Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**Лабораторная работа**

**Тема:** Решение задачи о 8 ферзях

Семестр: 2

Выполнил студент ИВТ-23-2б:

Злыгостев Денис Николаевич

(дата, подпись)

Проверила доцент кафедры ИТАС:

Полякова Ольга Андреевна

(дата, подпись)

2024 г.

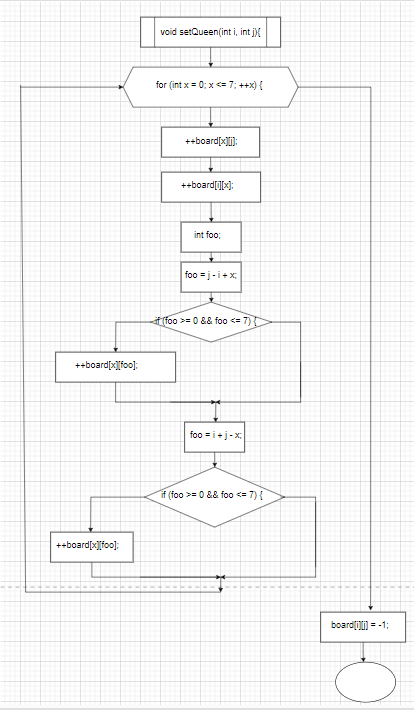
**Постановка задачи**

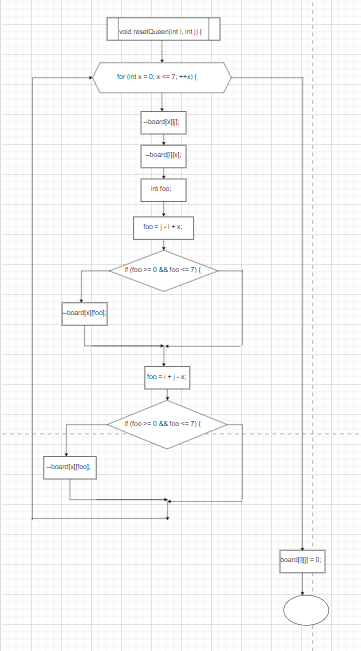
Создать программу, которая располагает ферзи в таком положении, чтобы они не били друг друга

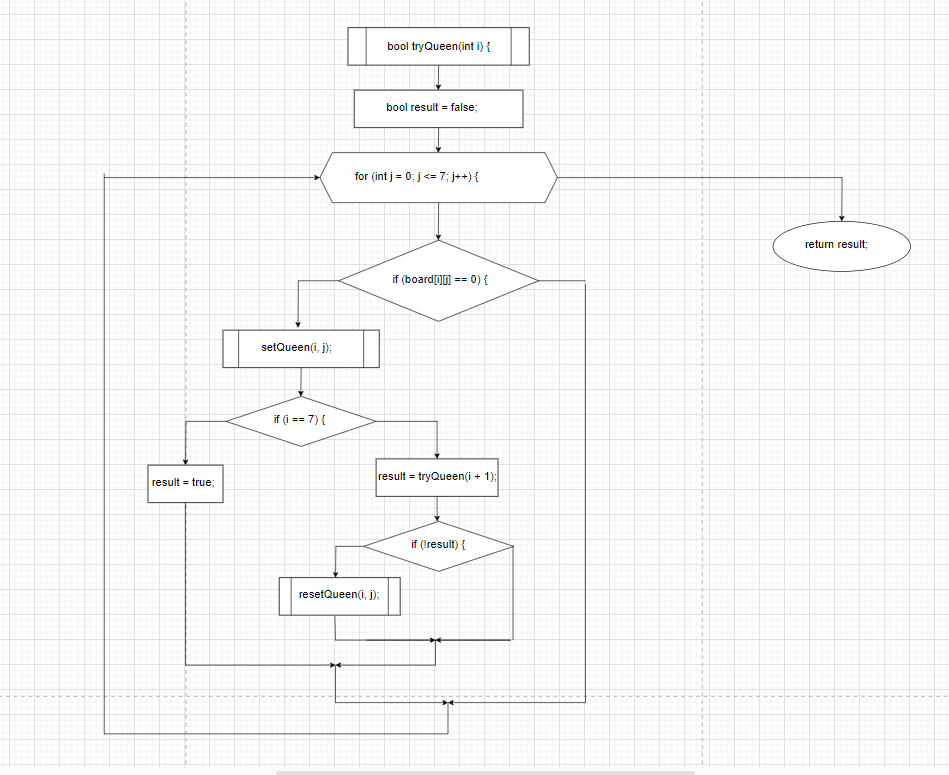
**Анализ задачи**

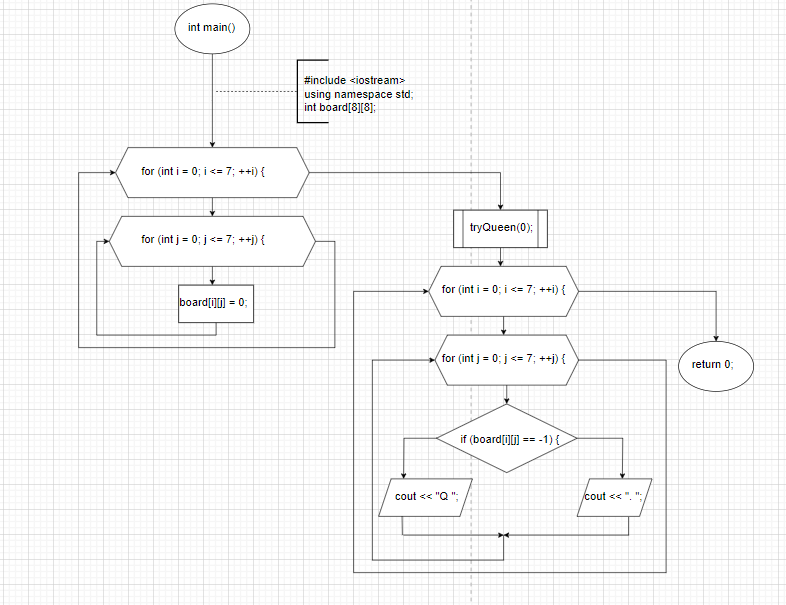
1. Ферзь может ходить в любом направлении по горизонтали, вертикали, диагонали и на любое количество клеток, рубит он так же, как ходит.
2. Чтобы ферзи друг друга не «били», на каждой строке, диагонали и каждом столбце должен находиться один ферзь.
3. Для расстановки ферзей требуется:
4. Поставить первого ферзя на позицию а1 (первая клетка первой строки).
5. Перейти на следующую строку и поставить ферзя так, чтобы первый ферзь его не бил.
6. 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.
7. Повторяем пункты 2 и 3, пока не расставим всех ферзей.

**Блок схема**









**Код**

#include <iostream>

using namespace std;

int board[8][8];

void setQueen(int i, int j)

{

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

++board[x][j];

++board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7) {

++board[x][foo];

}

foo = i + j - x;

if (foo >= 0 && foo <= 7) {

++board[x][foo];

}

}

board[i][j] = -1;

}

void resetQueen(int i, int j) {

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

--board[x][j];

--board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7) {

--board[x][foo];

}

foo = i + j - x;

if (foo >= 0 && foo <= 7) {

--board[x][foo];

}

}

board[i][j] = 0;

}

bool tryQueen(int i) {

bool result = false;

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

if (board[i][j] == 0) {

setQueen(i, j);

if (i == 7) {

result = true;

}

else {

result = tryQueen(i + 1);

if (!result) {

resetQueen(i, j);

}

}

}

}

return result;

}

int main()

{

tryQueen(0);

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

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

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

cout << "I ";

}

else {

cout << "0 ";

}

}

cout << endl;

}

return 0;

}

**Скрин из GitHub**

