Министерство образования и науки Челябинской области

Муниципальное автономное общеобразовательное учреждение

"Миасская средняя общеобразовательная школа № 16"

Индивидуальный проект

"Разработка интерактивного рабочего листа по решению задач ЕГЭ 19-21 с использованием WEB-технологий"

Автор проекта: Петров Владислав Эдуардович,

Ученик 11-В класс МАОУ "МСОШ№16"

Наставник проекта: Горяинов Сергей Игоревич,

Учитель информатики

Миасс

2022

ЕГЭ по информатике включает в себя множество заданий с повышенной сложностью. Одни из таких заданий являются *19,20,21*. Эти номера представляют из себя упражнения, посвященные теории игр. Мы знаем много игр, которые непосредственно связаны с последовательностью ходов. Например, шашки, шахматы и т.д. Здесь - то же самое. Последовательные ходы в игре обычно совершают “Петя” и “Ваня” (“Петя” – Первый, “Ваня” – Второй). И у каждого есть выигрышная стратегия. Смысл игровых стратегий, как и любых других стратегий, заключается в том, чтобы приблизиться к цели как можно более эффективно. Либо проще и более глобально – одержать победу над противником. Для того чтобы найти выигрышную стратегию, нужно последовательно рассмотреть все возможные позиции игры. Все позиции игры можно перебрать, построив дерево игры (граф).

Основываясь на статистике от ФИПИ (*Федеральный институт педагогических измерений*) можно сделать вывод о том, что эти задания являются достаточно сложными для сдающих.



Рисунок 1. Статистика от ФИПИ

Было решено создать сайт, на котором будет размещена теория, описывающая все способы решения задач 19-21, а также тестирующая система для отработки навыков решения. Такой продукт будет полезен сдающим.

Задания 19-21 можно решать разными способами: на бумаге, в Excel, программированием. Для написания теории потребовалось достаточно много времени. Для структурирования информации способы решения разделили на две категории.

Первая категория – это “Рациональный” способ. Данный способ был взят с сайта Полякова. Проконсультировавшись с руководителем по проекту, способ был назван “Рациональным”, так как он является достаточно быстрым в решении, универсальным и удобным. К примеру, ниже описан способ решения задачи 19-21 “рациональным” способом из реального ЕГЭ по информатике 2022 года.

Задача №1. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, т. е. первым получивший кучу, в которой будет 29 или больше камней. В начальный момент в куче было S камней, 1 ≤ S ≤ 28.  
Ответьте на следующие вопросы:  
Задание 19.  
Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.  
Задание 20.Найдите два таких значения S, при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:  
— Петя не может выиграть за один ход;  
— Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.  
Найденные значения запишите в ответе в порядке возрастания.  
Задание 21.  
Для игры, описанной в задании 19, найдите значение S, при котором одновременно выполняются два условия:  
— у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;  
— у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.  
Если найдено несколько значений S, в ответе запишите минимальное из них.

Реализация будет описана на языке программирования С++.

Для решения нам понадобиться четыре библиотеки – это iostream, algorithm, vector, climits.

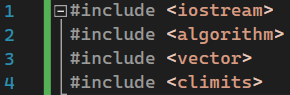


Рисунок 2. Импортирование библиотек

Дальше мы должны создать вектор (Динамический массив), чтобы туда записывать всевозможные ходы игры. Также добавим в нашу программу функцию, которая будет взаимодействовать с данными.

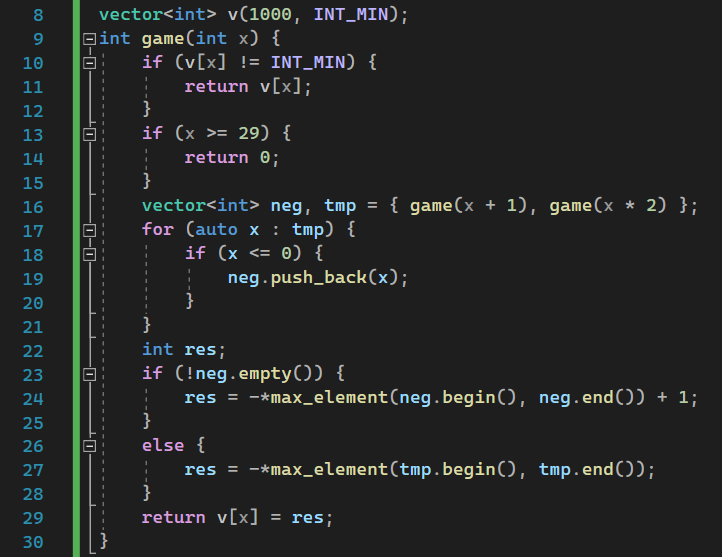


Рисунок 3. Создание вектора и главной функции

Строка 10-12. Мы возвращаем возможные “ходы”;

Строка 13-15. Проверяем условие задачи;

Строки 16. Определяем рекурсивно всевозможные ходы игроков;

Строка 17-19. “Кладем” эти ходы в вектор;

Строка 22. Создаем переменную для результата;

Строка 23. Если вектор neg не пустой, то мы должны положить в него результат;

Строка 24. Инициализируем переменную. Берем максимальное из отрицательных значений или 0, т.к. наличие отрицательного (или 0) числа говорит нам о том, что в этой комбинации есть победные значение, чтобы сходить в эту позицию, нам нужно вернуть число по модулю и добавить к нему 1;

Строка 26-28. Если в векторе tmp нет отрицательных, то игрок в позиции x и y проигрывать и его цель затянуть игру, то есть его цель – проиграть за максимальное количество шагов.

Далее в основной функции main мы должны создать цикл, который переберет количество каменей, также нужно добавить условие для вывода ответа.

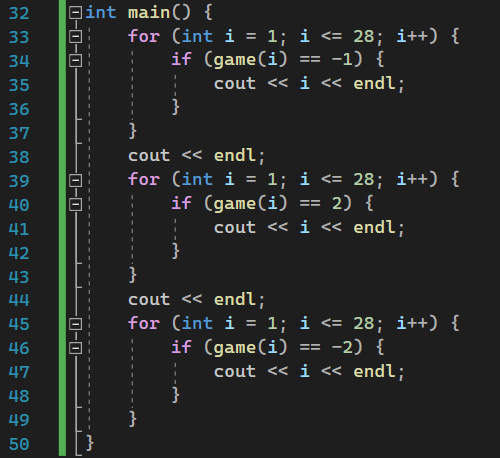


Рисунок 4. Цикл for. Условие задачи

После компиляции проекта, мы получим ответ на вопросы.

Вторая категория – это “Распространённый”. Из самых популярных источников РешуЕГЭ, code-enjoy, Stepik была структурирована теория по решению задач. Данный способ вовсе не универсальный.

Приведу пример решения задачи с сайта К.Ю. Полякова на языке Python.

Задача №2. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч один камень или увеличить количество камней в куче в два раза. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 69. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 69 или больше камней.  
В начальный момент в первой куче было 5 камней, во второй куче – S камней, 1 ≤ S ≤ 63. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.  
Ответьте на следующие вопросы:

Задание 19. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S, при котором это возможно.

Задание 20.Найдите два таких значения S, при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Найденные значения запишите в ответе в порядке возрастания.  
Задание 21. Укажите минимальное значение S, при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

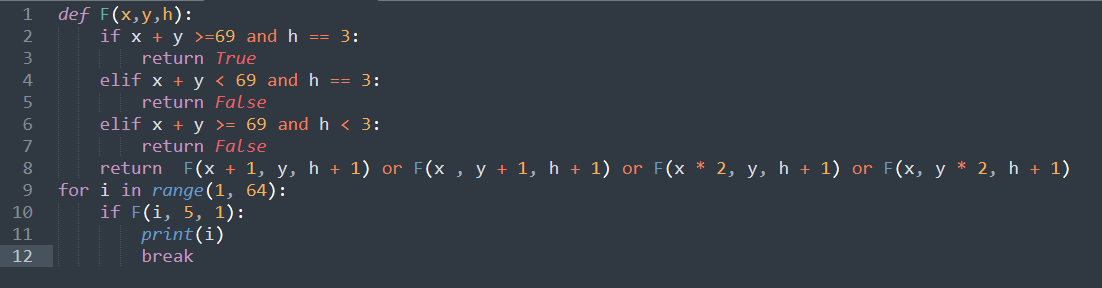


Рисунок 5. Задание 19

Функция f принимает 3 аргумента: x - число камней в первой куче, y - число камней во второй куче, h - чей ход (1 - начало игры, 2 - ход Пети, 3 - ход Вани, 4 - ход Пети и т.д.)

Строки 2-3. Если число камней в двух кучах больше или равно 69 и ход Вани, то вернуть истину.

Строки 4-5. Если ход Вани, но число камней в куче меньше 69, то вернуть ложь.

Строки 6-7. Если выиграл кто-то раньше, чем ход Вани, то вернуть ложь.

Строка 8. Продолжаем игру: функция вызывает саму себя по четырем направлениям, увеличивая при этом ход. Функция OR служит связкой этих направлений и означает "Хотя бы одно из этих направлений должно выдать истину". Если бы в условии было сказано, что "Ваня должен выиграть своим первым ходом при любом ходе Пети", то мы бы использовали функцию AND вместо OR.

Строки 10-12. Подбираем число X (число камней в первой куче), если при таком значении функция выдает истину, печатаем его. (Промежуток, который указан в коде, мы взяли не из головы, а из задачи)

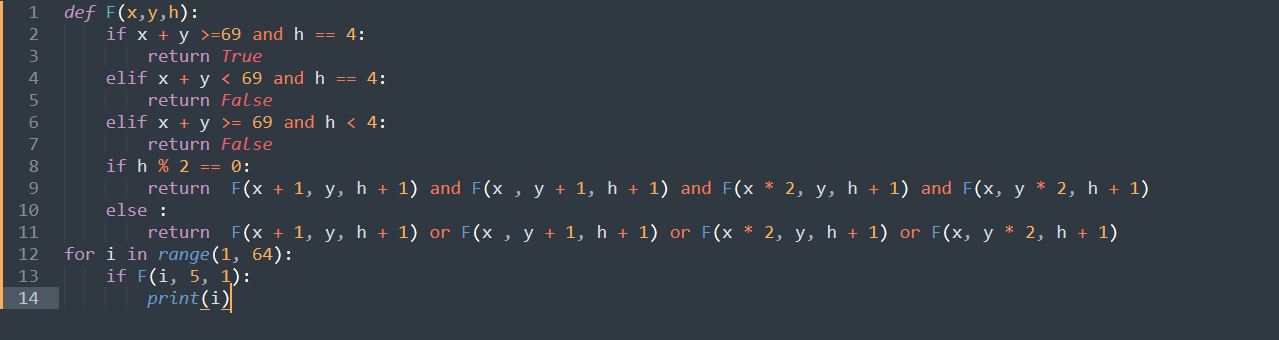


Рисунок 6. Задание 20

Для ответа на второй вопрос нужно изменить код нашей программы. Петя должен выиграть вторым ходом (h == 4)! “…два значения S… Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня…”

Значит, что Петя ходит как ему нужно (достаточно одного правильного хода из четырёх), а Ваня - как ему вздумается (все четыре хода должны привести Петю к выигрышу). Ходы Пети чётные (2, 4), ходы Вани - нечётные (3, 5). Значит после нечётного h (после Вани ходит Петя) мы будем использовать "или" (OR), а после чётного - "и" (AND).

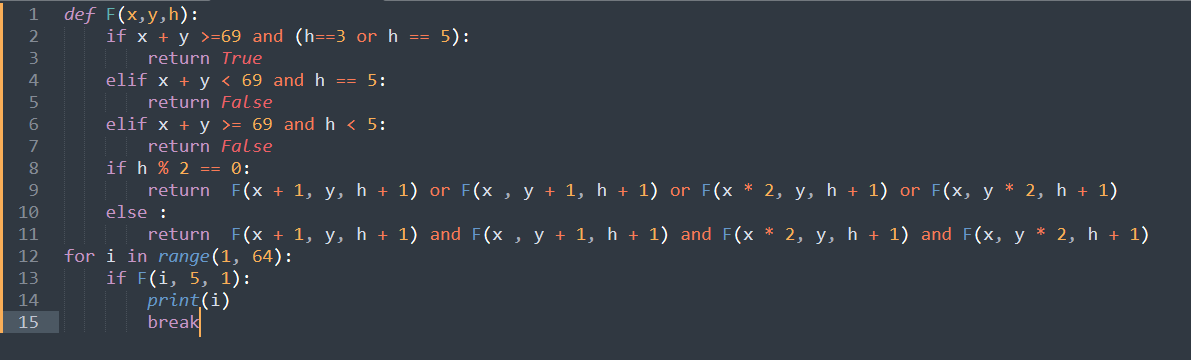


Рисунок 6. Задание 21

Теперь мы должен выиграть Ваня первым или вторым ходом (h == 3 или h == 5). Значит, Ваня ходит так, как ему нужно, а Петя - как ему вздумается. Ваня ходит после чётного (

1-Исходное; 2-Петя; 3-Ваня…, таким образом, если h чётный, то пишем "или", а если нечётный - "и".

Как видно из примера, мы постоянно подстраиваемся под вопрос. Это занимает определенное количество времени, а на экзамене этот ресурс ограниченный.

Следующая задача была заключена в том, чтобы изучить средства, позволяющие написать сайт максимально удобным и интерактивным, затем разместить на нем теорию по решению задач.

Для этих целей были выбраны такие средства: язык разметки HTML, каскадная таблица стилей CSS, язык программирования JavaScript.

Также веб-сайт должен быть адаптивным, то есть нормально функционировать на устройствах с разными размерами экранов. Для упрощения адаптации веб-сайта был использован Фреймворк (средство, которое объединяет в себе различные технологии, позволяющее упростить процесс разработки продукта) Materialize, основанный на популярном направлении, которое было разработано Google, - material design. Фреймворк предоставляет 12 колоночную разметку, которая позволяет удобно располагать элементы веб-страницы относительно друг друга, в зависимости от ширины экрана на котором просматривается содержимое страницы.

К примеру, с помощью Materialize было создано меню сайта

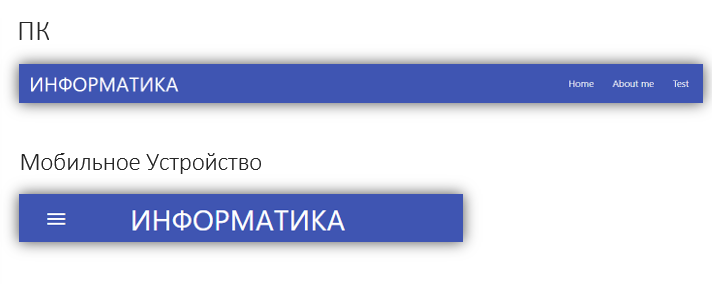


Рисунок 8. Меню сайта

Далее было размещена теория на сайт. Было решено разделить теорию на 3 разных способа решения: Excel, Программированием, на бумаге. Также нужно было представить их в виде карточек для выбора. При нажатии карточки всплывает краткая информация о статье.

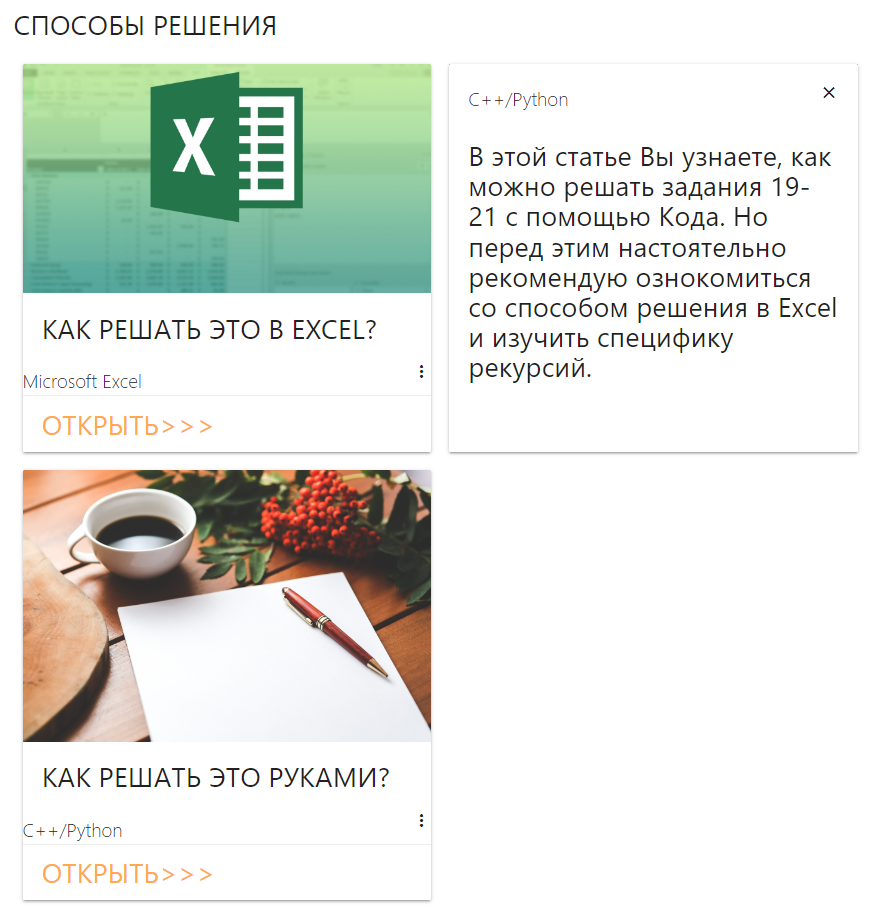


Рисунок 9. Карточки

Следующий целью было создание тестирующей системы. Планировалось вставить в тест 15 вопрос различных типов задач 19-21 и написать решение к ним (Рациональным способом). А после проверки теста нам будет показывается процент выполнения, а также неправильные и правильные ответы.

Первое, что нужно было выполнить – это отобрать и решить задачи. В основном, задачи были взяты с сайта К. Ю. Полякова. Пару задач взяты с РешуЕГЭ.

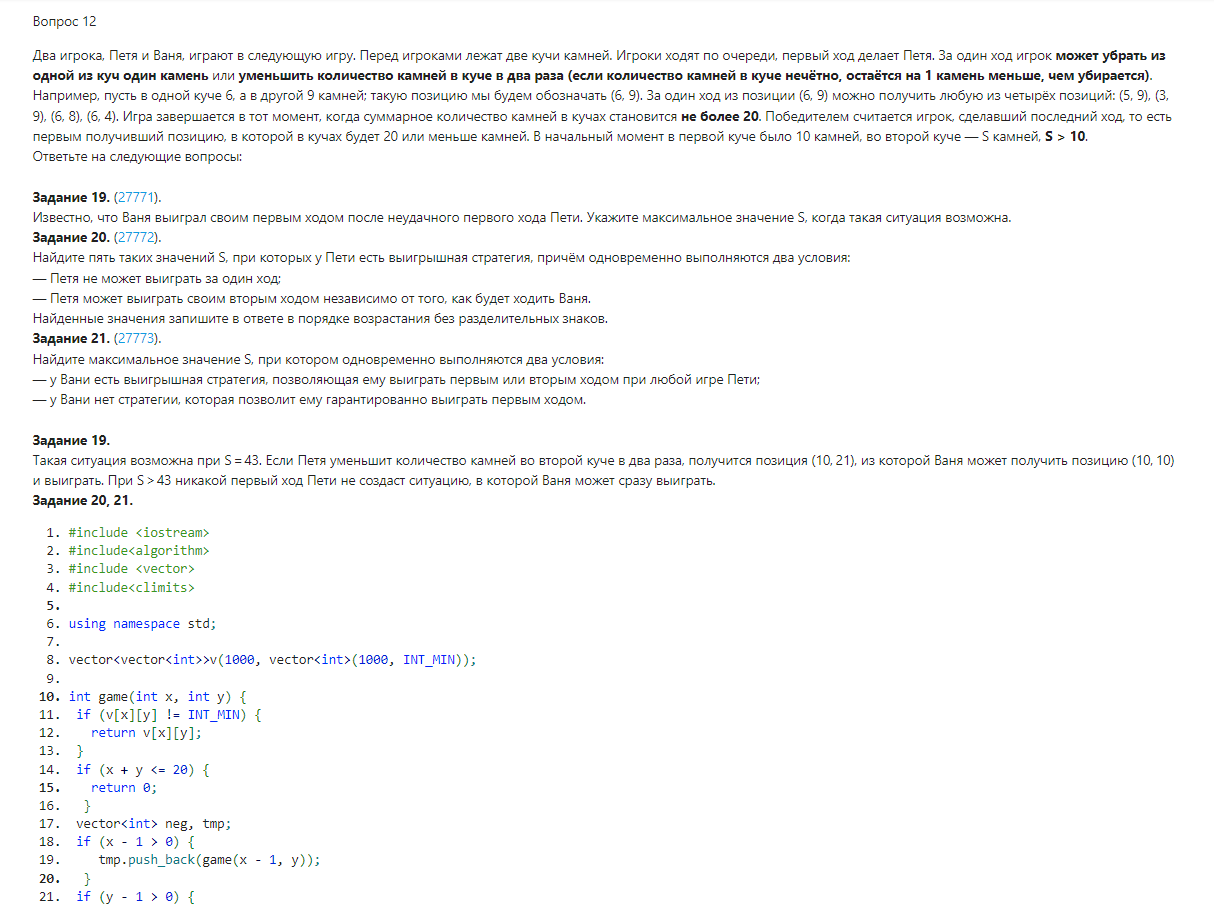


Рисунок 10. Решение 12 вопроса

После этого этапа, была написана логика теста. Данный код отвечает за переключением между вопросами

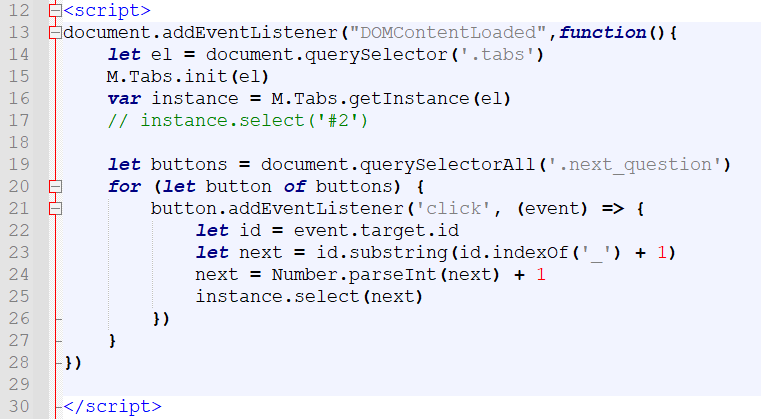


Рисунок 11. Переключение между вкладками

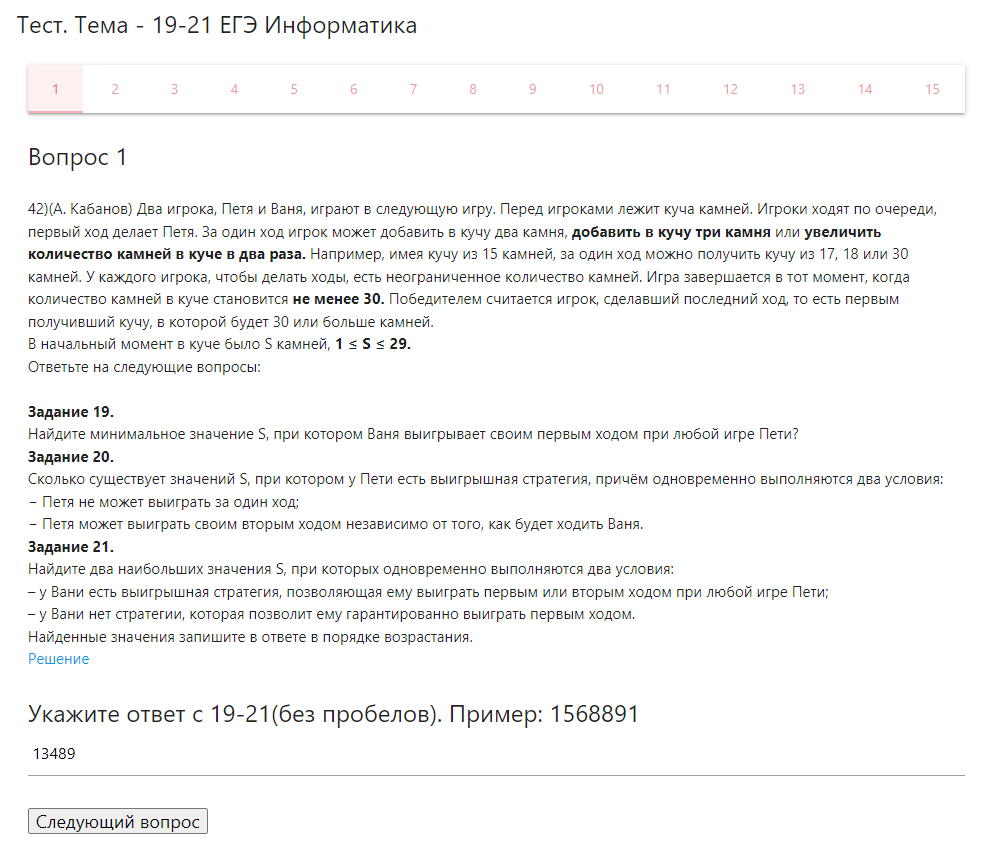


Рисунок 12. Тест

Каждая вкладка – это некий блок, при нажатие которого у нас сохраняется ответ в том блоке, где мы находимся, и переходит нам нужный.

В конце теста можно проверить ответы, проверка ответов происходит за счет этого кода

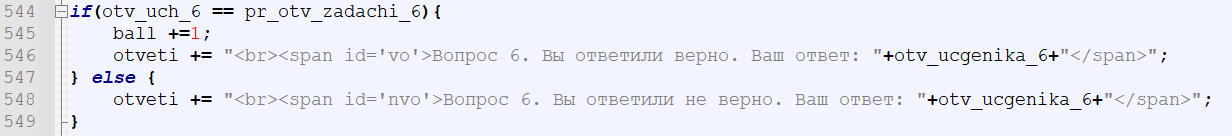


Рисунок 13. Код проверки

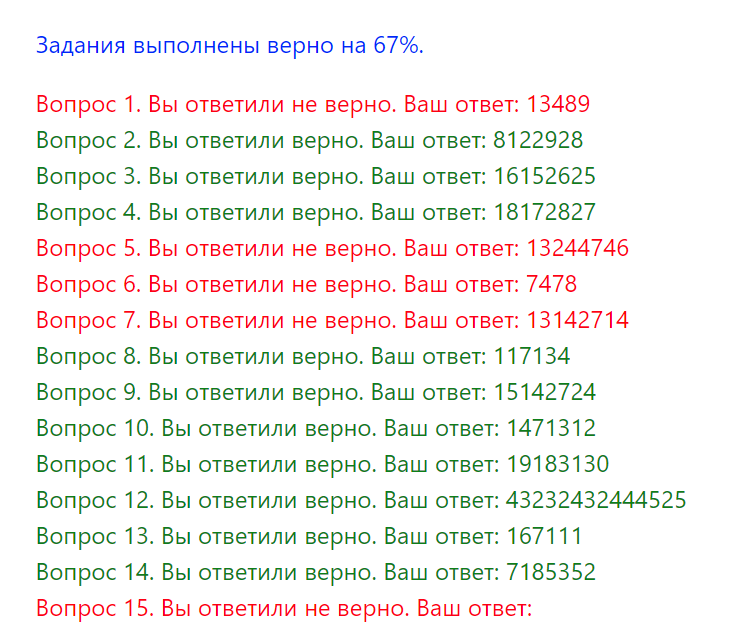


Рисунок 14. Проверка ответов

**Используемые интернет ресурсы:**

1. [https://kpolyakov.spb.ru](https://kpolyakov.spb.ru/) – сайт Константина Полякова.
2. [https://egekp.unoforum.pro](https://egekp.unoforum.pro/) – форум: “ЕГЭ по информатике”
3. <https://developer.mozilla.org/ru> – Ресурс для изучения HTML и CSS
4. [https://materializecss.org](https://materializecss.org/) – Фреймворк Materialize.
5. [https://learn.javascript.ru](https://learn.javascript.ru/) – Современный учебник JavaScript.
6. [https://code-enjoy.ru](https://code-enjoy.ru/) – Самые распространенный способы решения
7. [https://ege.sdamgia.ru](https://ege.sdamgia.ru/) – РешуЕГЭ
8. <https://welcome.stepik.org/ru> – Stepik
9. [https://fipi.ru](https://fipi.ru/) – ФИПИ. Статистика решения