**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ МОРСКОЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

(СПб ГМТУ)

**Лабораторная работа №1**

По дисциплине:

**«Программирование»**

Специальность 09.03.01.03

Подготовил студент группы 20121:

Наталенко И. А.

Проверил преподаватель:

Поделенюк П.П.

Санкт-Петербург

2022 г.

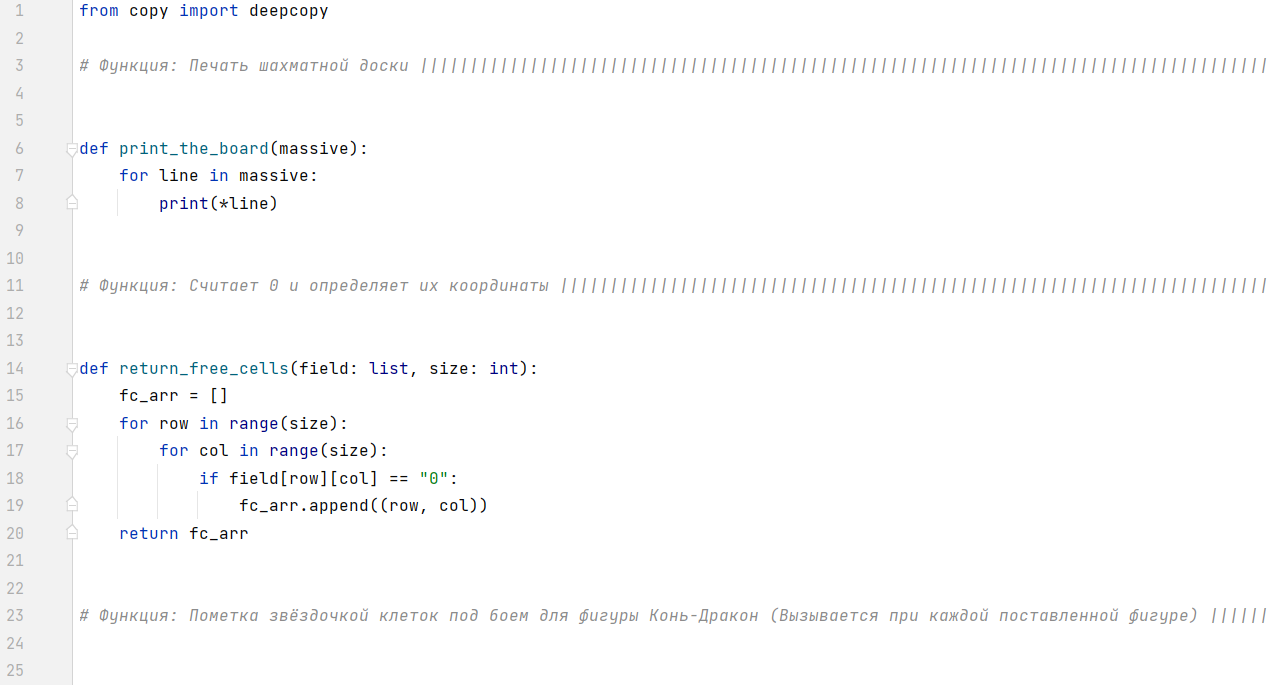
**Цель работы**:

Научиться применять на практике знания по работе с файлами, работе с несколькими функциями, работе по нисходящему методу, работе с локальными и глобальными переменными и применению рекурсии.

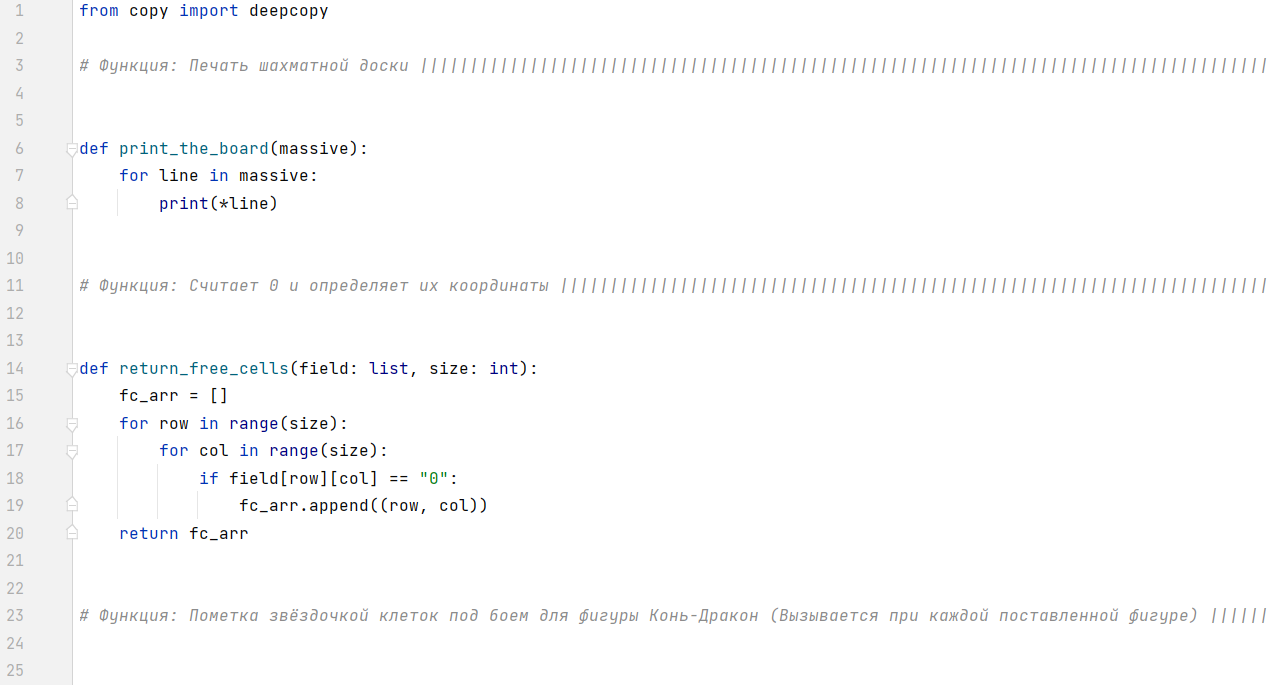
**Ход работы**:

Я написал функции, облегчающие дальнейшую работу с данными.

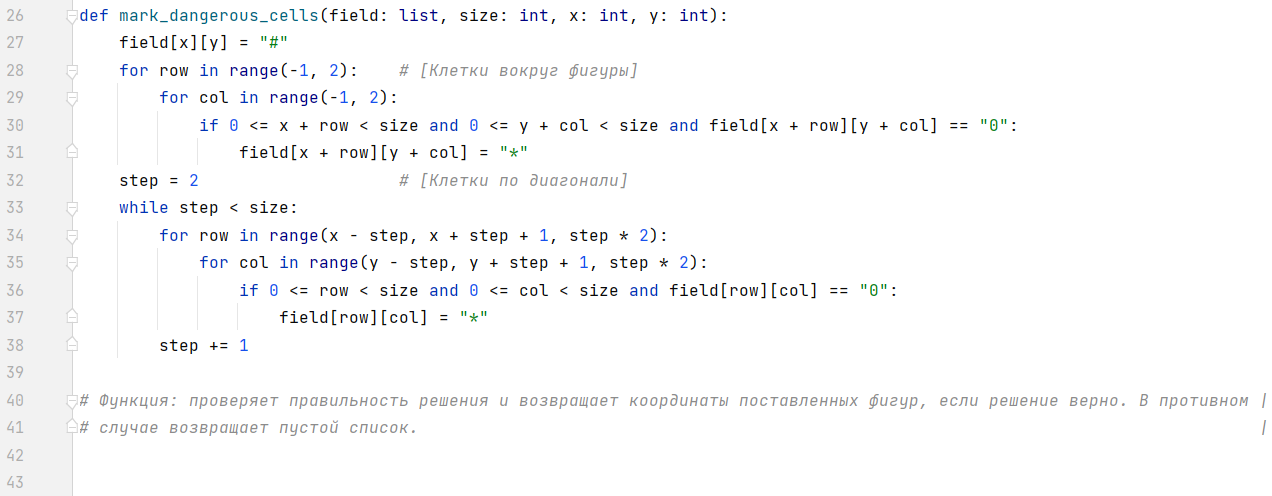
Функция print\_the\_board принимает двухмерный массив (шахматную доску) и выводит его в консоль в удобном формате – через пробелы.



Функция return\_free\_cells принимает ту же шахматную доску и её размер, проверяет, находятся ли клетки под боем и возвращает список кортежей, в которые записаны координаты свободных клеток.

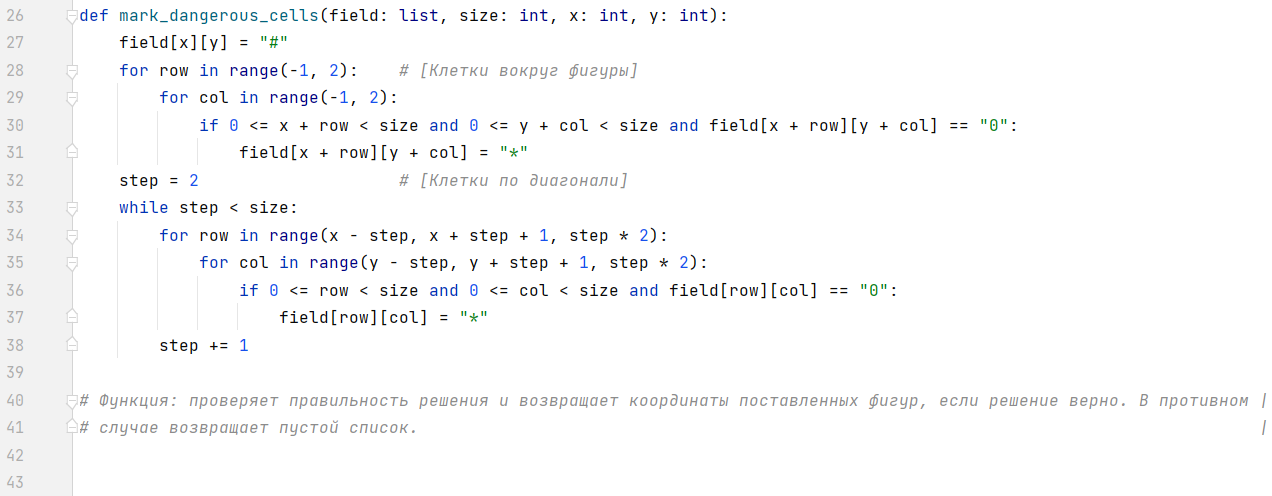


Функция mark\_dangerous\_cells принимает доску, её размер, а также координаты, куда до вызова ставится фигура, ставит её и отмечает звёздочками клетки, находящиеся под боем.

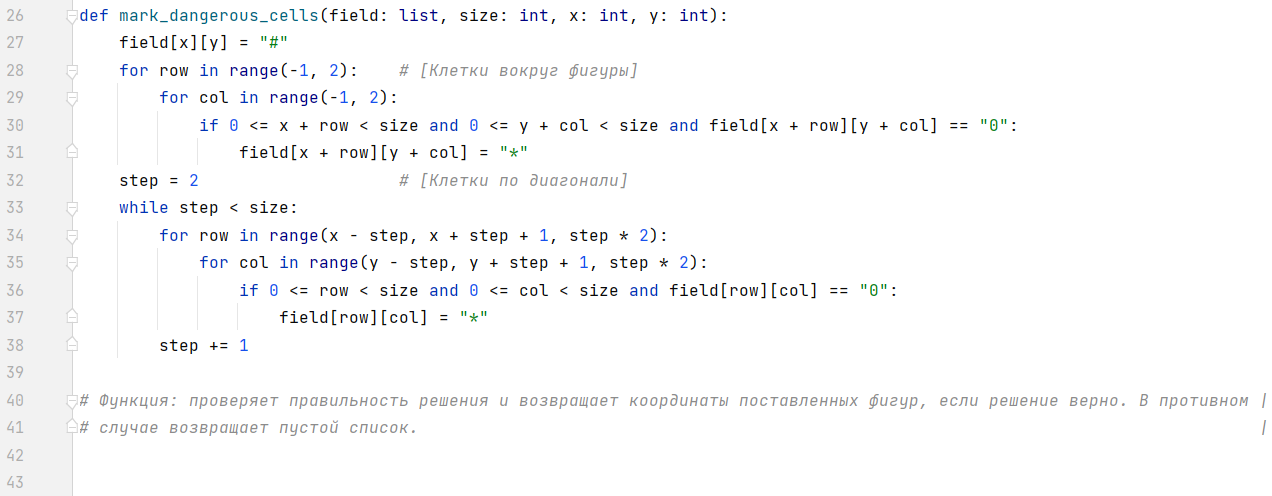


Для фигуры Конь-дракон (смесь короля со слоном) я выбрал следующий алгоритм: клетки вокруг фигуры (то есть те, на которые может походить король) помечаются отдельно от клеток по диагонали (клетки слона).

Чтобы просканировать клетки вокруг, я использовал два цикла for, которые перебирают значения на промежутке [-1;1]. Эти значения прибавляются к координатам x и y, что позволяет получить все 8 координат вокруг исходной клетки.



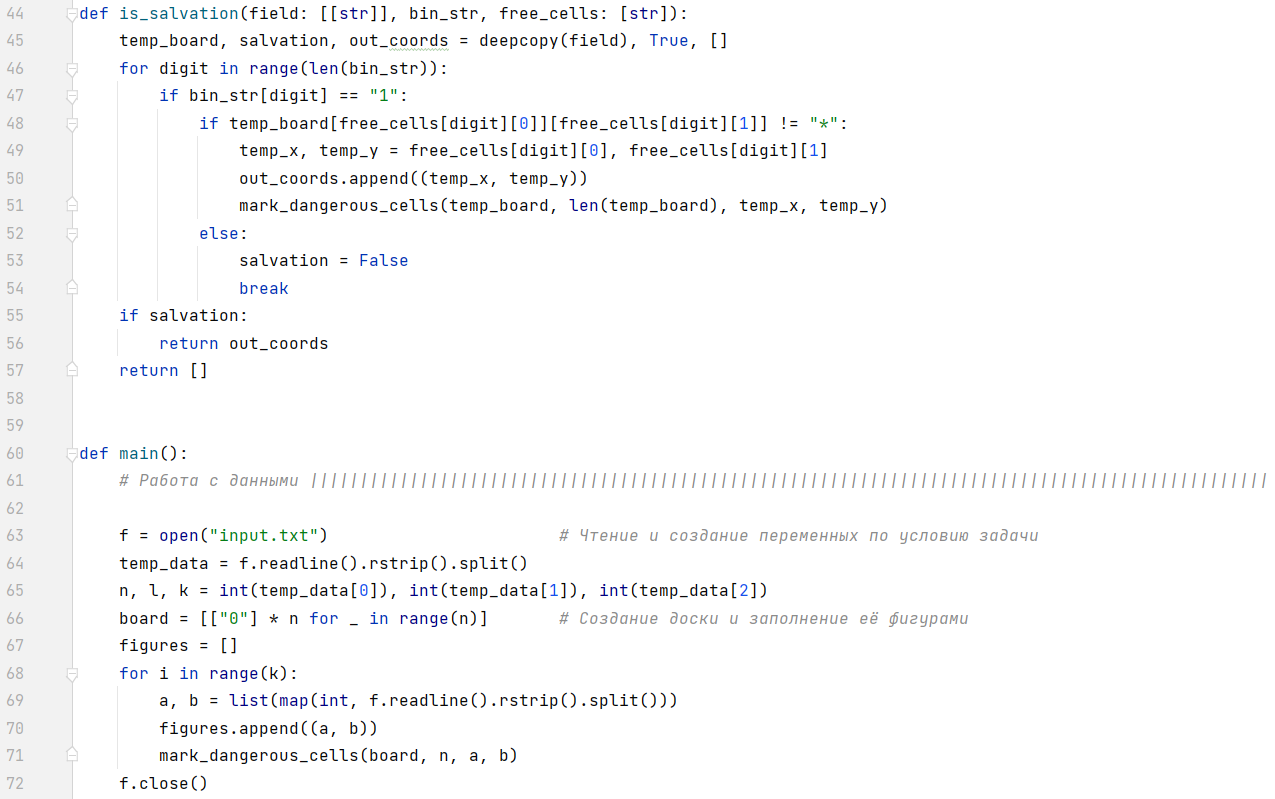
Для пометки клеток по диагонали алгоритм следующий: берётся переменная-шаг, не позволяющая координатам выйти далеко за пределы координатной системы доски. Изначально она равна 2, так как клетки на расстоянии 1 от фигуры были отмечены ранее. Чтобы не рассматривать лишние клетки, шаг в циклах равняется двойному значению переменной-шага step. При отдалении от фигуры каждый раз step увеличивается, и шаг в циклах увеличивается так, чтобы клетка с рассматриваемыми координатами всегда находилась по диагонали к фигуре.



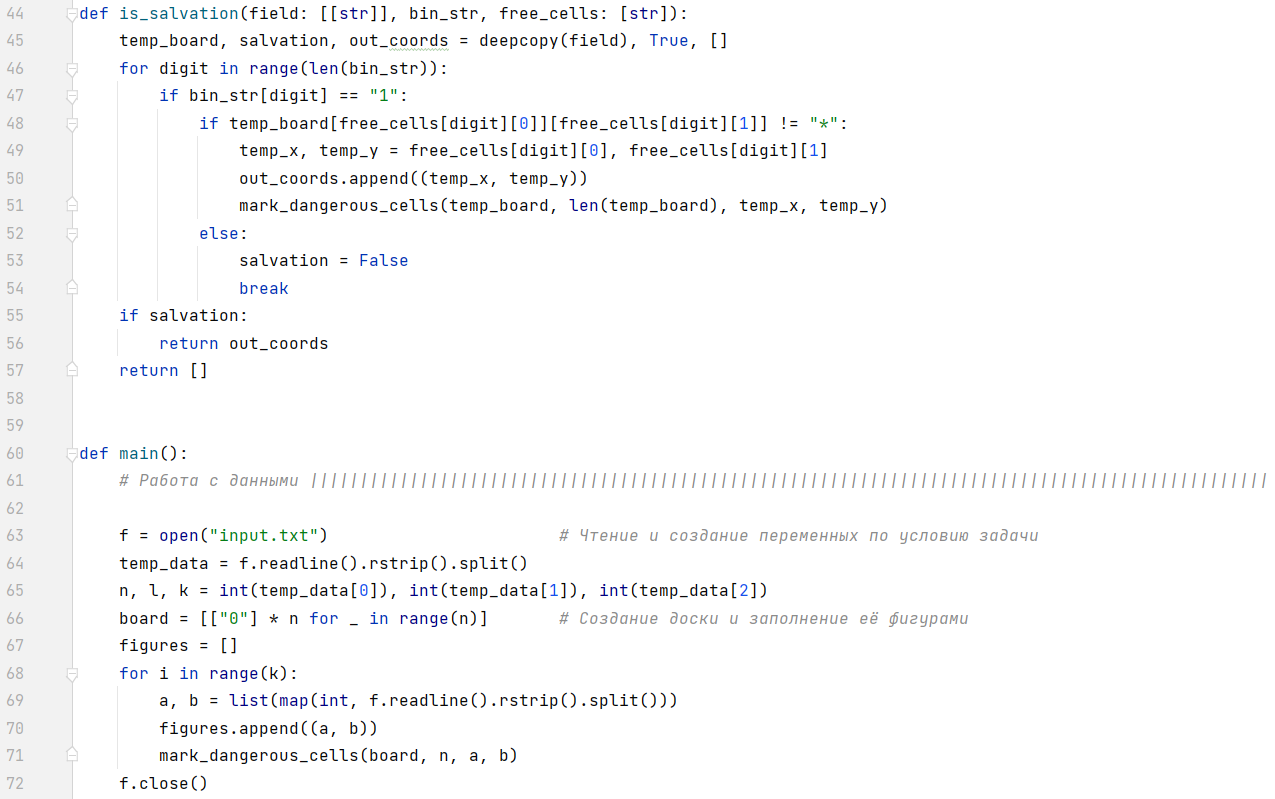
Так, на первой итерации между двумя клетками соседних диагоналей при том, что клетки находятся на одном расстоянии от фигуры, есть ещё три другие клетки. Чтобы дойти от первой до второй, всегда требуется вдвое больше шагов, чем step, в данном случае 4.

Выход из цикла производится тогда, когда step превысит размер доски. К тому времени координаты всех 4-х проверяемых клеток будут уже за границей координат доски.

Функция is\_salvation() проверяет, является ли решение в виде бинарной строки правильным. Принимает доску, саму бинарную строку и список координат свободных клеток. Здесь пригождается ранее импортированная функция deepcopy(), которая возвращает копию объекта, не являющуюся тем же объектом.

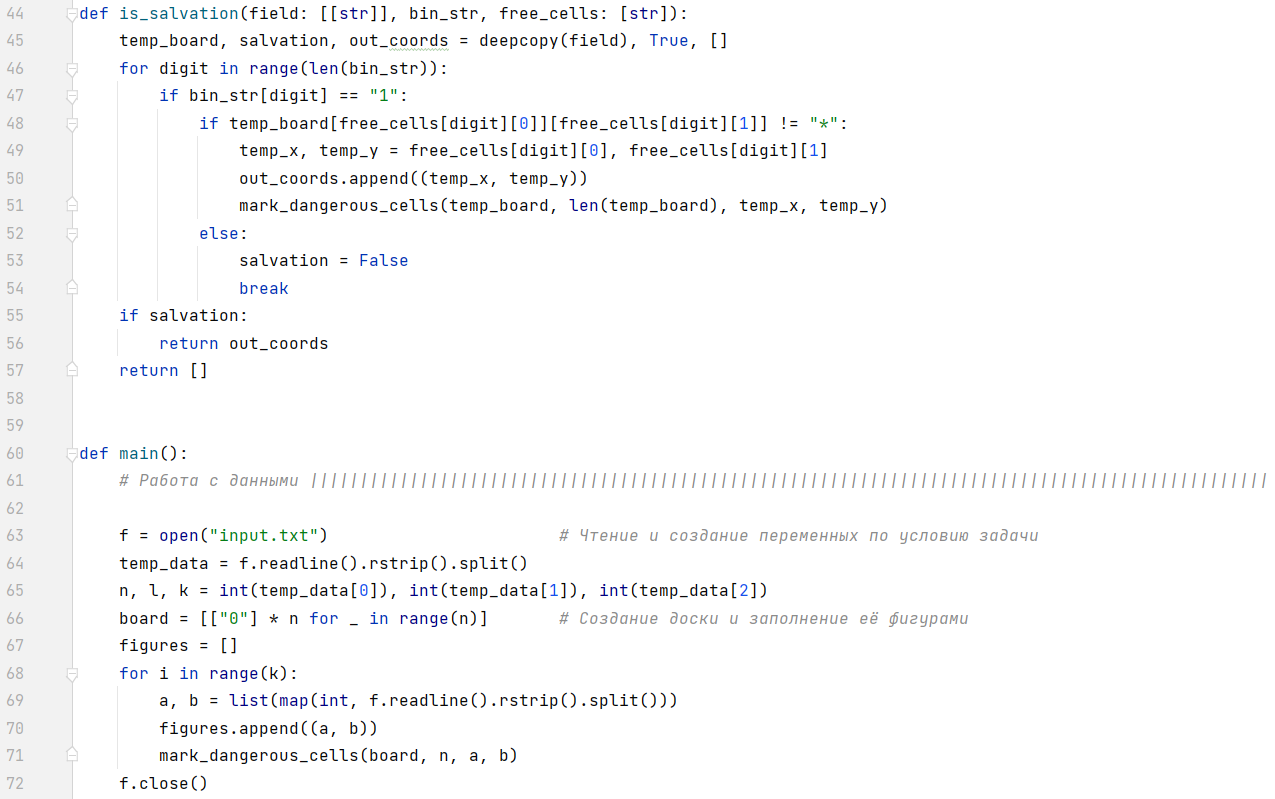


Под бинарной строкой понимается строка, в которую записаны нули и единицы. В данной задаче её длина равна длине списка координат клеток не под боем (free\_cells), и, если программа перебирает эту строку и находит единицу, это значит, что фигура должна быть поставлена. Тогда в списке свободных координат мы находим координаты по индексу digit и вытаскиваем их.



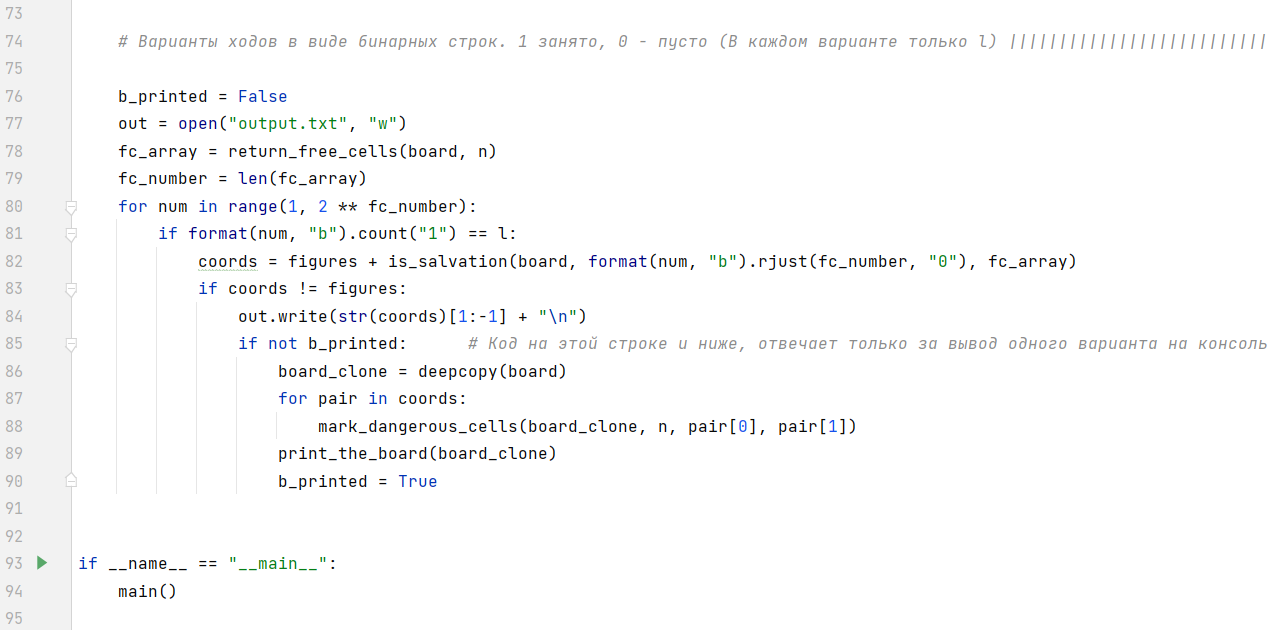
В процессе такого перебора может быть обнаружено, что фигура не может быть поставлена, так как находится под боем другой поставленной в этом переборе фигуры. Тогда переменная-флажок принимает значение False и цикл прерывается. В таких случаях вместо списка координат возвращается пустой список.

Далее идёт работа с вводным файлом, создаются переменные N, L, K с таким же смыслом, как и в условии задачи. Создаётся двухмерный массив board, заполненный нулями. Создаётся список figures, который будет заполняться координатами фигур из ввода. Он пригодится для дальнейшего вывода в файл.

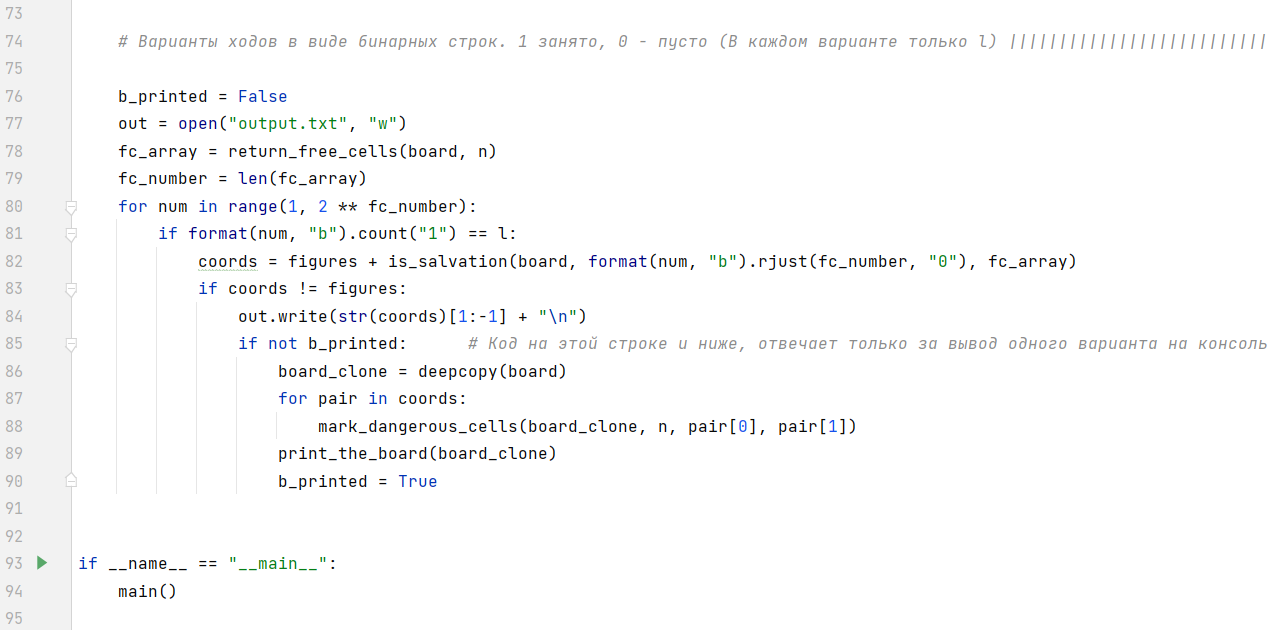


Созданная доска заполняется фигурами с координатами из входного файла, а figures – самими координатами.

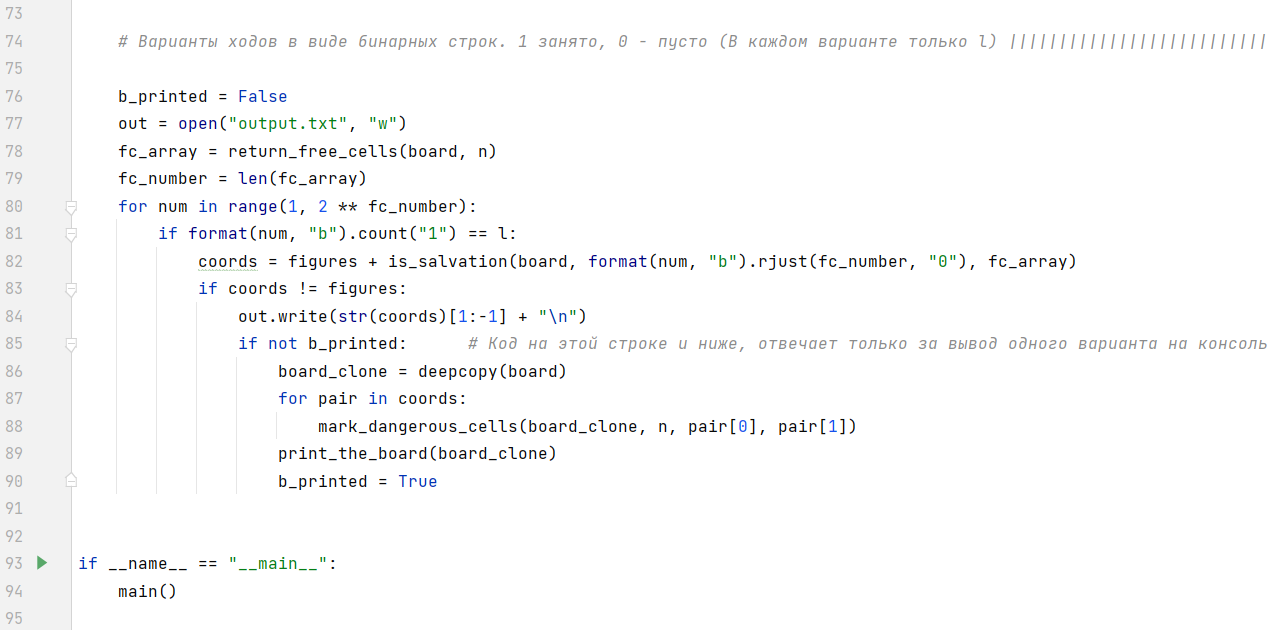
Следующим этапом идёт перебор всех существующих вариантов методом бинарных строк, описанным выше.



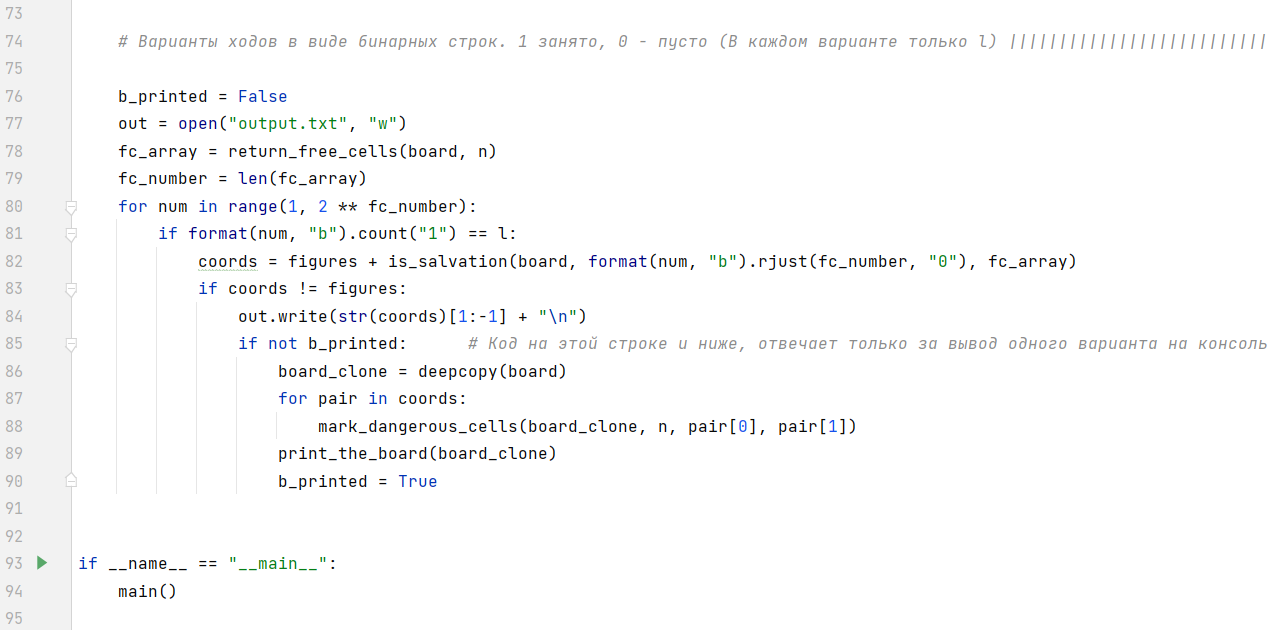
Создаётся переменная-флажок b\_printed, она пригодится для дальнейшего вывода в консоль. Открывается выводной файл. Создаются список координат свободных клеток fc\_array и целое число fc\_number.



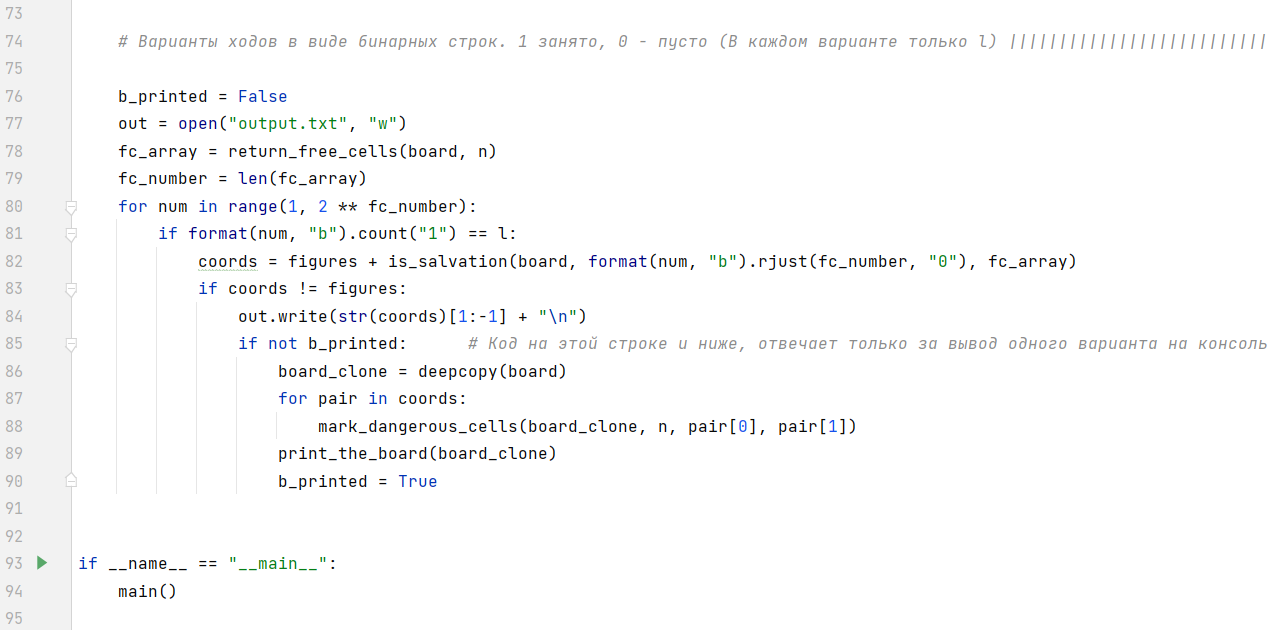
Цикл for перебирает все числа от одного до 2 в степени fc\_number для того, чтобы перебрались бинарные строки со всеми возможными вариантами расположения единиц. Специальное условие проверяет, что в строке L единиц. В противном случае решение уже неверно, и программа переходит к следующей, не тратя лишнего времени.



Если условие соблюдено, создаётся список coords из координат существующих фигур (figures) и результата описанной выше функции is\_salvation(). Если функция вернула пустой список, то есть проверяемая строка не является решением, тогда условие if coords != figures нарушено. Если условие пройдено, в выводной файл записываются координаты всех фигур решения (координаты поставленных на вводе фигур + поставленных сейчас).



Далее идёт код, которой выводит на консоль только одно верное решение. В конце первого же входа флажок b\_printed меняет значение на True, не позволяя напечатать доску когда-либо ещё. Печать осуществляется через функцию print\_the\_board(). Она используется только в этой части кода, но для удобства была вынесена в отдельную функцию.



**Результат работы**:

Программа работает исправно, однако по ходу написания выяснилось, что время её работы напрямую зависит от количества свободных клеток, оставшихся после поставленных изначально фигур (fc\_number) и количества фигур, которые ещё нужно поставить (L). Оптимизация программы лишь частично помогла уменьшить время, требуемое на выполнение.

**Вывод**:

В программе были использованы знания по всем аспектам программирования, обозначенным в целях, соответственно цели лабораторной работы выполнены.

**Листинг:**

