**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Санкт-Петербургский государственный морской технический университет»

(СПбГМТУ)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет цифровых промышленных технологий

Направление подготовки 09.03.01.03 "Интеллектуальные технологии

киберфизических систем"

Лабораторная работа №2

по дисциплине “Программирование”

Студент 1 курса группы 20121

Очного отделения

Хохлов Д.Р.

Проверил:

Поделенюк П. П.

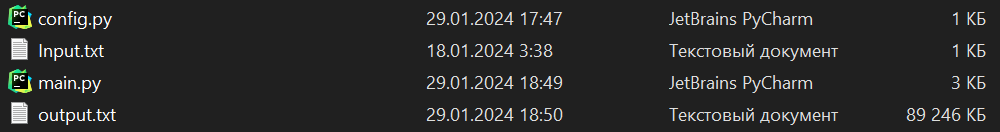
2023

Цель работы - заключается в разработке программы на языке программирования Python, решающей задачу размещения шахматных фигур (в моём случаем визирей) на доске размером NxN таким образом, чтобы они не находились под угрозой боя друг с другом.

Ход работы

Работа имеет 4 рабочих файла:

* main.py - основные принципы работы записаны в этом файле.
* config.py - в этом файле записаны функции для получения нужных данных (get)
* input.txt - тут записаны данные характеристик (например N - размер шахматной доски)
* output.txt - сюда будут записываться результаты вычислений



Работа алгоритма состоит из нескольких этапов:

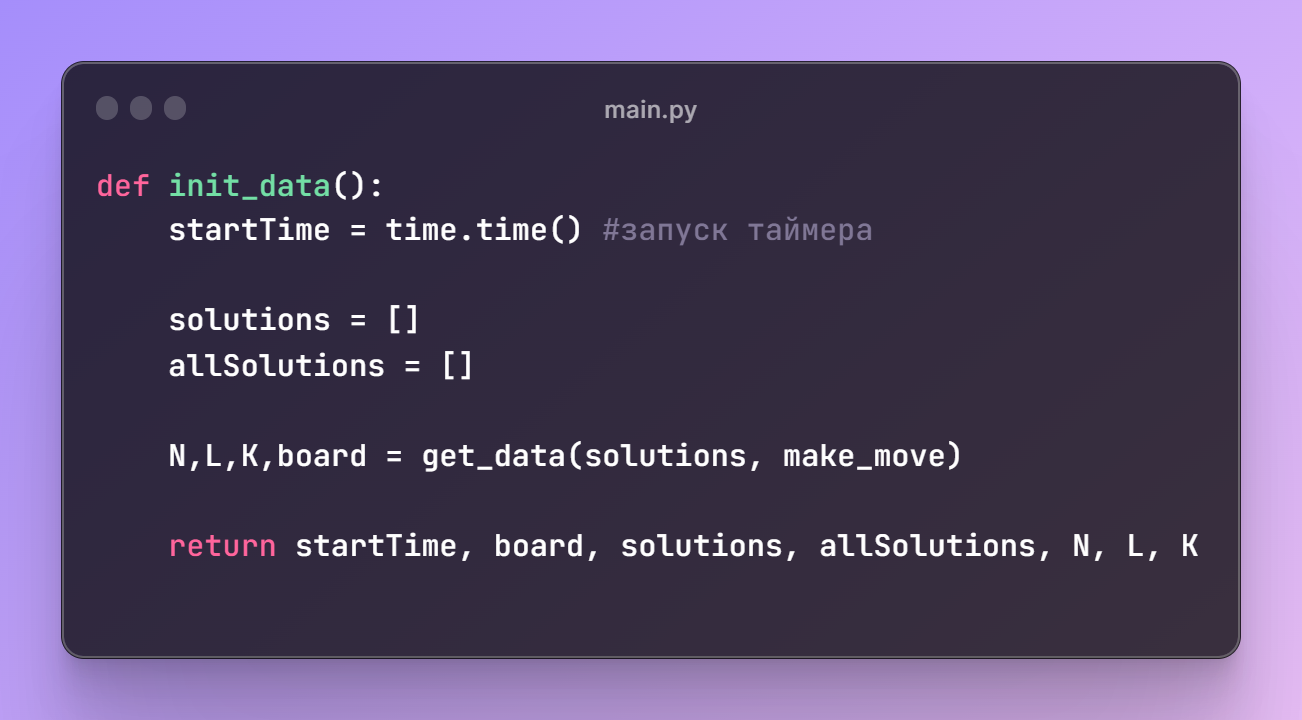
* Чтение данных
* Запуск рекурсивной функции
* Запись результатов и их вывод

Разберем каждый этап по-подробней

**Чтение данных:**

* импортируем файл *“config.py”* в главный файл *“main.py”*
* запускаем функцию “init\_data()” инициализируем некоторые переменные
* запускаем функцию “get\_data()”(она считывает данные из *“input.txt”*) из файла *“config.py”* и записываем в переменные N,L,K, board







**Запуск рекурсии:**

* Программа использует рекурсивную функцию “find\_solutions” в файле *“main.py”*, которая принимает текущую доску, количество оставшихся ферзей для размещения, текущую позицию (строку и столбец) и текущий список решений.
* В каждом шаге рекурсии программа проверяет возможность размещения ферзя в текущей позиции. Если ферзь может быть размещен, то он добавляется на доску, и соответствующие клетки отмечаются, чтобы исключить возможность размещения других ферзей, которые могли бы быть угрозой.
* Рекурсия продолжается для следующей фигуры и так далее.
* Если все фигуры успешно размещены, программа добавляет текущее решение в список найденных решений.
* Если не удается разместить фигуру в текущей позиции, программа откатывается назад и пытается другие варианты размещения.



Запись результатов и их вывод:

* После завершения выполнения алгоритма, программа выводит количество найденных решений и записывает их в файл "output.txt".

А также выводит время работы программы и количество решений



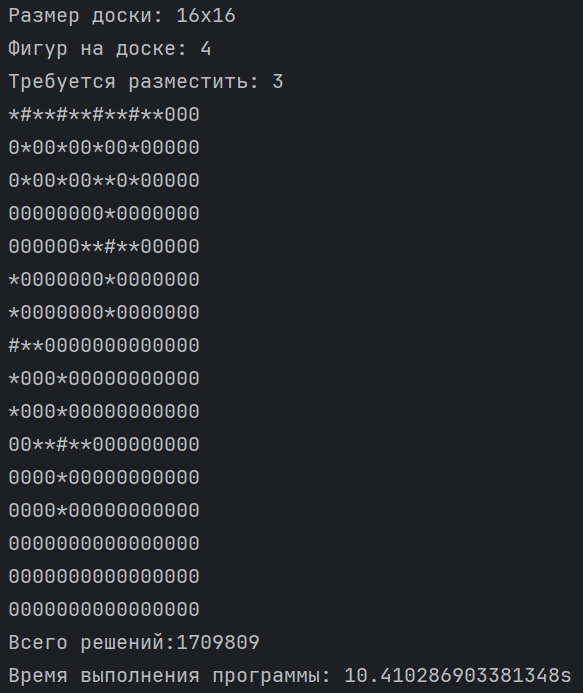
Результат работы

Введем характеристики в файл “input.txt”:

* N = 16, L = 3, K = 4
* Поставим некоторые фигуры на координаты:
  + (0,1)
  + (10,4)
  + (4,8)
  + (7,0)



Через небольшое количество времени получаем результат:



Вывод

Была написана программа на Python,построенная на рекурсии, размещающая шахматные фигуры на правильных клетка, оптимизированная для быстрой обработки большого количества входных данных

Весь код:



