Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа

**“Рекурсии”**

Выполнил:

студент группы РИС-23-1б

Жуланов Никита Андреевич

Проверила:

доцент кафедры ИТАС

О. А. Полякова

2024 г.

**Разработка алгоритм****ов рекурсии**

Четвёртая часть – задача о восьми ферзях

**Постановка задачи**: реализовать программу, в которой реализуется алгоритм расстановки 8 ферзей на доске 8х8 так, чтобы ферзи были расставлены в каждой строке и не «били» друг друга.

**Анализ задачи:**

Ферзь может ходить в любом направлении по горизонтали, вертикали, диагонали и на любое количество клеток, рубит он так же, как ходит.

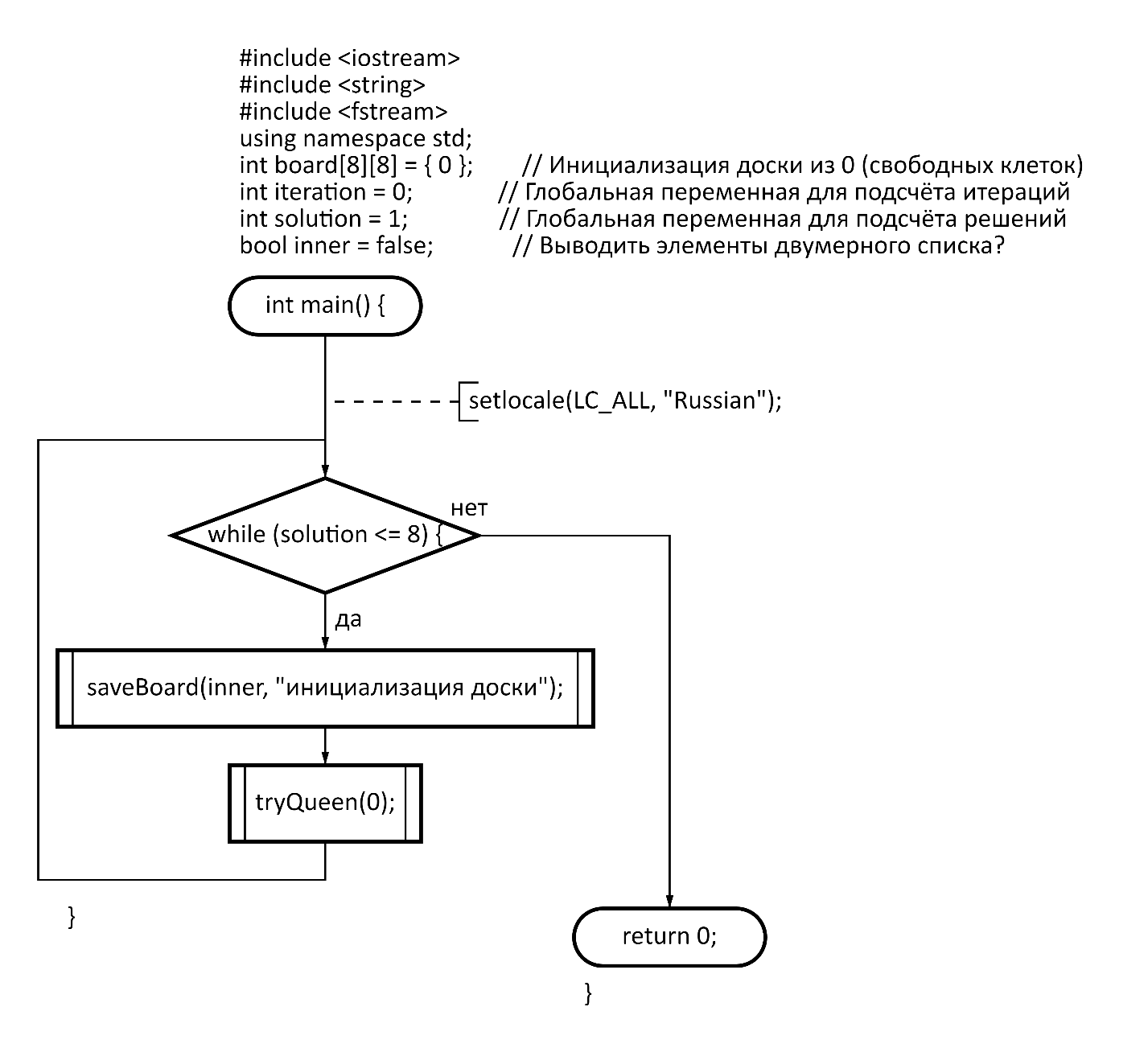
Чтобы ферзи друг друга не «били», на каждой строке, диагонали и каждом столбце должен находиться один ферзь.

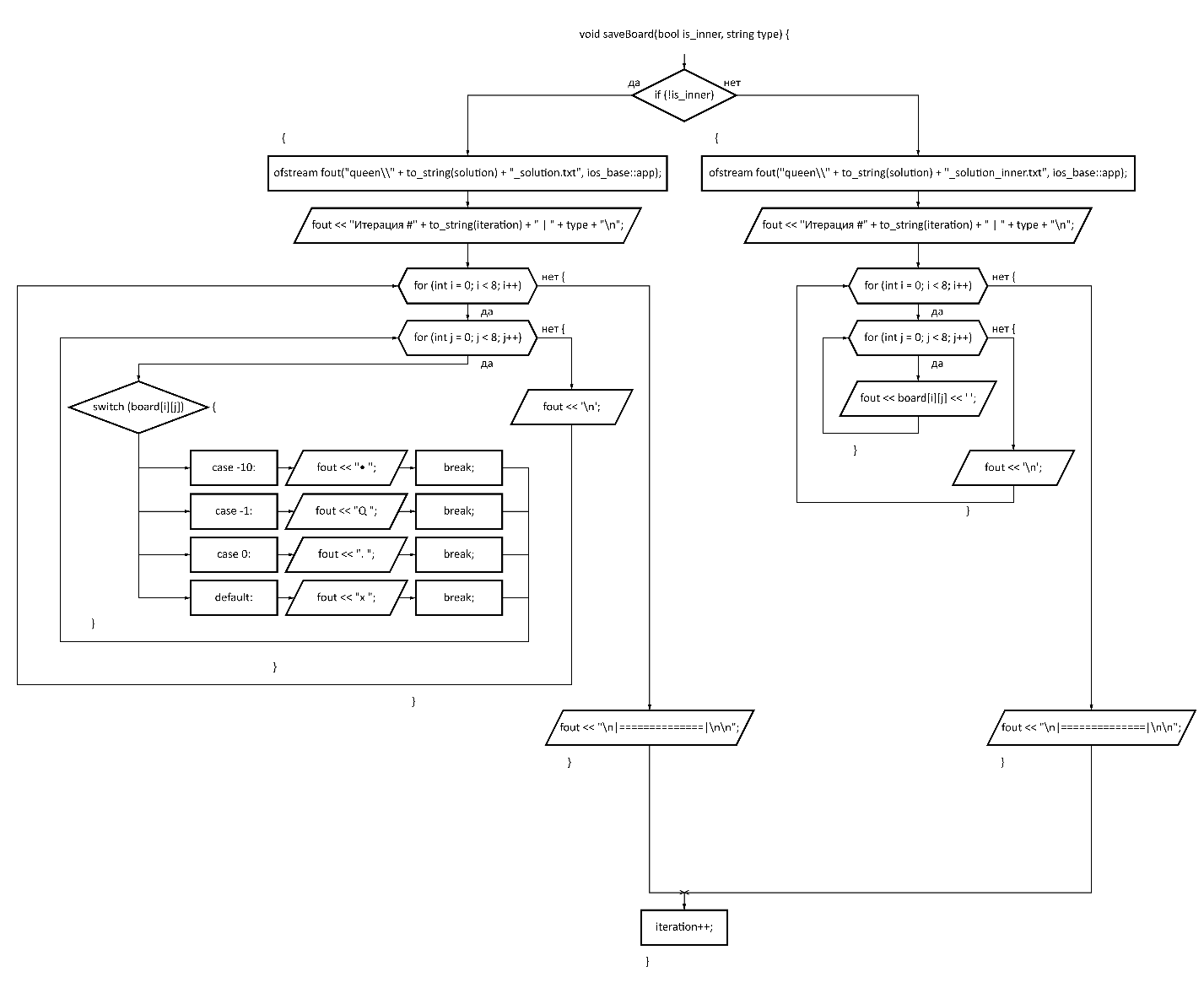
Для расстановки ферзей требуется:

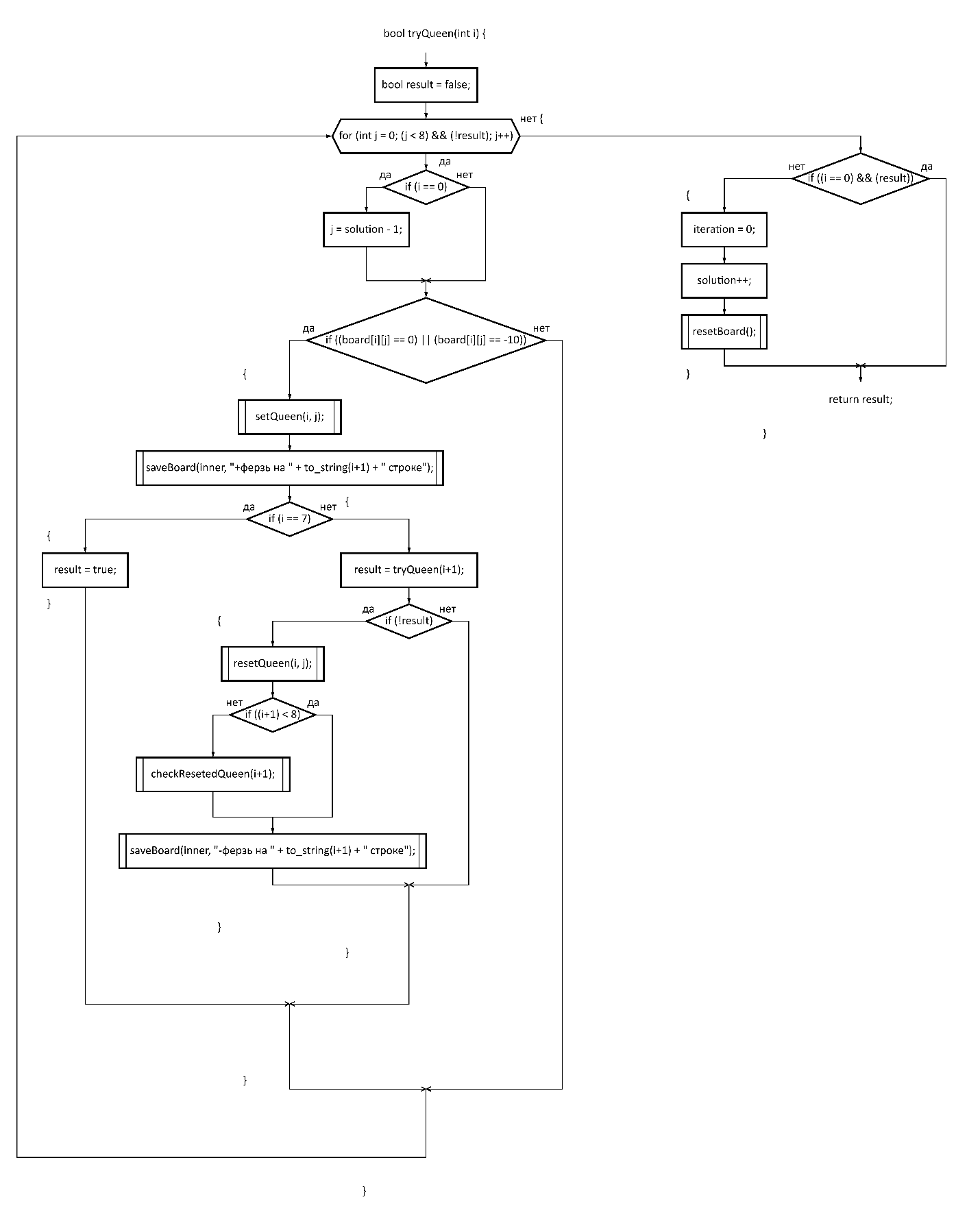
1. Поставить первого ферзя на позицию а1 (первая клетка первой строки).
2. Перейти на следующую строку и поставить ферзя так, чтобы первый ферзь его не бил.
3. Если на какой-либо строке поставить ферзя невозможно (так, чтобы они не «били» друг друга), то возвращаемся на предыдущую строку и ставим ферзя на следующую клетку строки.
4. Повторяем пункты 2 и 3, пока не расставим всех ферзей.

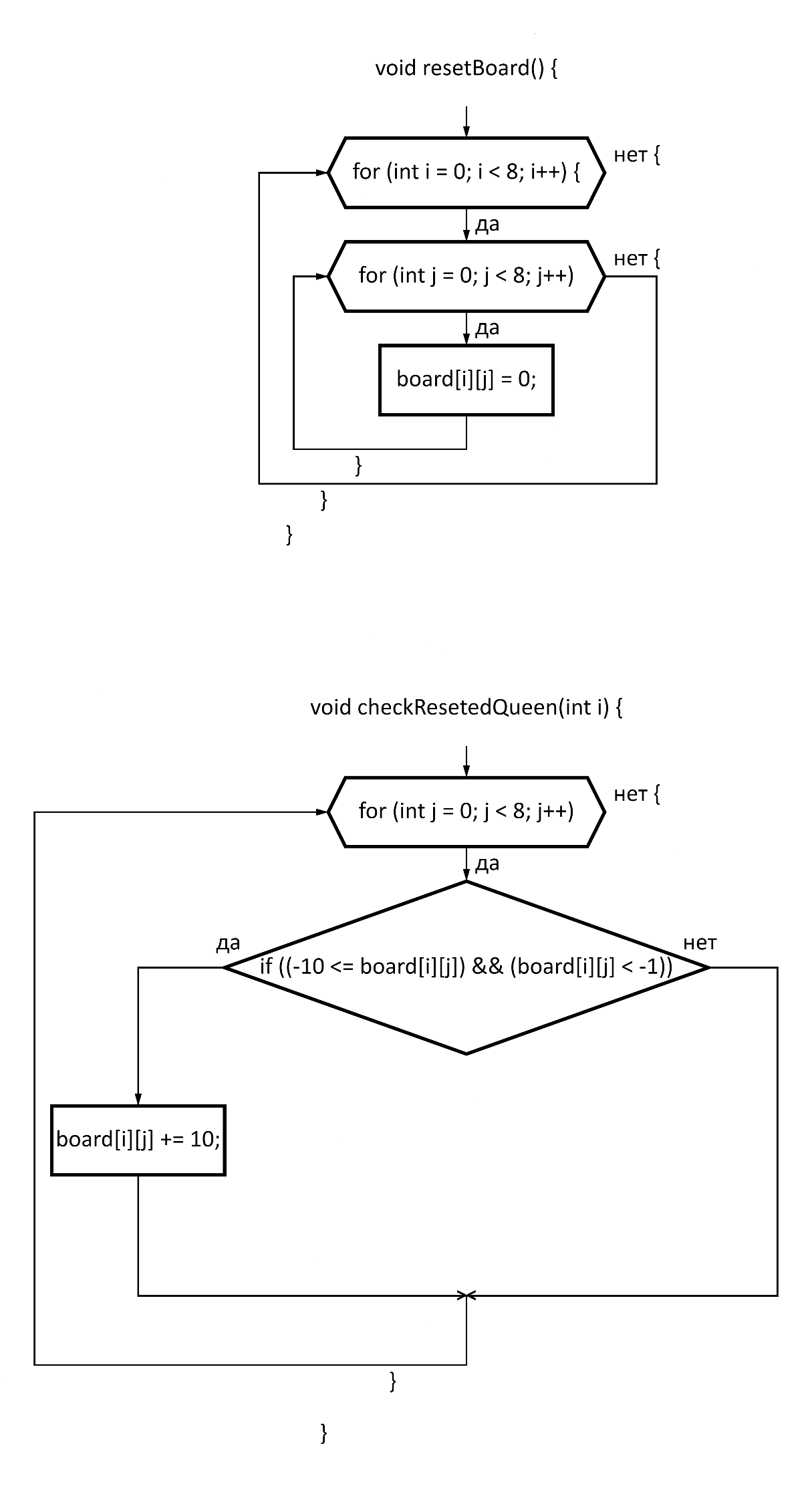
**Анализ алгоритма:**

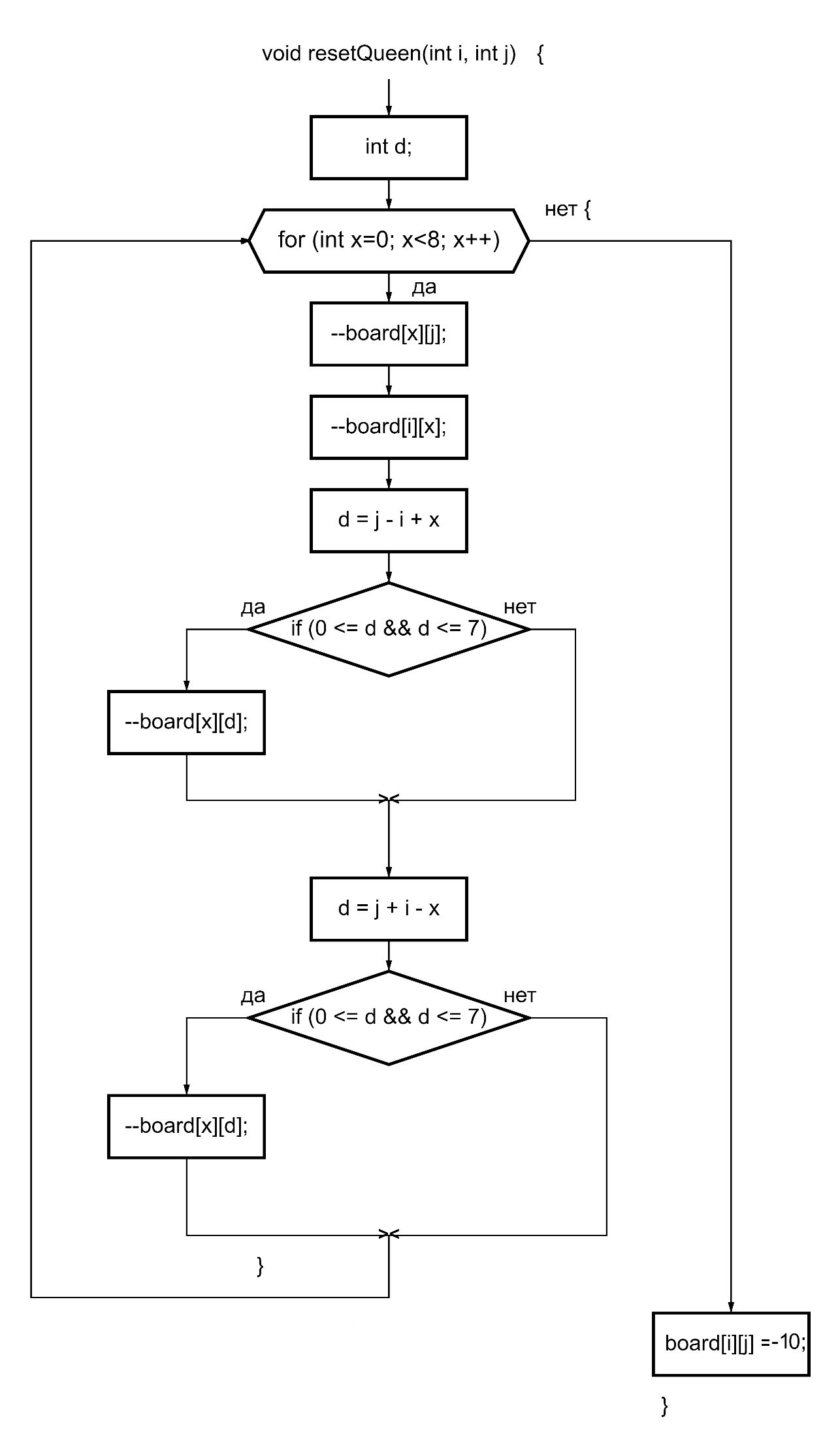
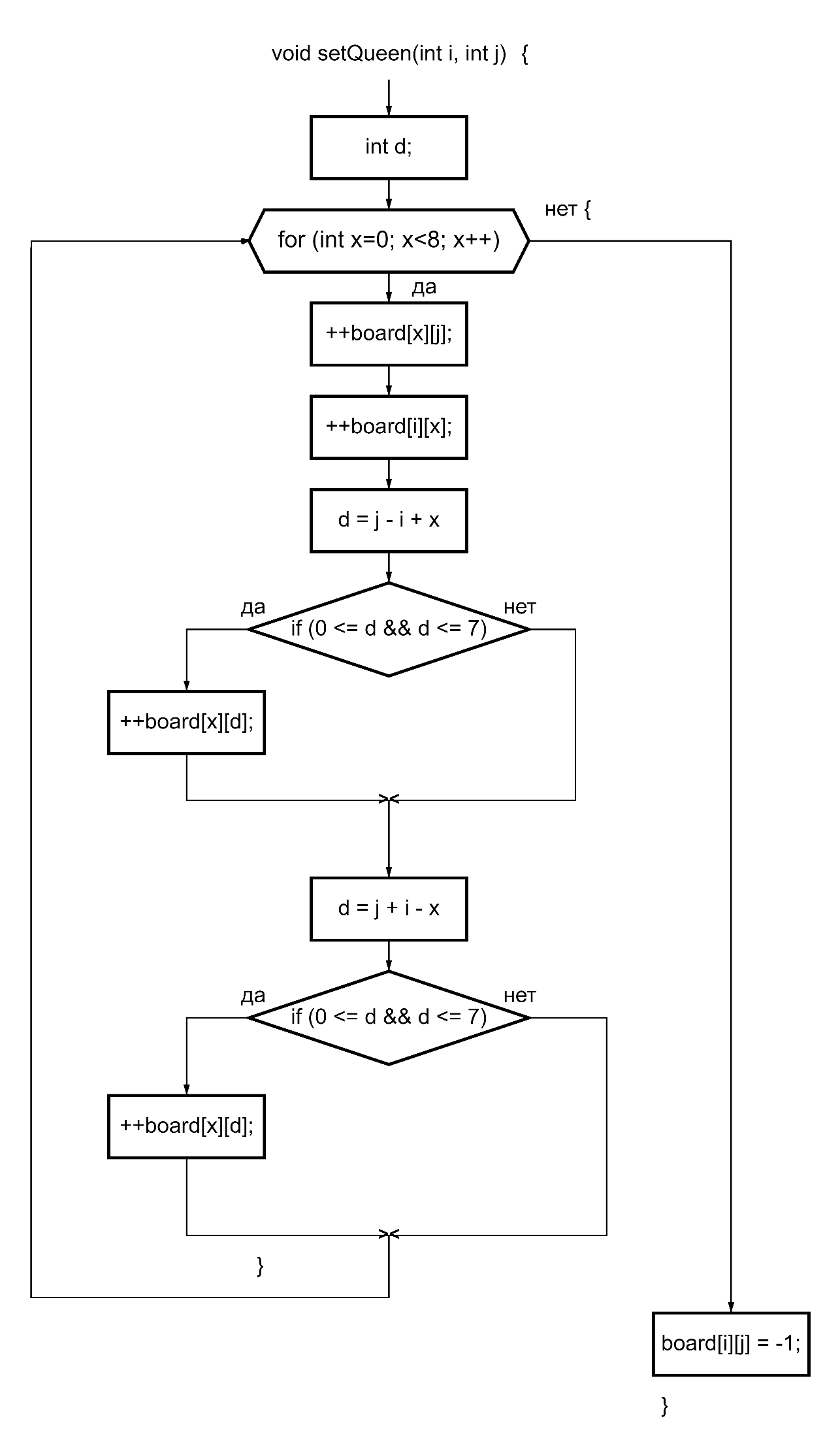
1. Алгоритм адаптирован для поиска нескольких решений, для этого после нахождения первого решения необходимо рассмотреть различные расположения ферзя в первой строке.
2. Итоговый алгоритм находит 8 решений, но для нахождения всех решений необходимо рассмотреть все позиции, на которых ферзя ещё не было (не реализовано).
3. Каждое решение пошаговое, сопровождается комментариями и сохраняется в отдельный текстовый документ в папке queen. Всё это обеспечивает функция saveBoard.
4. Для дальнейшей работы также были реализованы функции resetBoard – для обнуления доски, checkResetedQueen – для очистки лишних меток.

**Блок-схемы: **

****

****

****

****

**Код на языке C++:**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int board[8][8] = { 0 };        // Инициализация доски из 0 (свободных клеток)

int iteration = 0;              // Глобальная переменная для подсчёта итераций

int solution = 1;               // Глобальная переменная для подсчёта решений

bool inner = false;             // Выводить элементы двумерного списка?

void saveBoard(bool is\_inner, string type) {

    if (!is\_inner) {

        ofstream fout("queen\\" + to\_string(solution) + "\_solution.txt", ios\_base::app);

        fout << "Итерация #" + to\_string(iteration) + " | " + type + "\n";

        for (int i = 0; i < 8; i++) {

            for (int j = 0; j < 8; j++) {

                switch (board[i][j]) {

                    case -10: fout << "• "; break;

                    case  -1: fout << "Q "; break;

                    case   0: fout << ". "; break;

                    default : fout << "x "; break;

                }

            }

            fout << '\n';

        }

        fout << "\n|==============|\n\n";

    }

    else {

        ofstream fout("queen\\" + to\_string(solution) + "\_solution\_inner.txt", ios\_base::app);

        fout << "Итерация #" + to\_string(iteration) + " | " + type + "\n";

        for (int i = 0; i < 8; i++) {

            for (int j = 0; j < 8; j++) {

                fout << board[i][j] << ' ';

            }

            fout << '\n';

        }

        fout << "\n|==============|\n\n";

    }

    iteration++;

}

void resetBoard() {

    for (int i = 0; i < 8; i++) {

        for (int j = 0; j < 8; j++) {

            board[i][j] = 0;

        }

    }

}

void checkResetedQueen(int i) {

    for (int j = 0; j < 8; j++) {

        if ((-10 <= board[i][j]) && (board[i][j] < -1)) {

            board[i][j] += 10;

        }

    }

}

void setQueen(int i, int j) {

    for (int x = 0; x < 8; x++) {

        ++board[x][j];          // Разметка вертикали

        ++board[i][x];          // Разметка горизонтали

        int d;

        // Разметка главной диагонали

        d = j - i + x;

        if (0 <= d && d <= 7) {

            ++board[x][d];

        }

        // Разметка побочной диагонали

        d = j + i - x;

        if (0 <= d && d <= 7) {

            ++board[x][d];

        }

    }

    board[i][j] = -1;           // Разметка ферзя

}

void resetQueen(int i, int j) {

    for (int x = 0; x < 8; x++) {

        --board[x][j];          // Разметка вертикали

        --board[i][x];          // Разметка горизонтали

        int d;

        // Разметка главной диагонали

        d = j - i + x;

        if (0 <= d && d <= 7) {

            --board[x][d];

        }

        // Разметка побочной диагонали

        d = j + i - x;

        if (0 <= d && d <= 7) {

            --board[x][d];

        }

    }

    board[i][j] = -10;           // Разметка ферзя

}

bool tryQueen(int i) {

    bool result = false;

    for (int j = 0; (j < 8) && (!result); j++) {

        if (i == 0) { j = solution - 1; }

        if ((board[i][j] == 0) || (board[i][j] == -10)) {

            setQueen(i, j);

            saveBoard(inner, "+ферзь на " + to\_string(i+1) + " строке");

            if (i == 7) {

                result = true;  // Последнего ферзя поставили

            }

            else {

                result = tryQueen(i + 1);

                if (!result) {

                    resetQueen(i, j);

                    if ((i+1) < 8) { checkResetedQueen(i+1); }

                    saveBoard(inner, "-ферзь на " + to\_string(i+1) + " строке");

                }

            }

        }

    }

    if ((i == 0) && (result)) {

        iteration = 0;

        solution++;

        resetBoard();

    }

    return result;

}

int main() {

    setlocale(LC\_ALL, "Russian");

    while (solution <= 8) {

        saveBoard(inner, "инициализация доски");

        tryQueen(0);

    }

    return 0;

}

**Результаты:**

|  |  |  |  |
| --- | --- | --- | --- |
| inner = false | | inner = true | |
|  | |  | |
|  |  |  |  |
|  | |  | |

**Вывод:**

Я смог реализовать программу, находящую расстановку ферзей для задачи о восьми ферзях.