**ПНИПУ**

**Пермский Национальный Исследовательский Политехнический Университет**

**Лабораторная работа – задача о n ферзях**

*Выполнил:*

*Герасимов Максим Николаевич РИС-23- 3б*

*Проверила:*

*доцент кафедры ИТАС О.А. Полякова*

2024

**РЕШИТЬ ЗАДАЧУ О N ФЕРЗЯХ**

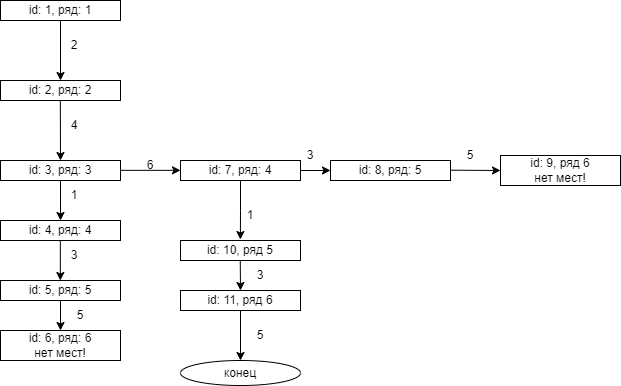
**АНАЛИЗ ЗАДАЧИ**

1. Изобразим шахматную доску в виде двумерного массива, все элементы которого могут равняться значениям в интервале [-1, n], -1 означает, что в клетке стоит ферзь, любое число выше означает, сколько ферзей бьет эту клетку (0 – ни один, 1 – один и т.д.);
2. Задачу следует выполнить через рекурсивный алгоритм. Каждую итерацию рекурсии будет начинаться цикл, в нём будет выбираться случайная свободная клетка, в которую будет поставлен ферзь;
3. Затем, в этом же цикле, будет создаваться новая итерация рекурсии, работающая по таким же принципам. Если на ряде не будет клеток, в которые можно поставить ферзя, то цикл “прервется” и мы вернемся на предыдущую итерацию рекурсии;
4. Поняв, что закончить задачу невозможно, программа выберет новое место для ферзя, а со старого его уберет, пометив его как “невалидное”. После этого цикл продолжится;
5. Если мы сможем успешно поставить ферзя в последний ряд, то мы возвращаемся на предыдущие итерации рекурсии с этой информацией, заканчивая их.

**АНАЛИЗ РАБОТЫ РЕКУРСИИ**

Разберем пример работы рекурсии в нашей программе на примере доски 6x6, на которую надо поставить 6 ферзей;

Схематично процесс будет выглядеть так:



Блоки обозначают собой итерации рекурсии, каждой присвоен id номер, идущий строго по возрастанию (id: 1 – первая итерация, id: 11 – последняя). Цифры около стрелок обозначают место, на которое было решено поставить ферзя.

Как можно увидеть, программа успешно расставляла ферзей вплоть до 6-ого ряда, в котором не было мест. Программа вернулась на 5 ряд в итерацию с id = 5, чтобы поменять клетку ферзя там. Однако, в 5 ряду не было никаких свободных мест, кроме уже занятого. Программа вернулась на 4 ряд, в итерацию с id = 4, где наткнулась на туже самую проблему, что и в 5-ом ряду. Переставить ферзя можно только в 3-ем ряду, что программа и сделал, переставив ферзя в ряде 3 с 1-ой клетки на 6-ую.

После этого программа поставила ферзей в 3-ей клетке 4-ого ряда и в 5-ой клетке 5-ого ряда, вновь не оставив места для ферзя в 6-ом ряду.

В этот раз ближайшим рядом, в котором можно поменять ферзя, является 4-ый, программа переставила в нем фигуру с 3-ей клетки на 1-ую.

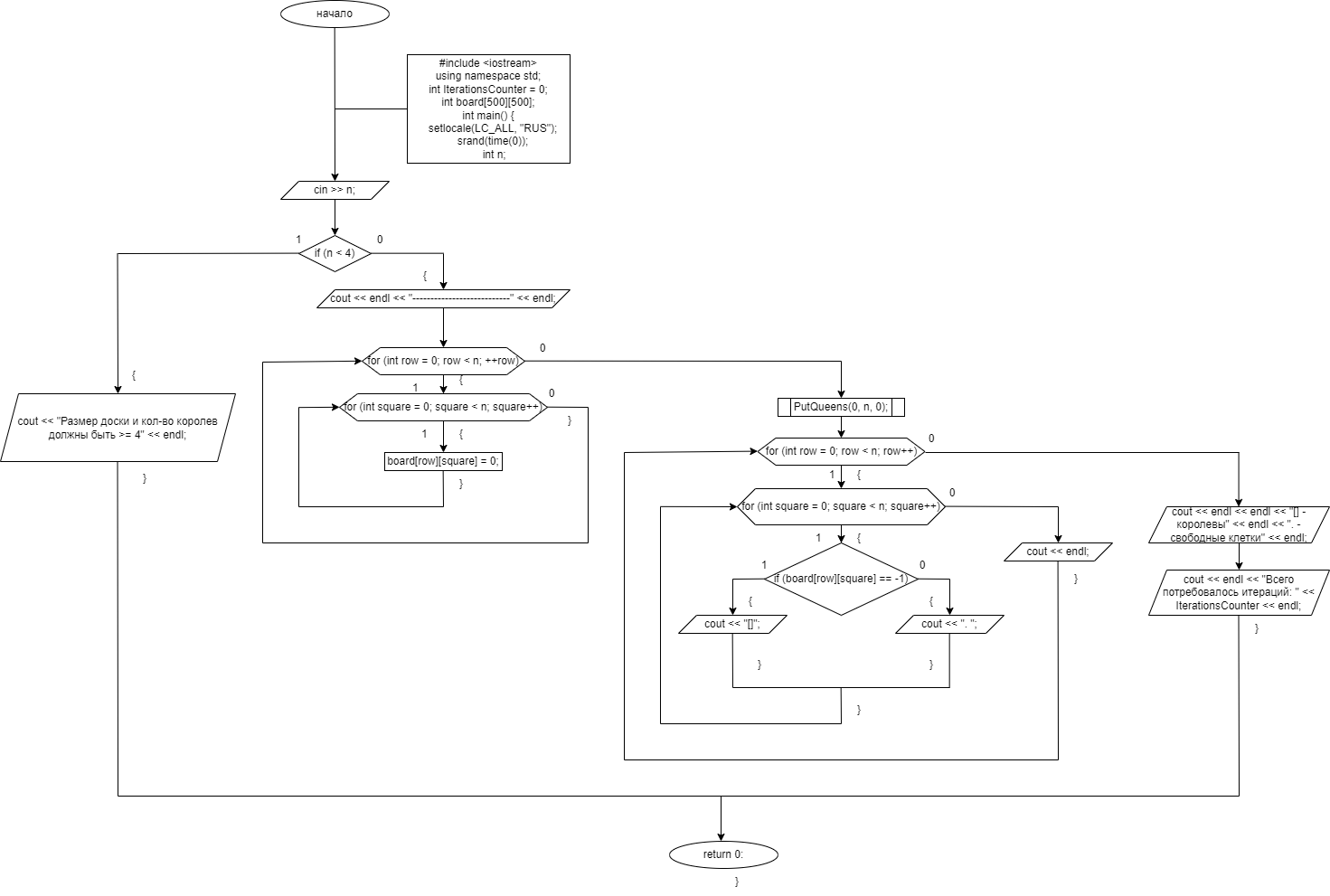
Далее программа поставила ферзей в 3-ей клетке 5-ого ряда и в 5-ой клетке 6-ого ряда, тем самым успешно решив задачу!

От каждой итерации рекурсии может отойти от 1 до n веток включительно, каждая из которых обозначает собой клетку в ряду. Так как в нашей задаче n = 6, то от каждой ветки рекурсии может отойти от 1 до 6 веток включительно.

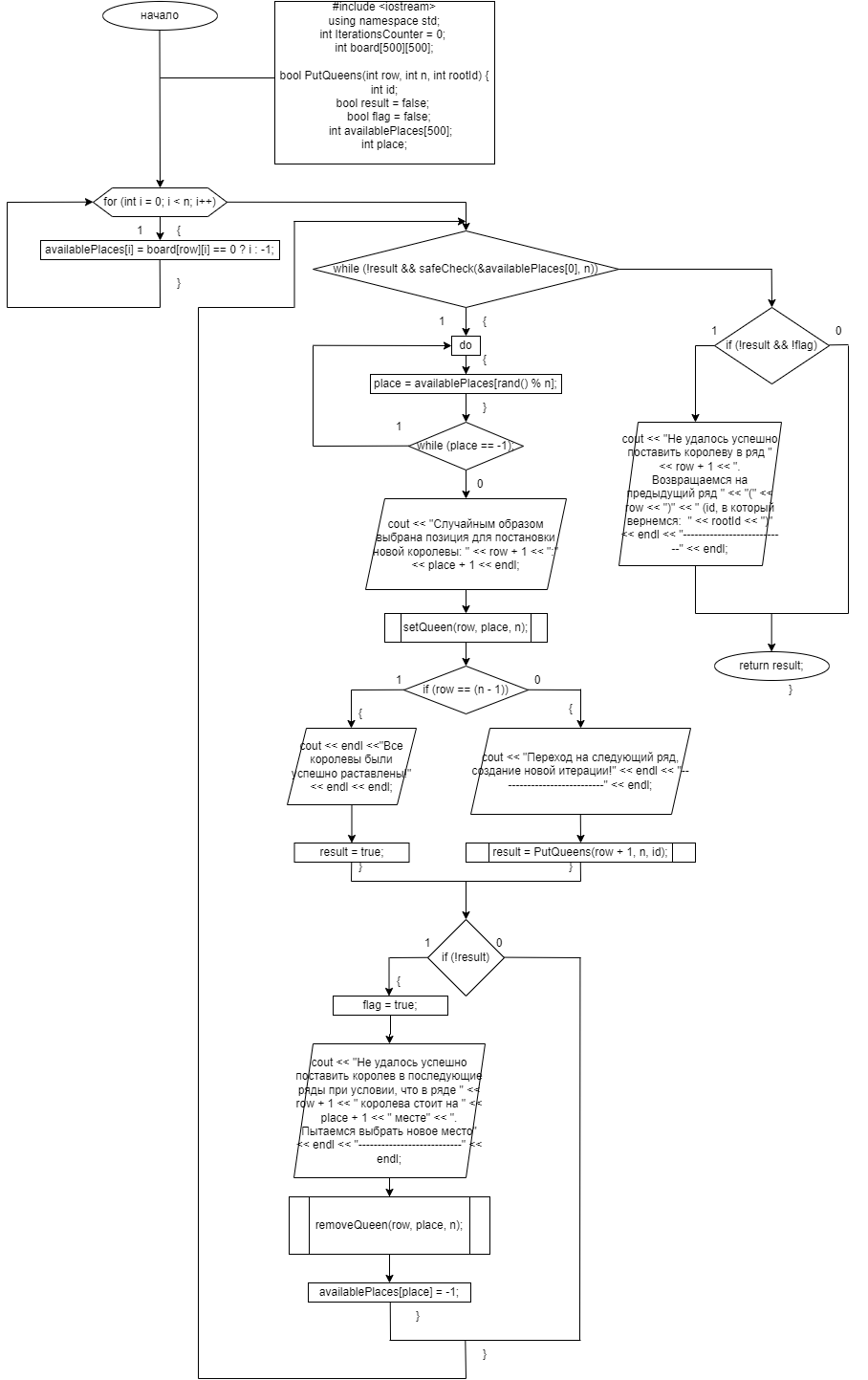


**БЛОК СХЕМЫ**

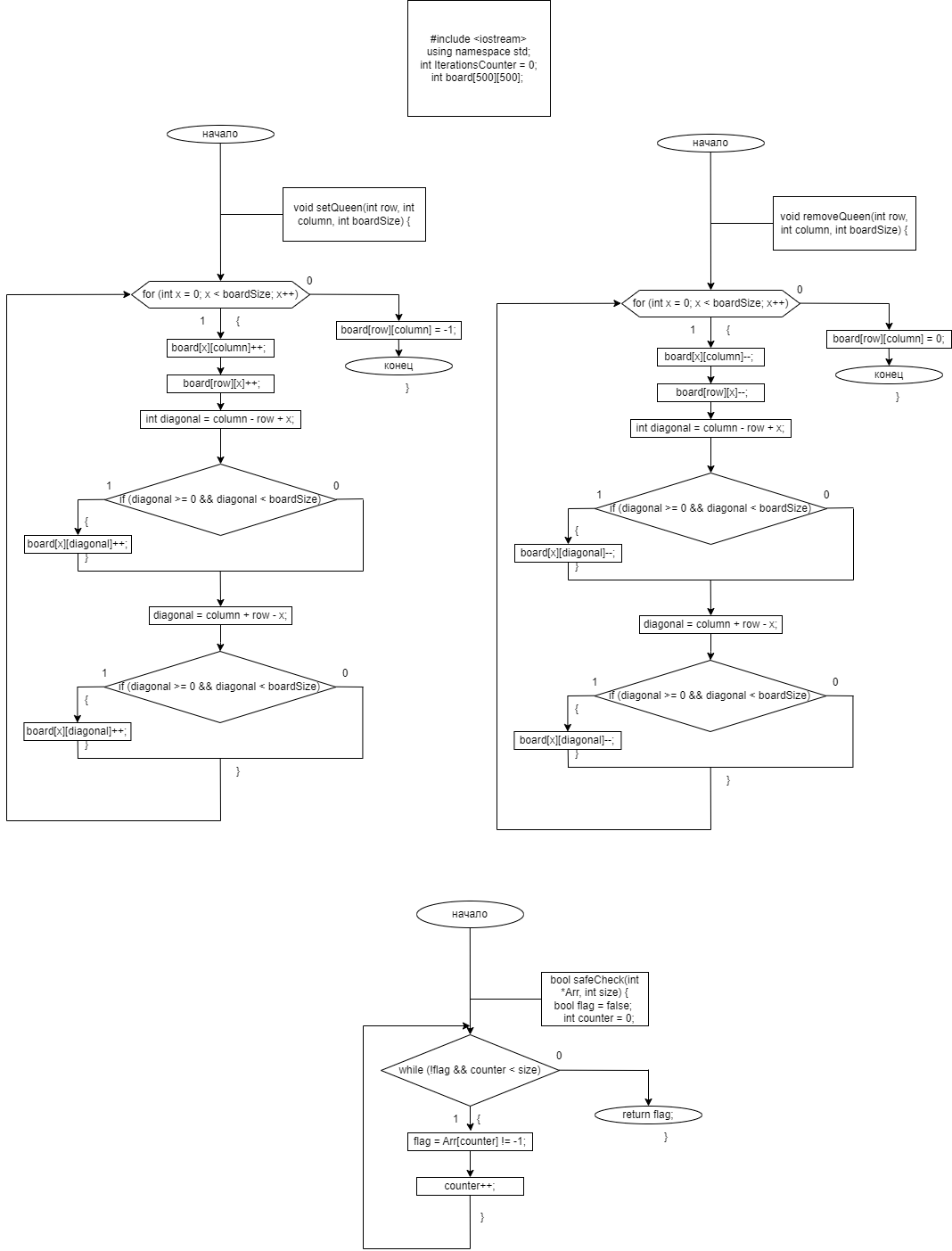
**Блок main:**



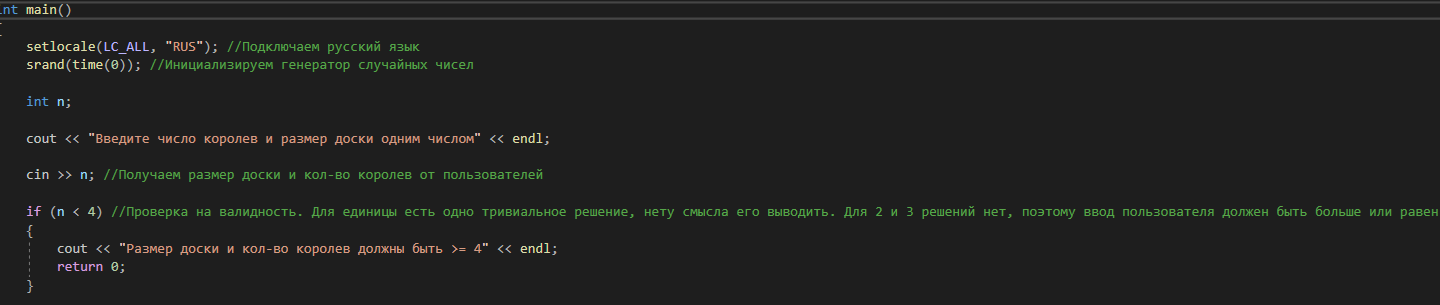
**Рекурсивная функция:**

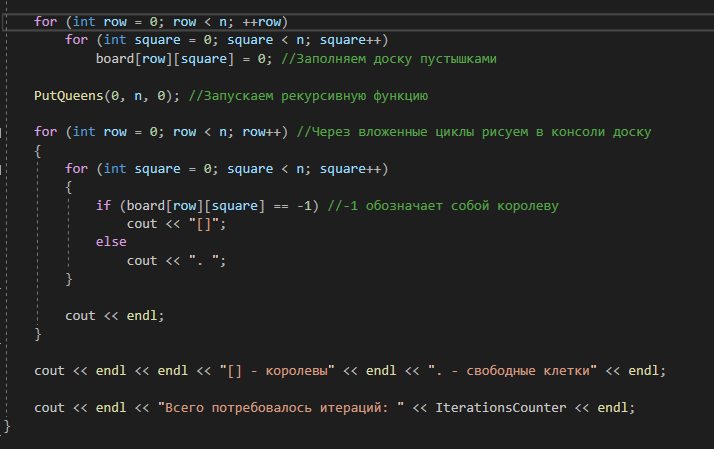


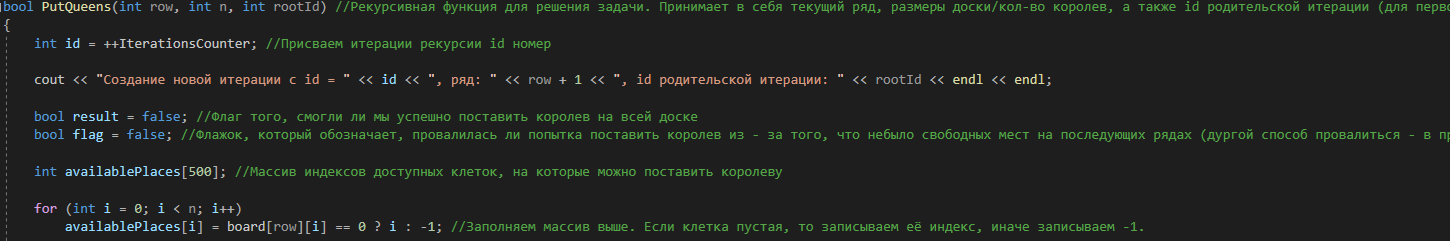
**Вспомогательные функции:**

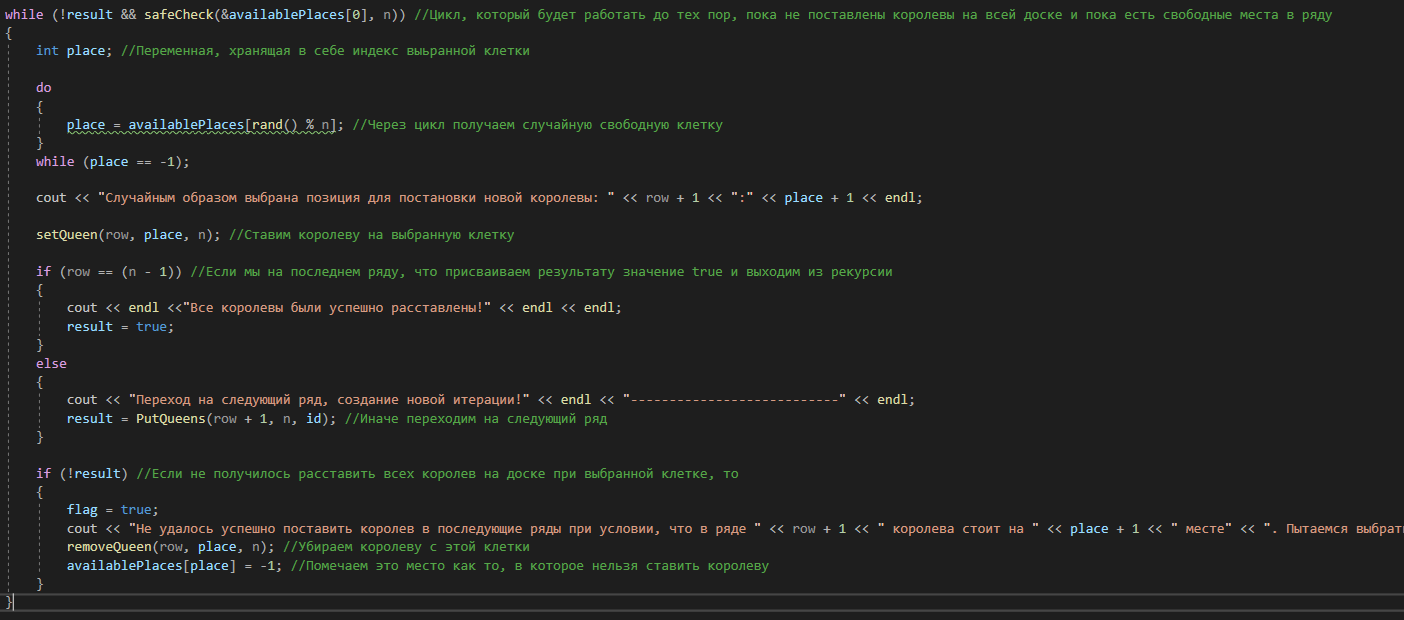


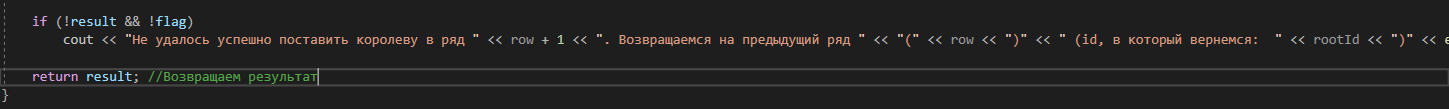
**КОД С КОМЕНТАРИЯМИ**

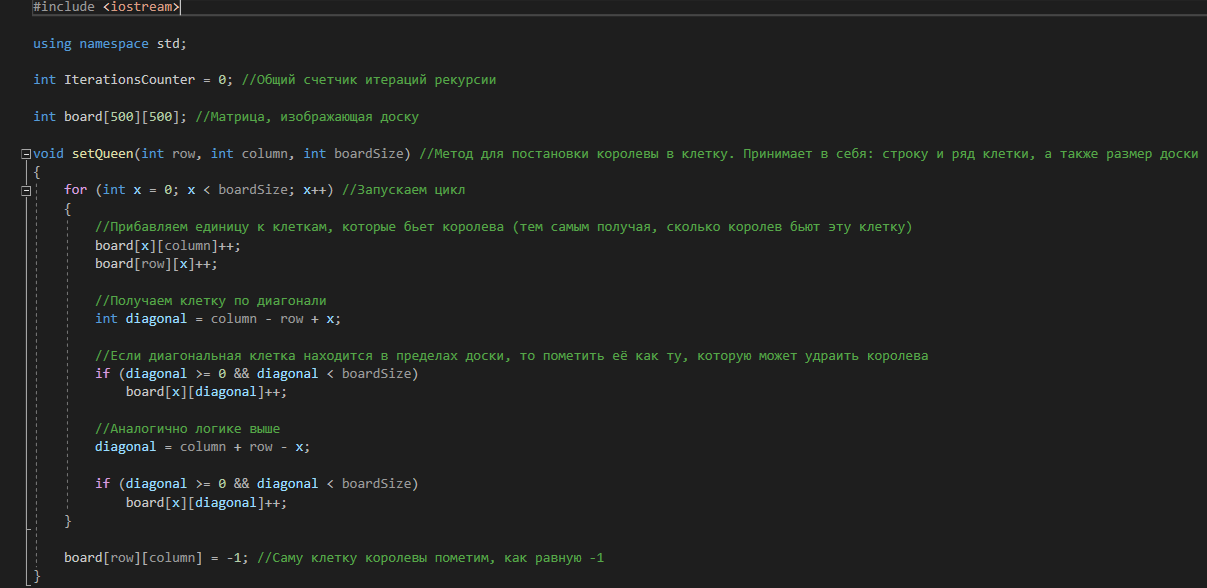
****

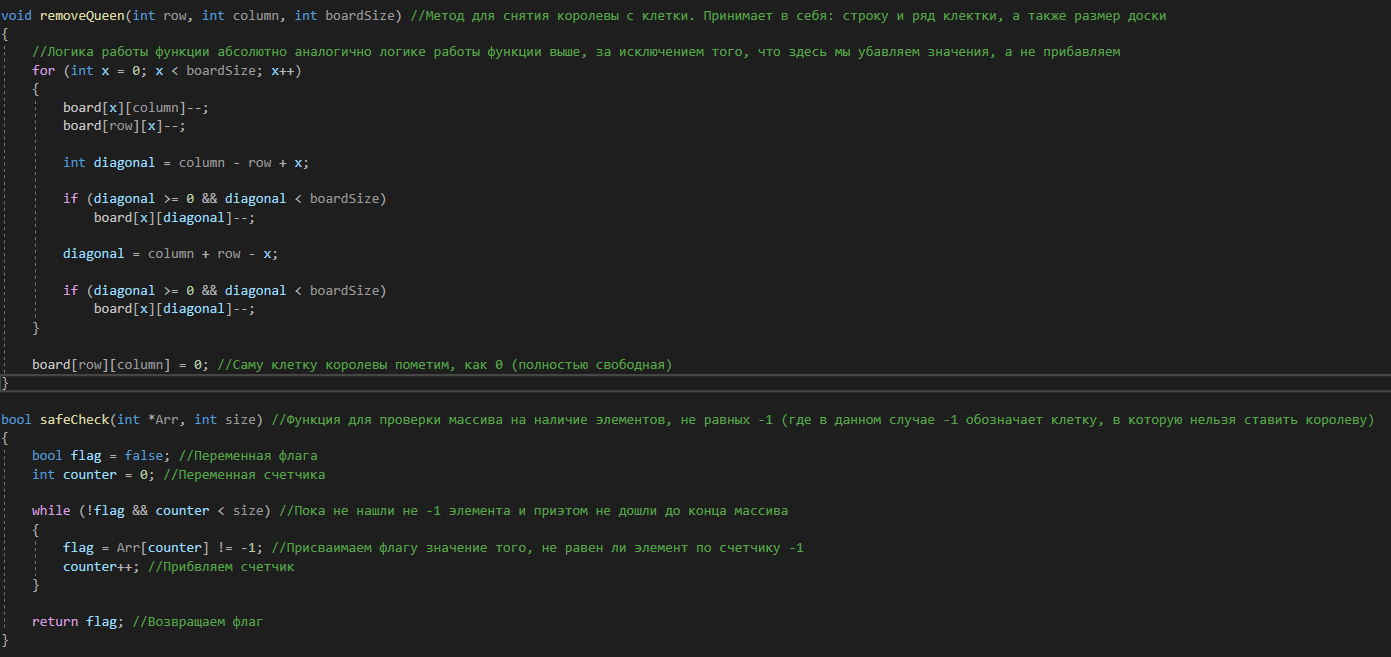
****

****

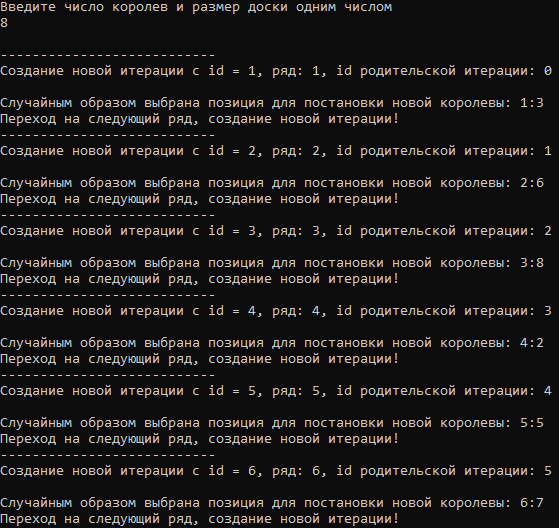
****

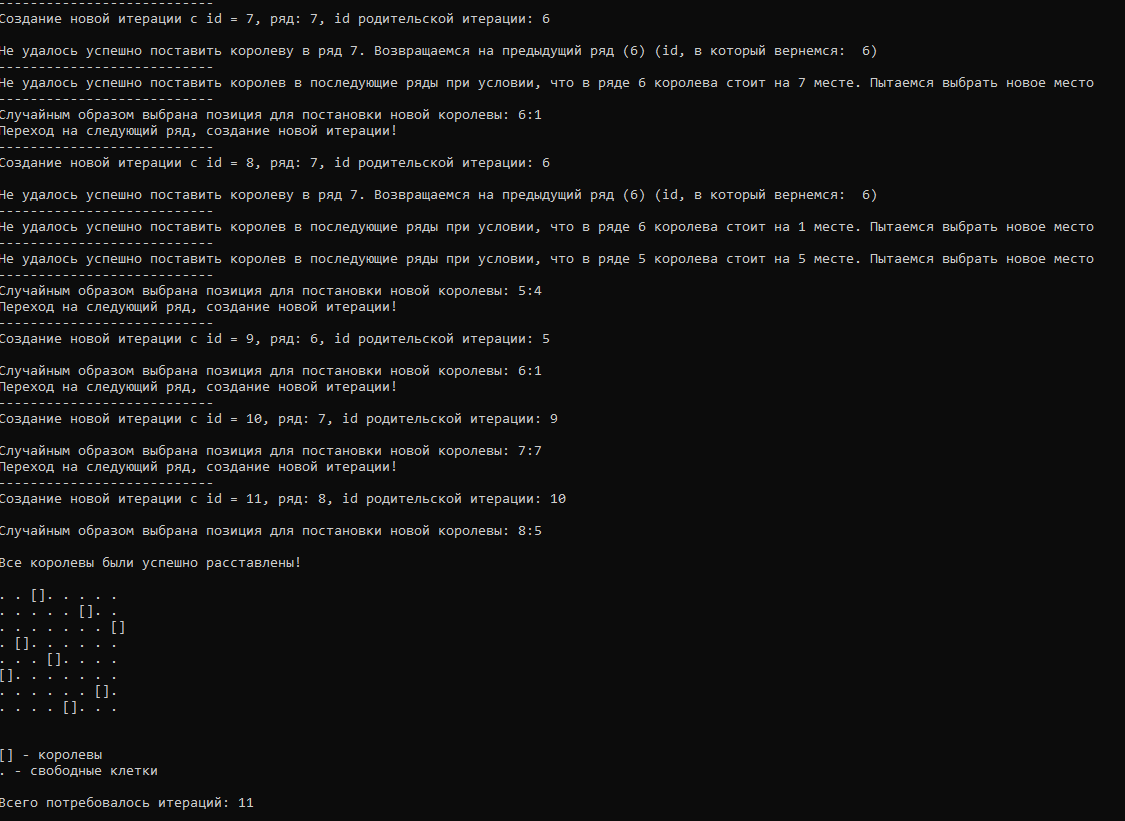
****

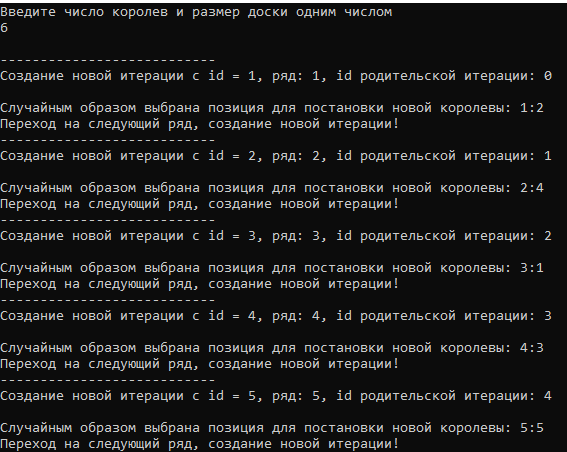
****

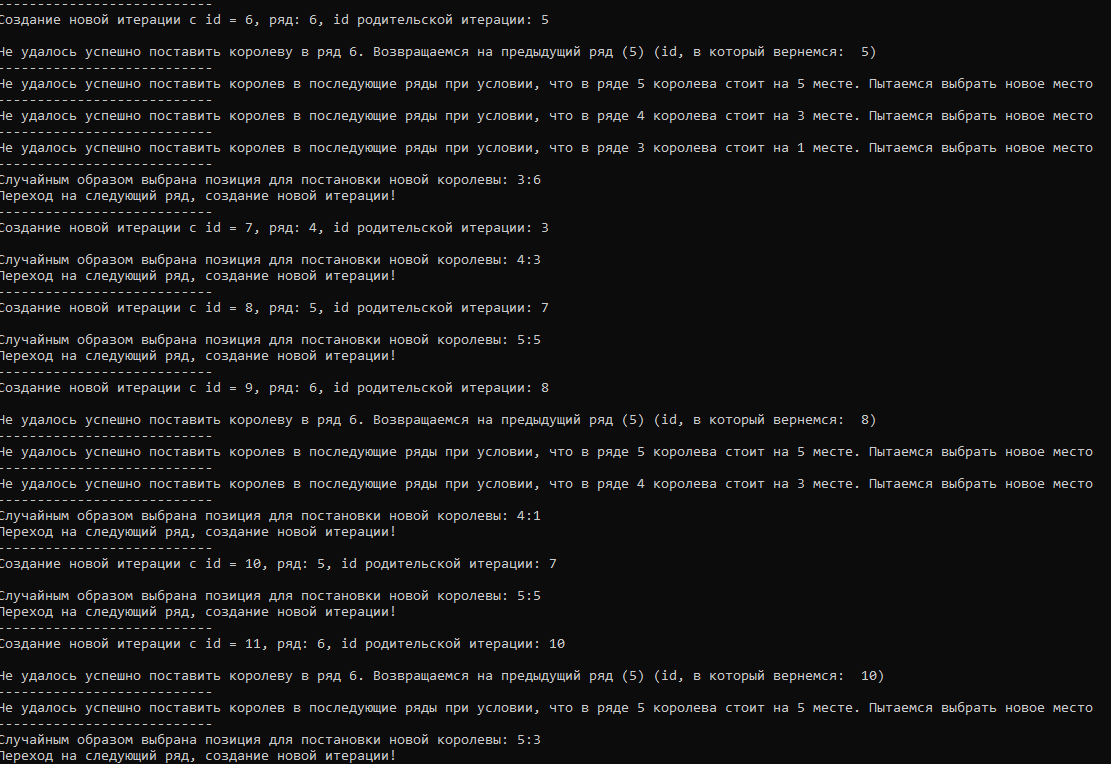
****

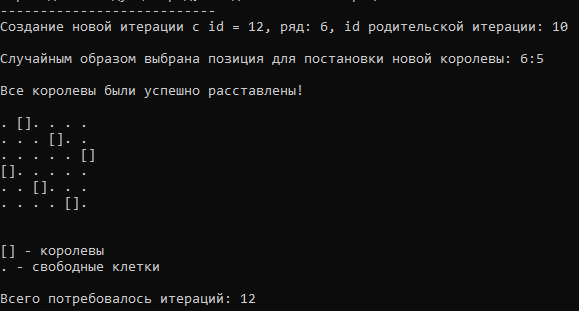
**РЕЗУЛЬТАТЫ РАБОТЫ**











**ВЫВОДЫ**

Данная задача интересна тем, что в ней итерации рекурсии не имеют строго заданного количества веток. Как ранее говорилось, итерация рекурсии может иметь от 1 до n веток, отходящих от неё в другие итерации.

**ГИТХАБ**

[Ссылка на репозиторий](https://github.com/BlackSerperior6/LabWorkQueens)

