Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**ЛАБОРАТОРНАЯ РАБОТА**

«Задача расстановки ферзей»

**Выполнил:**

студент группы РИС-23-2б

Борисов Никита Андреевич

**Проверила:**

доцент кафедры ИТАС

О.А. Полякова

Пермь, 2024 г.

**Задача:**

Написать программу, в которой реализуется алгоритм расстановки ферзей на привычной шахматной доске размерностью 8 на 8. Главное условие создания программы − работа с помощью рекурсии. Сложность поставленной задачи заключается в том, чтобы ферзи были расставлены в каждой строке и не могли срубить друг друга.

**Анализ:**

Ферзь может ходить в любом направлении по горизонтали, вертикали, диагонали и на любое количество клеток. Чтобы ферзи не могли срубить друг друга, нужно придерживаться принципа расстановки:

1. Поставить соответствующего по счёту ферзя на соответствующую ему строку в первую возможную клетку.
2. Перейти на следующую строку и поставить следующего ферзя так, чтобы предыдущий ферзь не мог его срубить.

* Если на какой-либо строке поставить ферзя невозможно (т.е. нет такой клетки в строке, чтобы другие ферзи не могли срубить текущего ферзя), то следует возвратиться на предыдущую строку и поставить ферзя в следующую возможную клетку строки, если это возможно. Если нет, то уходим ещё на уровень назад в дереве рекурсии.

1. В конечном счёте надо расставить все 8 ферзей на 8 соответствующих им строк.

**Решение:**

1. Необходимо реализовать две функции: поставить ферзя и убрать ферзя. Первая будет ставить в заданную клетку ферзя и отмечать все клетки, которые находятся под боем, т.е. те, в которые ставить следующего ферзя нельзя. Вторая, соответственно, будет убирать ферзя и клетки шахматной доски, находящиеся под боем.
2. Помимо этого, должна быть функция, которая выбирает, куда поставить ферзей, она поочередно проходит строки и ставит ферзей так, чтобы они находились не под боем *(ферзи не должны находиться в одной строке, в одном столбце и на одной диагонали)*. Но если на какой-то строке функция уже не может поставить ферзя, то она возвращается на шаг назад, убирает предыдущего ферзя и пытается поставить его на другое место. И так до тех пор, пока все ферзи не займут свои места.

Изначально определен двумерный массив board, который обозначает шахматную доску 8 на 8.

В главной функции значение всех элементов двумерного массива приравнивается к 0. Вызывается функция tryQueen, в качестве параметра передается значение 0 т.е. первая строка шахматной доски.

Для дальнейших функций также можно применить способ отображения состояния на доске showBoard.

*Функция проверки возможности постановки ферзя* − tryQueen проверяет, можно ли поставить ферзя на данную позицию.

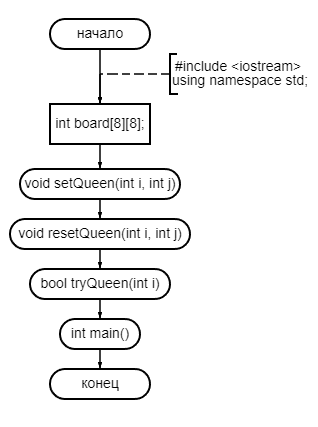
В цикле for проверяются все элементы строки; если очередной элемент равен 0, на данную позицию ставится ферзь. Далее осуществляется переход на следующую строчку. Если в какой-либо из последующих строк ни в один столбец нельзя поставить ферзя, осуществляется переход на предыдущий шаг, и ферзь ставится уже на следующую допустимую позицию.

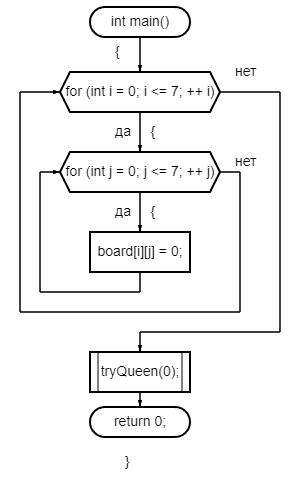
Функция установки ферзя − setQueen ставит на позицию [i][j] ферзя и отмечает те позиции, которые данный ферзь бьёт.

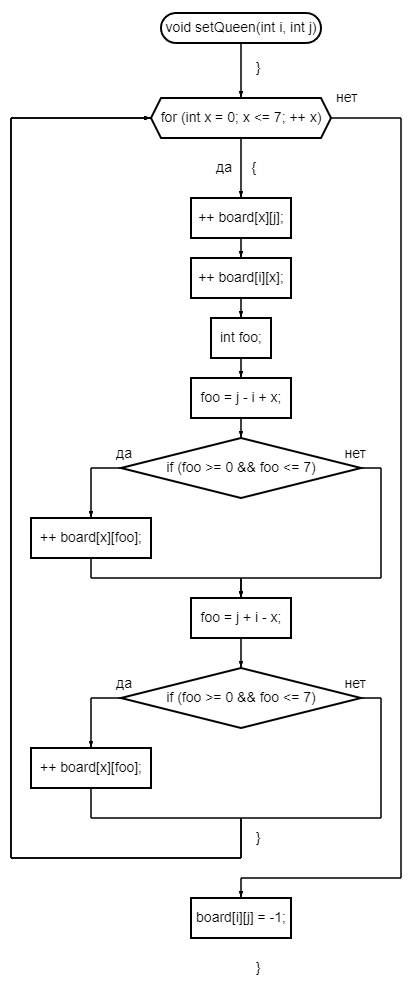
Чтобы поставить ферзя, сначала надо отметить позиции, которые данный ферзь бьёт, т.е. прибавить 1 к значениям элементов, которые находятся под «боем», а затем значение в клетке с ферзём заменить на -1.

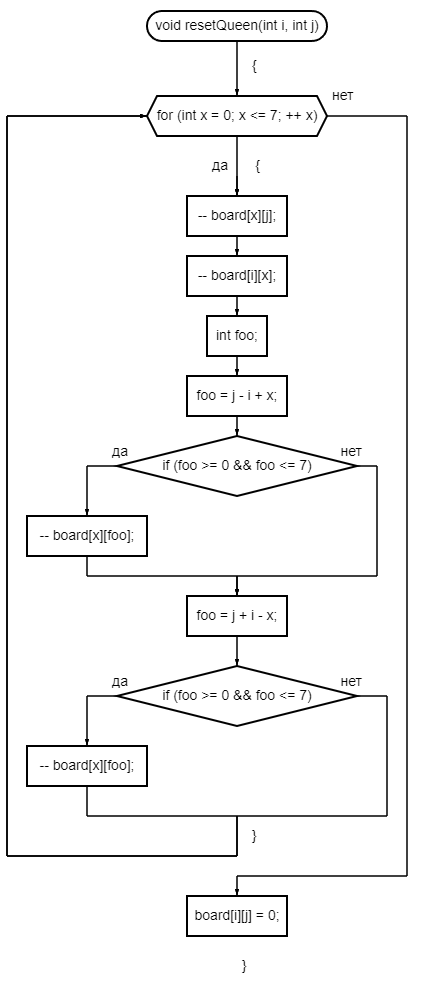
Функция отмены постановки ферзя − resetQueen убирает с позиции [i][j] ферзя и убирает отметки с тех позиций, которые данный ферзь «бил».

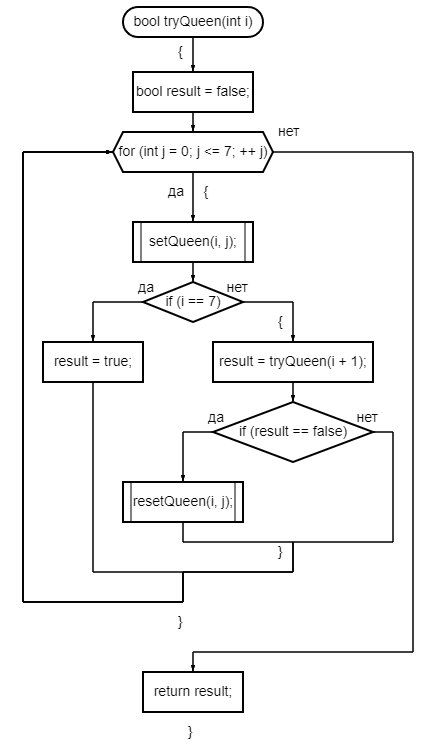
**Моделирование:**





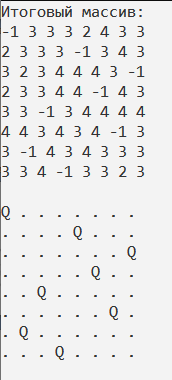


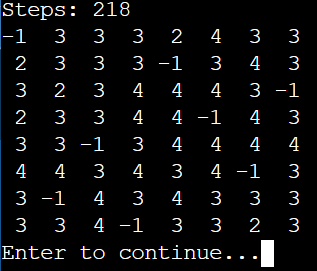




Визуализацию дерева рекурсии для данной задачи можно посмотреть в файле the\_recursion\_tree\_of\_the\_eight\_queens\_arrangement.jpg, а также ознакомиться с презентацией на данную тему Задача о восьми ферзях.pptx.

**Результаты работы программы:**





Более подробные результаты выведены в файл visualization\_of\_the\_solution\_to\_the\_problem\_of\_placing\_eight\_queens.txt

**Код программы:**

#include <iostream>

using namespace std;

int board[8][8]; // Шахматная доска (размерность 8x8)

void showBoard() { // Функция для визуализации изменений на доске

for (int i = 0; i <= 7; ++ i) {

for (int j = 0; j <= 7; ++ j) {

if (board[i][j] >= 0) cout << " " << board[i][j] << " ";

else cout << board[i][j] << " ";

}

cout << endl;

}

// Для удобства просмотра в консоли по шагам

cout << "Enter to continue...";

string pause;

getline(cin, pause);

}

void setQueen(int i, int j) { // Функция принимает на вход ту клетку, куда можно ставить ферзя

for (int x = 0; x <= 7; ++ x) { // значения x перебираюся 8 раз потому что этого достаточно, чтобы пройти по всей доске в любом направлении

// Действия в цикле служат для обозначения клеток под боем по разным направлениям

// Во все клетки под боем надо добавить 1

++ board[x][j]; // По столбцу сверху вниз

++ board[i][x]; // По строке слева вправо

int foo; // Задаёт направление двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя

foo = j - i + x;

if (foo >= 0 && foo <= 7) ++ board[x][foo]; // Сверху вниз по диагонали, напрвленной вправо

foo = j + i - x;

if (foo >= 0 && foo <= 7) ++ board[x][foo]; // Сверху вниз по диагонали, направленной влево

}

board[i][j] = -1; // Ферзь поставлен

showBoard();

}

void resetQueen(int i, int j) { // На вход функции принимается клетка, в которой на данный момент стоит ферзь

for (int x = 0; x <= 7; ++ x) {

// Цикл убирает все клетки под боем ферзя по разным направлениям

// Из всех таких клеток вычитается 1

-- board[x][j]; // По столбцу сверху вниз

-- board[i][x]; // По строке слева вправо

int foo; // Задаёт направление двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя

foo = j - i + x;

if (foo >= 0 && foo <= 7) -- board[x][foo]; // Сверху вниз по диагонали, напрвленной вправо

foo = j + i - x;

if (foo >= 0 && foo <= 7) -- board[x][foo]; // Сверху вниз по диагонали, направленной влево

}

board[i][j] = 0; // Ферзь был убран

showBoard();

}

bool tryQueen(int i) { // Функция принимает на вход индекс строки

bool result = false; // Флажок, знаменующий конец расстановки

for (int j = 0; j <= 7; ++ j) { // Осуществляется проход по всем клеткам строки

if (board[i][j] == 0) { // Проверка на пустую клетку (если -1, то в ней ферзь, если > 0, то под боем)

setQueen(i, j); // Установка ферзя в пустую клетку

if (i == 7) result = true; // Когда индекс строки будет равен 7, это значит, что все ферзи были успешно расставлены

else {

result = tryQueen(i + 1); // Показатель возможности постановки следующего ферзя

if (result == false) resetQueen(i, j); // Если следующего ферзя нельзя поставить, меняется позиция текущего

}

}

}

return result;

}

int main() {

for (int i = 0; i <= 7; ++ i) { // Создаём пустую доску, заполненную нулями

for (int j = 0; j <= 7; ++ j) {

board[i][j] = 0;

}

}

tryQueen(0); // Начинаём расставлять ферзей

return 0;

}

**Ссылка на Git:** <https://github.com/Exateym/Study>