Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ.**

Тема: «Задача о восьми ферзях».

Выполнил работу

Студент группы РИС-23-1б

Валинуров Р.Н.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь, 2024.

**Постановка задачи.**

Требуется разработать программу, которая будет решать задачу с 8-ю ферзями.

**Анализ задачи.**

1. Создаем двумерный массив board[8][8], который используется для представления шахматной доски.

2. Функции setQueen() и resetQueen() обновляют значения доски при установке и снятии ферзей, чтобы указать клетки, которые они атакуют.

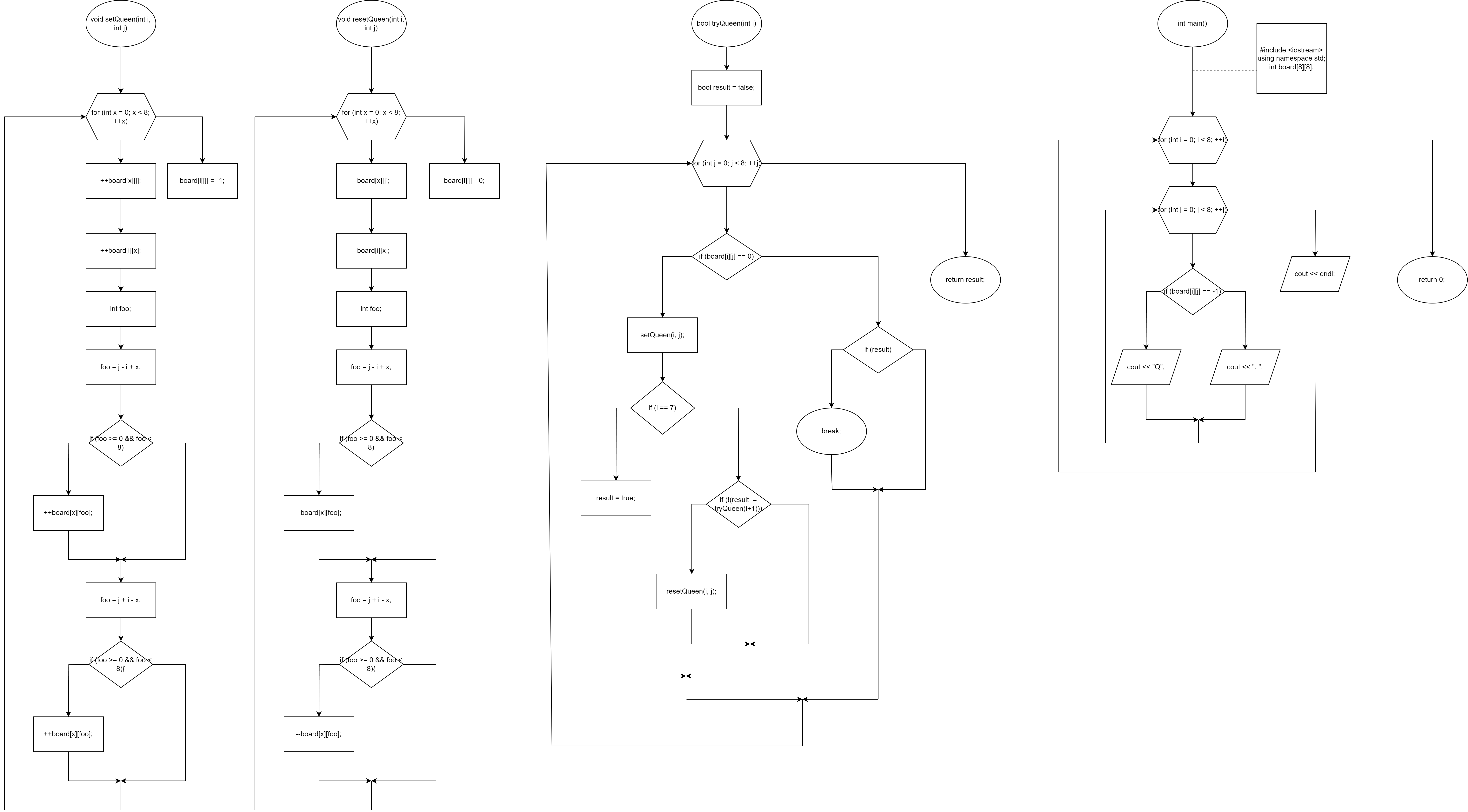
3. Функция tryQueen() пытается разместить ферзя в каждой строке доски, начиная с верхней.

4. При поиске свободной клетки в строке, вызывается setQueen() для установки ферзя и обновления значений доски. Если ферзь успешно размещен в текущей строке, функция рекурсивно вызывается для следующей строки.

5. Все клетки, атакуемые ферзем, увеличиваются на 1, и уменьшаются при снятии ферзя. Это позволяет проверять, не находится ли другой ферзь в области атаки каждого нового ферзя.

6. Проверки в setQueen() и resetQueen() будут гарантиртировать, что ферзь не будет размещен в клетку, которая находится под атакой других ферзей.

7. Когда все 8 ферзей успешно размещены, программа выводит конечное состояние доски, где ферзи отмечены символами "Q", а пустые клетки - ". ".

**Блок-Схема.**

**Код.**

#include <iostream>

using namespace std;

int board[8][8];

void setQueen(int i, int j)

{

for (int x = 0; x < 8; ++x)

{

++board[x][j];

++board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo < 8)

++board[x][foo];

foo = j + i - x;

if (foo >= 0 && foo < 8)

++board[x][foo];

}

board[i][j] = -1;

}

void resetQueen(int i, int j)

{

for (int x = 0; x < 8; ++x)

{

--board[x][j];

--board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo < 8)

--board[x][foo];

foo = j + i - x;

if (foo >= 0 && foo < 8)

--board[x][foo];

}

board[i][j] = 0;

}

bool tryQueen(int i)

{

bool result = false;

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

{

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

{

setQueen(i, j);

if (i == 7)

result = true;

else

{

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

resetQueen(i, j);

}

}

if (result)

break;

}

return result;

}

int main()

{

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

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

board[i][j] = 0;

tryQueen(0);

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

{

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

{

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

cout << "Q";

else

cout << ". ";

}

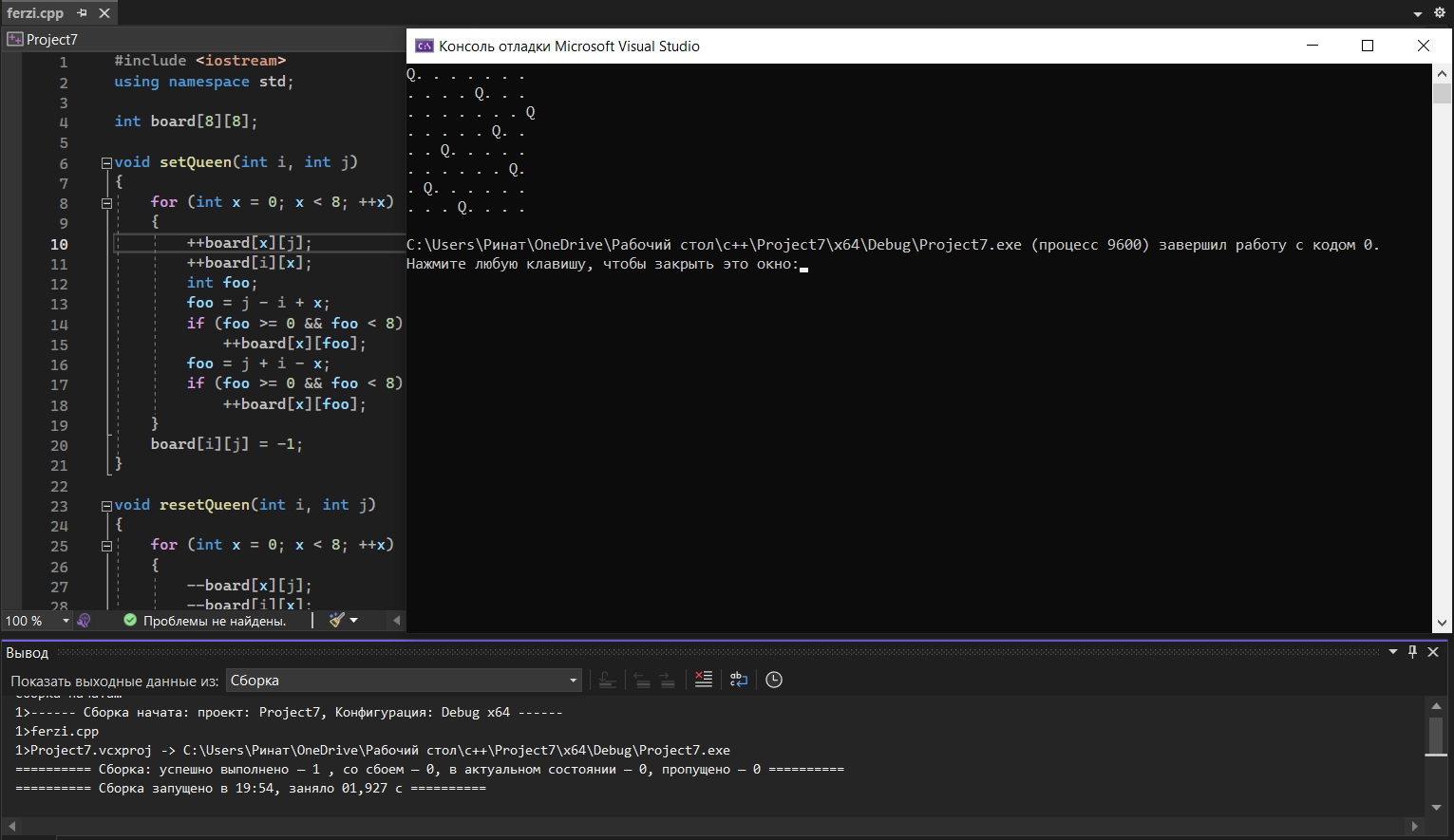
cout << endl;

}

return 0;

}

**Демонстрация работы кода.**



**Вывод.**

код решает задачу успешно и выводит на экран правильное распределение 8 ферзей на доске.