**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Санкт-Петербургский Государственный Морской Технический университет**

**“Кафедра Киберфизических Систем”**

Лабораторная работа № 2

по дисциплине: « Программирование»

на тему:

«Работа с текстовыми файлами, нисходящими методами разработки и локально-глобальными функциями»

Выполнил:

студент 1 курса 20121 группы

Соболев Артём Дмитриевич

Преподаватель

Преподаватель по программированию

Поделенюк П.П

Санкт-Петербург 2023

|  |  |  |
| --- | --- | --- |
| **Содержание**  Оглавление  [Цели работы 3](#_Toc123838966)  [Формулировка задания 3](#_Toc123838967)  [Ход работы 3](#_Toc123838968)  [Результаты работы 4](#_Toc123838969)  [Вывод 6](#_Toc123838970)  [Листинг кода 7](#_Toc123838971) |  |  |
|  |  |  |

# Цели работы

При работе над данной лабораторной работой будут затронуты следующие вопросы:

1. Работа с текстовыми файлами,
2. Работа с локальными и глобальными переменными,
3. Работа с несколькими функциями, а не только с main,
4. Нисходящий метод разработки программ,
5. Анализ ходов.

# Формулировка задания

Дана квадратная шахматная доска размером N x N. На доске уже размещено K фигур. Фигуры размещены так, что находятся не под боем друг друга.

Необходимо расставить на доске еще L фигур так, чтобы никакая из фигур на доске не находилась под боем любой другой фигуры. **Необходимо найти все возможные решения.**

Вы работаете с фигурой: ( Конь-дракон Объединяет в себе возможности короля и слона; Фигура ходит как конь-дракон, но только на три поля по диагонали.)

Входные данные в файле input.txt. На первой строке файла записаны три числа: N L K (через пробел). Далее следует K строк, содержащих числа x и y (через пробел) - координаты уже стоящей на доске фигуры (фигуры стоят правильно). Координаты отсчитываются от 0 до N-1. 1 <= N <= 20.

**Алгоритм оптимизирован, для работы с доской 20.**

Выходные данные в файл output.txt. **На каждое найденное решение необходимо записать в файл одну строку**. Строка состоит из пар (x,y) - координаты фигур на доске. **В решение следует вывести координаты всех фигур, находящихся на доске.** Каждую фигуру необходимо записать в виде пары координат, разделенных запятой и обрамленных скобками. Координаты отсчитываются от 0 до N-1. Порядок, в котором фигуры перечислены в решении, не имеет значения. Если не было найдено ни одного решения, в файл необходимо записать no solutions.

Выходные данные на консоль — это доска N\*N, где фигура обозначается #, ее ходы обозначаются \*, а пустые клетки обозначаются 0.

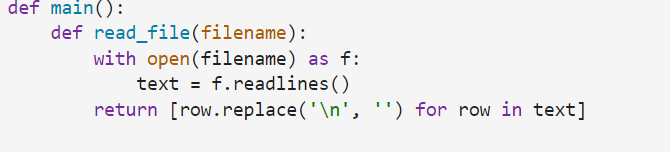
# Ход работы

Данная программа из входных данных (input.txt) получает 3 числа в первой строке (N L K) соответственно длина доски(1 сторона(n\*n)(20\*20 по заданию)), кол-во фигур, которые нужно расставить и кол-во фигур, которые уже стоят на доске. При этом нужно не допустить бои фигур друг на друга. Нужно найти всевозможные решения.

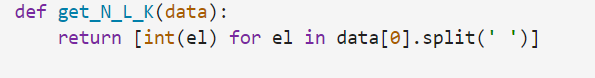
Для моей фигуры № 12 Конь-Дракон+ (отличие от обычного только в ограничении ходов в 3 клетки по диагонали) на самом деле прописать координаты x и y не стоило труда. Изначально нужно было прописать чтение входящего файла и чтение (NLK). Далее пришлось заводить функции для: существующих координат, генератор свободных координат, генератор возможных координат(через роспись всевозможных действий с x и y), генератор возможных координат( учитывая свободные и возможные), уход от дублирования существующих координат, ходы(для фигур на расставление), получение решений(на основе действий с генераторами), выводящий файл и вывод в консоль.

В конце кода собственно происходит работа с вводимым файлом и осуществляется сортировка и удаление дубликатов для вывода в (output.txt)

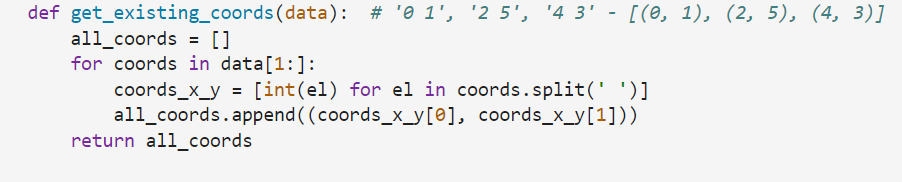
# Результаты работы



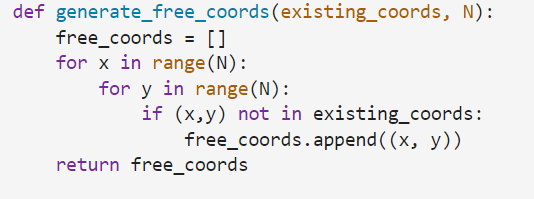
Как уже говорилось это функция нужна для чтения файла и переноса чтения на следующую строку.



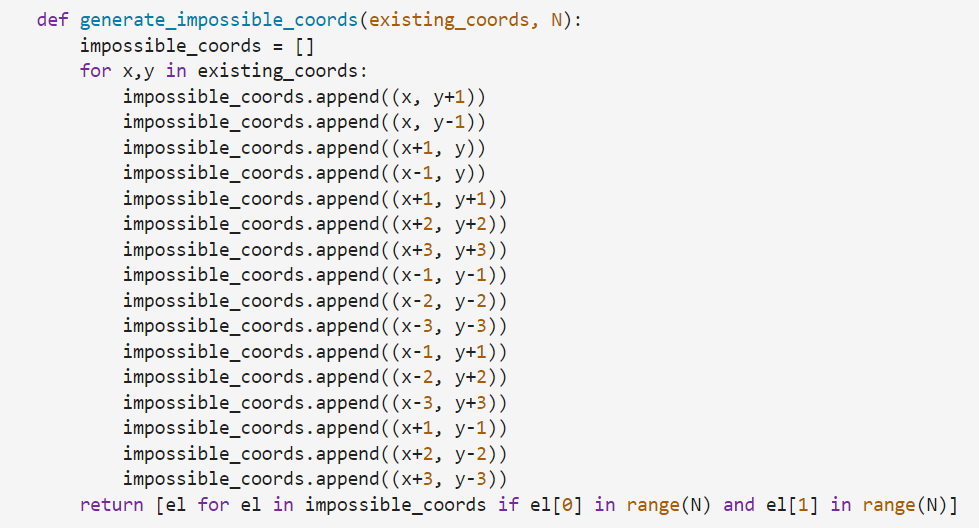
Определение (NLK) и возврат целых значений элемента с разделением пробелом.



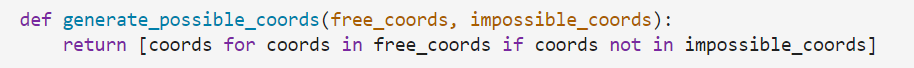
Определение существующих координат из (NLK) с разделением координат на x и y.



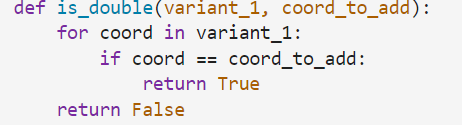
Генерация свободных координат учитывая существующие координаты и N, проверка координат на их первенство в использовании.



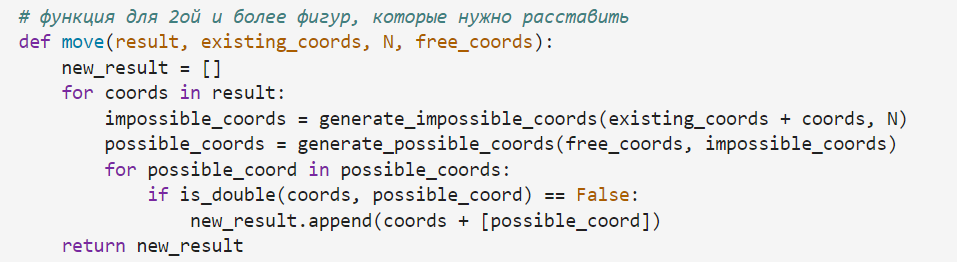
Генерация невозможных координат , и составление всевозможных существующих координат ( по фигуре № 12), возврат элемента в невозможных координатах.



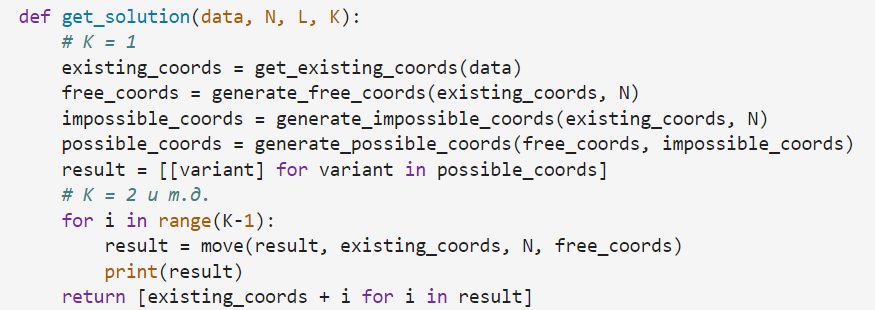
Генерация возможных координат при учете свободных и невозможных и возврат свободных координат если их нет в невозможных.



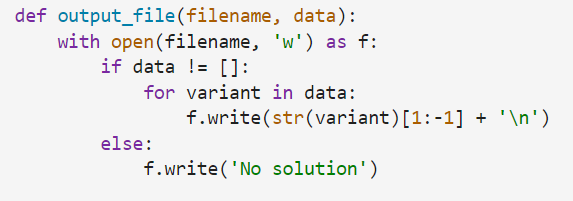
Функция проверяет координаты на дубликаты.



Функция , рассчитанная на вычисление ходов фигур , которые можно расставить путем арифметических вычислений результата , существующих координат, свободных координат и N. В итоге происходит возврат новых координат.

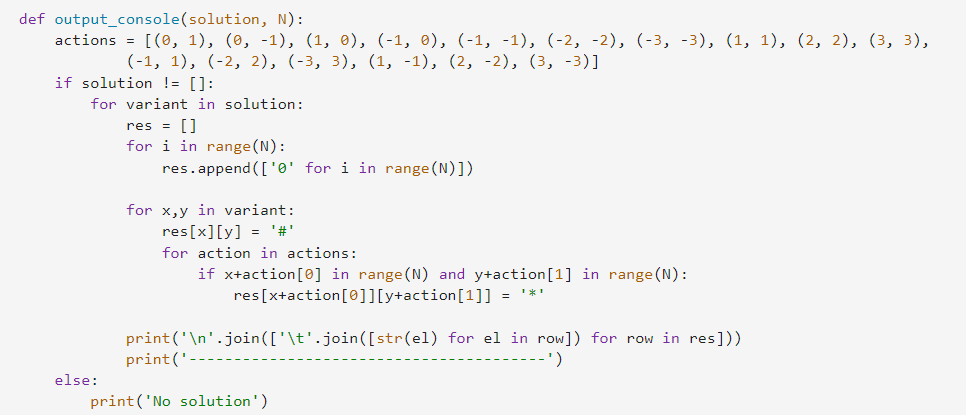


Функция , которая получает решение и выводит его.

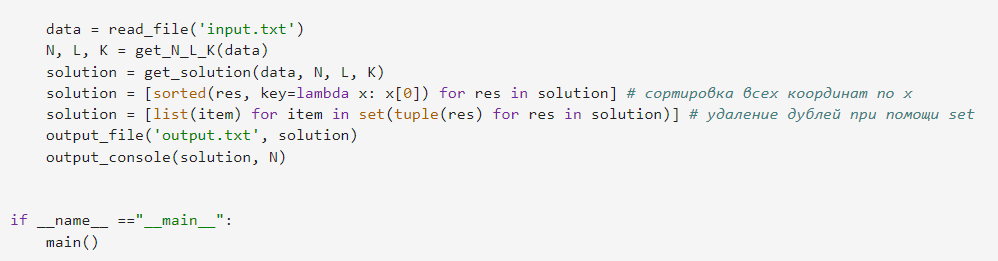


Функция для записи решения (с переносом на след. строку) в output.txt.

Если в data ничего нет, будет введено “No solution”.



Здесь происходит вывод местонахождения фигур и. т. д в КОНСОЛЬ , проверка на “пустоту” аналогична.



Здесь происходит чтение файла, сортировка по x, и удаление дубликатов. Общий результат записывается в output.txt.

# Вывод

Программа ,получающая на вход координаты и кол-во клеток доски, а также кол-во фигур уже стоящих и требуемых на расстановку, выполняет условия и выводит в выводящий файл, консоль и итоговое окно интерпретатора всевозможные решения. При высчитывании ходов может возникать ошибка, но она со временем исчезает и выводится полный ответ .

# Листинг кода





