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

ПНИПУ

**Лабораторная работа  
“Восемь ферзей”**

Выполнил:   
студент группы РИС-23-1б   
Хасаншин Арсен Рустамович

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

2024 г.

**Постановка задачи:**Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого». Подразумевается, что ферзь бьёт все клетки, расположенные по вертикалям, горизонталям и обеим диагоналям.

**Анализ задачи:**

1) Ферзей восемь, так же как и строк – следовательно на каждой строке должен находиться ферзь.

2) Устанавливаем первого ферзя на доску и отмечаем клетки, которые он бьет. (Увеличивая их значение на 1)

3) Ищем на следующей строке первое свободное место для установки следующего ферзя и так же отмечаем клетки, которые он бьет.

4) Продолжаем заполнять доску ферзями, пока не достигнем 8 строки, либо пока на следующей строке не будет места для установки ферзя.

5) Если следующая строка занята, то убираем ферзя и ищем в текущей строке следующее свободное место, если его нет, то идем к предыдущей строке и повторяем те же действия.

В конце цифры заменяются на q и \* для удобства восприятия.

**Код**

#include <iostream>

using namespace std;

int board[8][8] = { 0 };

int k = 0;

// Функция для вывода результата

void print() {

cout << "Ход номер " << ++k << endl;

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

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

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

}

cout << endl;

}

cout << endl;

}

// Заполнение поля и установка ферзя

void setQueen(int i, int j) {

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

// Строки и столбцы

++board[k][j];

++board[i][k];

int diag;

// Заполнение главной диагонали

diag = j - i + k;

if (diag >= 0 && diag < 8) {

++board[k][diag];

}

// Побочная диагональ

diag = j + i - k;

if (diag >= 0 && diag < 8) {

++board[k][diag];

}

}

board[i][j] = -1; // Метка отличающая ферзя от остальных клеток

print();

}

// Удаление ферзя с доски, действия аналогичный установке, но со знаком минус

void resetQueen(int i, int j) {

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

--board[k][j];

--board[i][k];

int diag;

diag = j - i + k;

if (diag >= 0 && diag < 8) {

--board[k][diag];

}

diag = j + i - k;

if (diag >= 0 && diag < 8) {

--board[k][diag];

}

}

board[i][j] = 0; // Обозначение свободной клетки

print();

}

bool tryQueen(int i) {

bool result = false;

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

if (board[i][j] == 0) { // Если находиться свободное место, то устанавливаем ферзя

setQueen(i, j);

if (i == 7) {

result = true; // Если дошли до конца, то завершаем функцию

}

else {

// В противном случае удаляем ферзя

if (!(result = tryQueen(i + 1))) {

resetQueen(i, j);

}

}

}

if (result) {

break;

}

}

return result;

}

int main() {

setlocale(LC\_ALL, "RU");

// Вызываем функцию с 1 строки

tryQueen(0);

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

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

// Заменяем ферзей на q, а остальные цифры на \*

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

cout << "Q ";

}

else {

cout << "\* ";

}

}

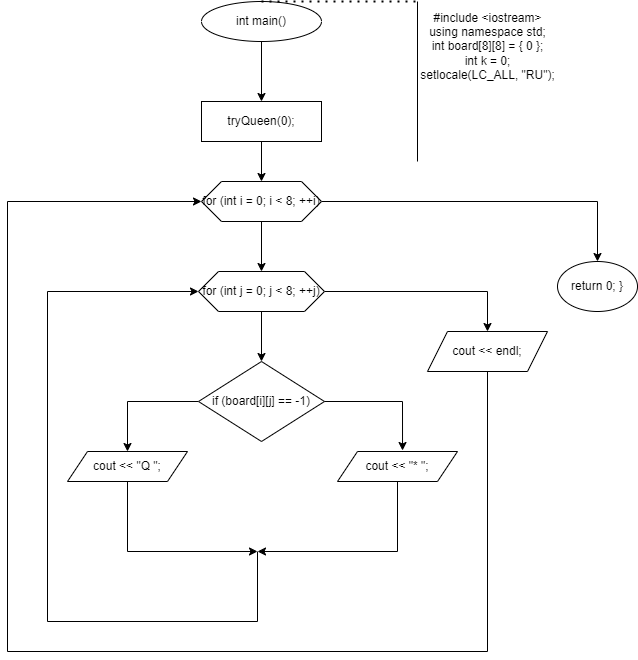
cout << endl;

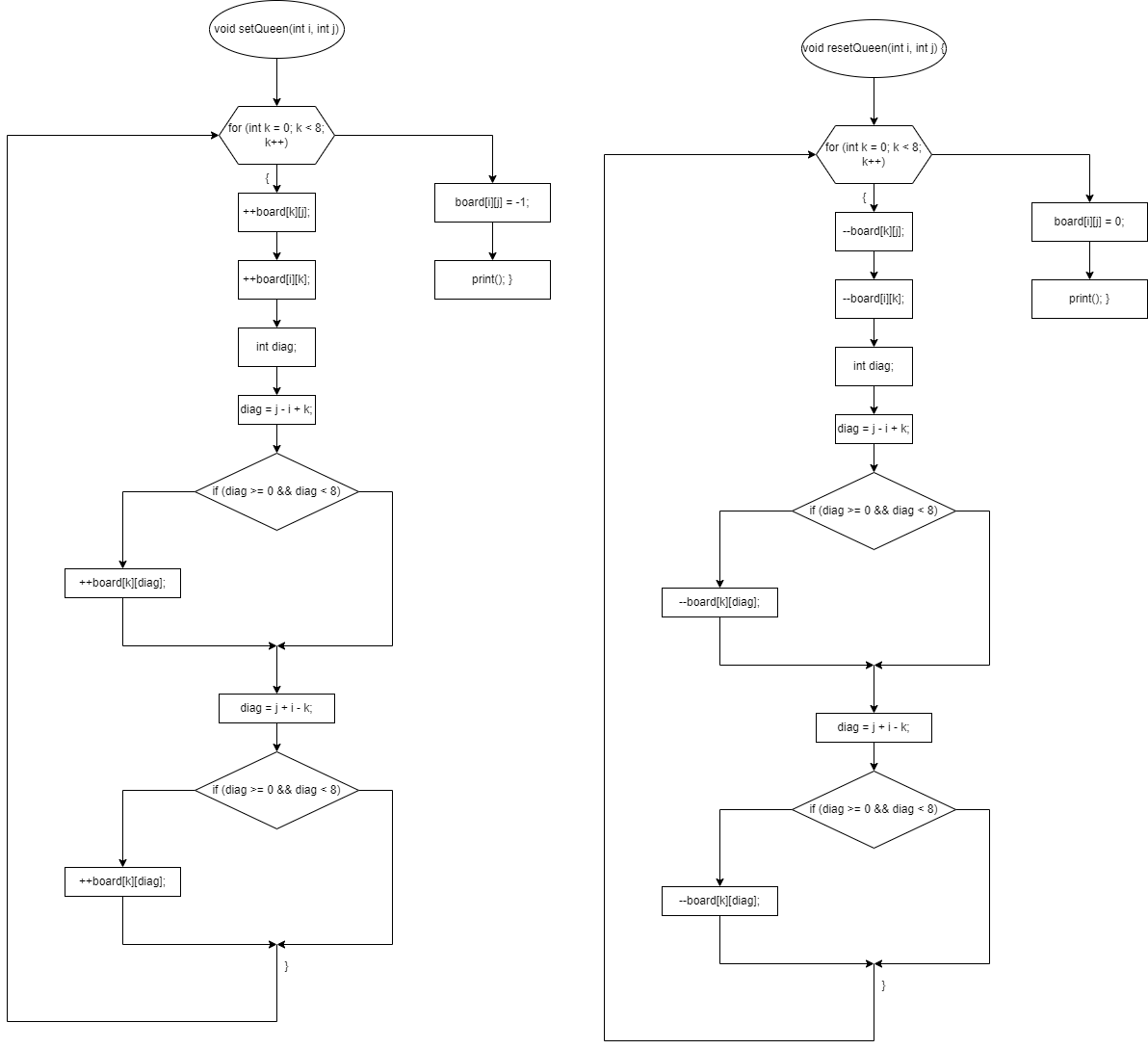
}

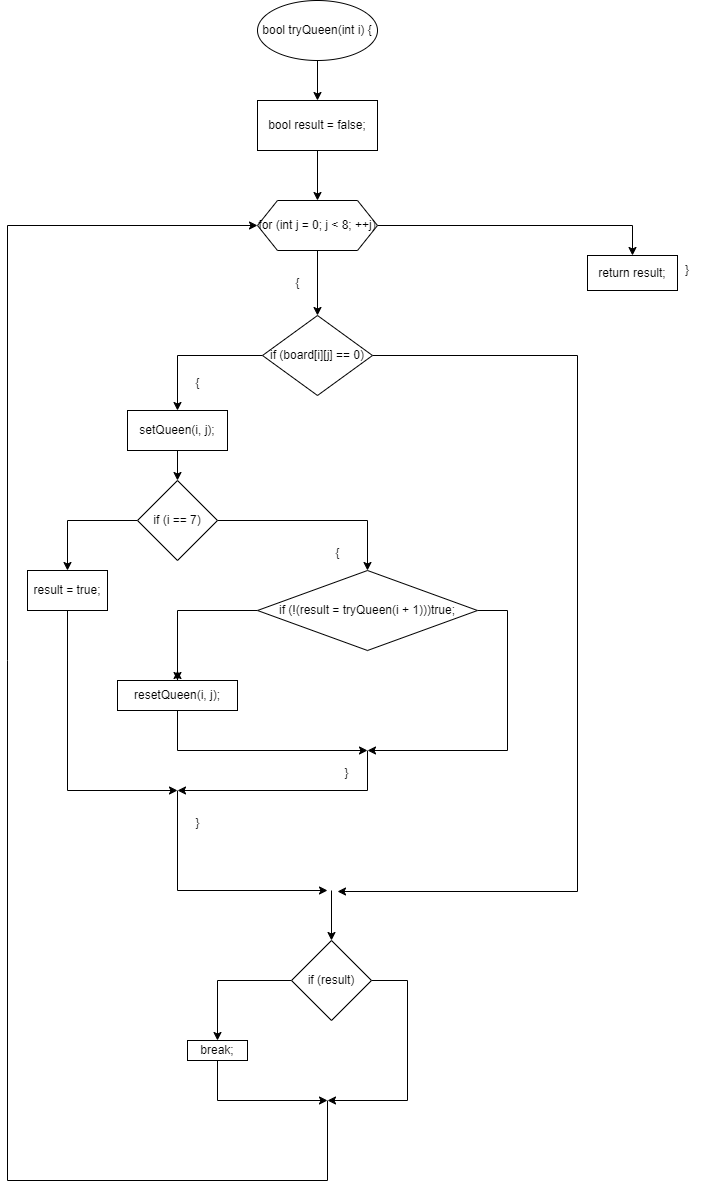
return 0;

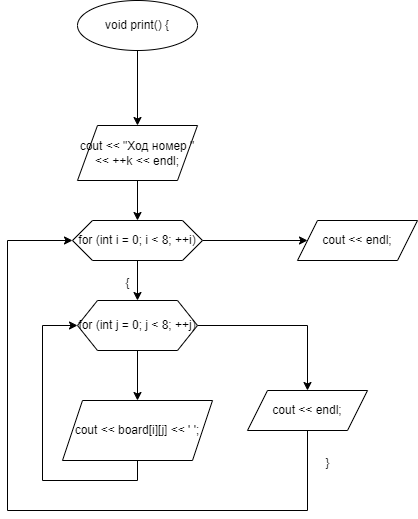
}

**Блок схема.**

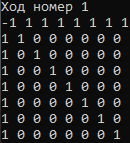
****

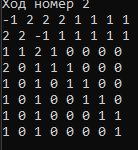
****

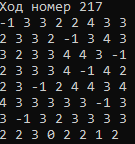
****

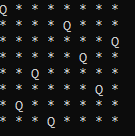


**Пример работы**









**Выводы.** Программа пошагово показала все шаги алгоритма и выдала корректный финальный результат.