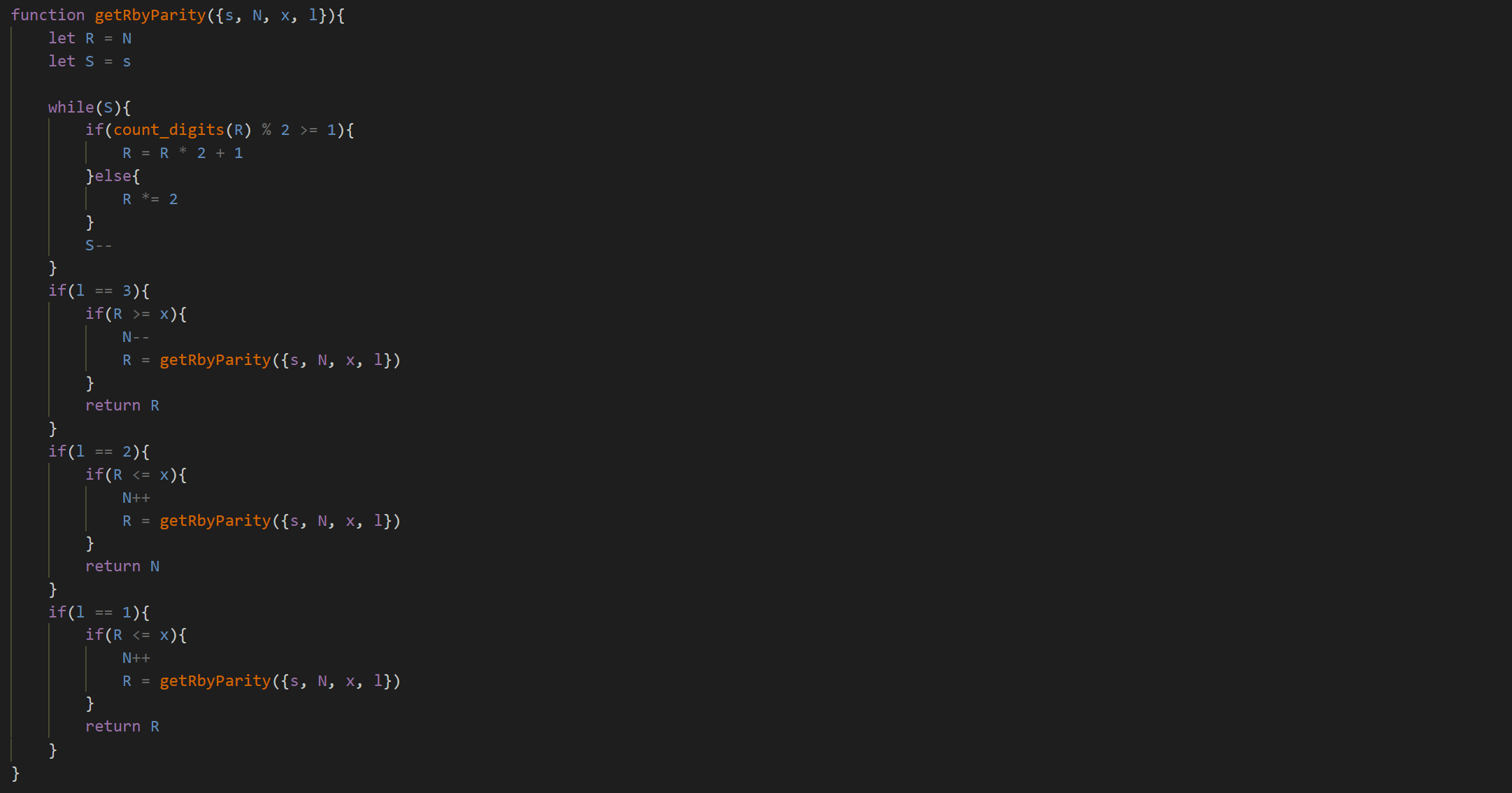
После всего вышеперечисленного строится сам текс задачи.

Таким образом видно, что задача не просто выбирается из каких-то готовых, а собирается из набора множества различных заданий и условий, позволяя иногда получать совершенно уникальные задачи.

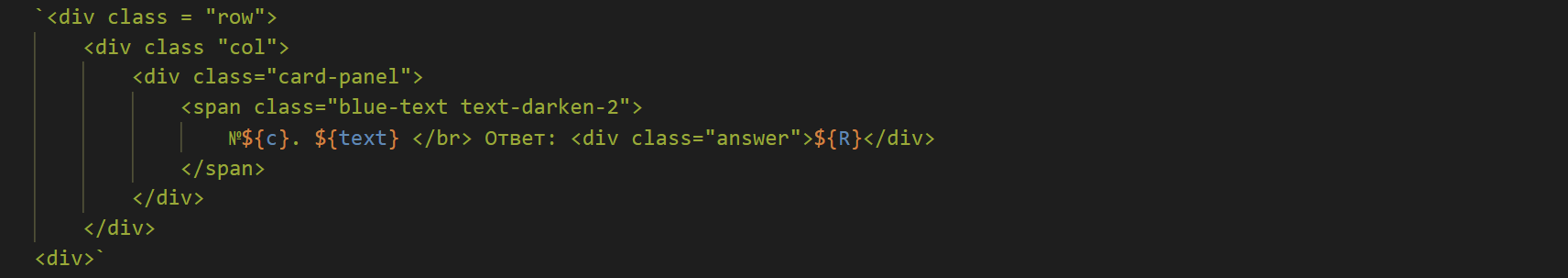
Вместе с этим сделали рабочими кнопки. На этом этапе не возникло никаких проблем, но при разработке скрипта, генерирующего ответы на задачи, возникли проблемы - не было понятно, как вставить в задачу нужный ответ. Чтобы справиться с ними, пришлось обратиться к темам в онлайн-учебнике, таким как, “Объект: основы” и “Рекурсия”, а также к отечественным и зарубежным форумам. Именно благодаря объекту получилось записать ответ функции в переменную и решить данную проблему.



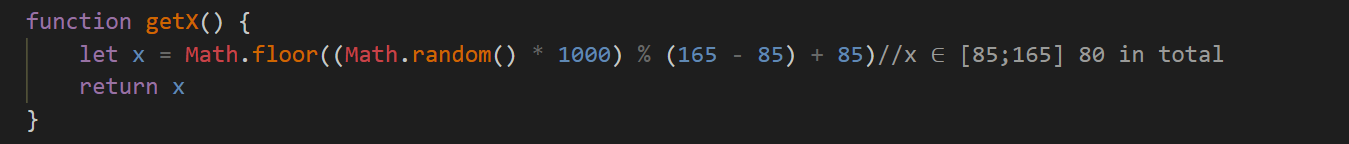
После изучения вышеуказанных тем, был создан объект, который при нужных данных позволял записать ответ в переменную.



Здесь рассматривается конкретный пример (бит чётности) получения ответа, однако вторая половина с проверкой переменной l (эль) является общей, так как там, если ответ не с первого раза удовлетворяет условию, то значение N (эн) корректируется в нужном направлении и расчёты данной функции ведутся заново.



В результате, объединяя вызовы всех этих функций, формируется задача со вставленными в неё параметрами и ответ к задаче.



Для получения большинства числовых переменных была использована функция Math.random, которая с помощью данной формулы и округлением от Math.floor до целого числа, позволяет получить число из нужного диапазона, который можно изменять и отталкиваясь от которого программа выдаёт ответ.

После проделанной работы, все задачи, которые были поставлены перед нами, выполнены. Цель проекта достигнута, продукт полностью соответствует ей. Знания и умения, приобретённые в ходе работы, будут использоваться в дальнейшем обучении.

Используемая литература:

1) Википедия – https://ru.wikipedia.org

2) Materialize – https://materializecss.com

3) Сайт Полякова – https://kpolyakov.spb.ru/