Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

**«Задача о восьми ферзях»**

Выполнил:

студент группы РИС-23-2б

Кобзев Станислав Игоревич  
Проверила:

доцент кафедры ИТАС

О.А. Полякова

Пермь 2024 г.

**Задача о восьми ферзях**

**Постановка задачи:**

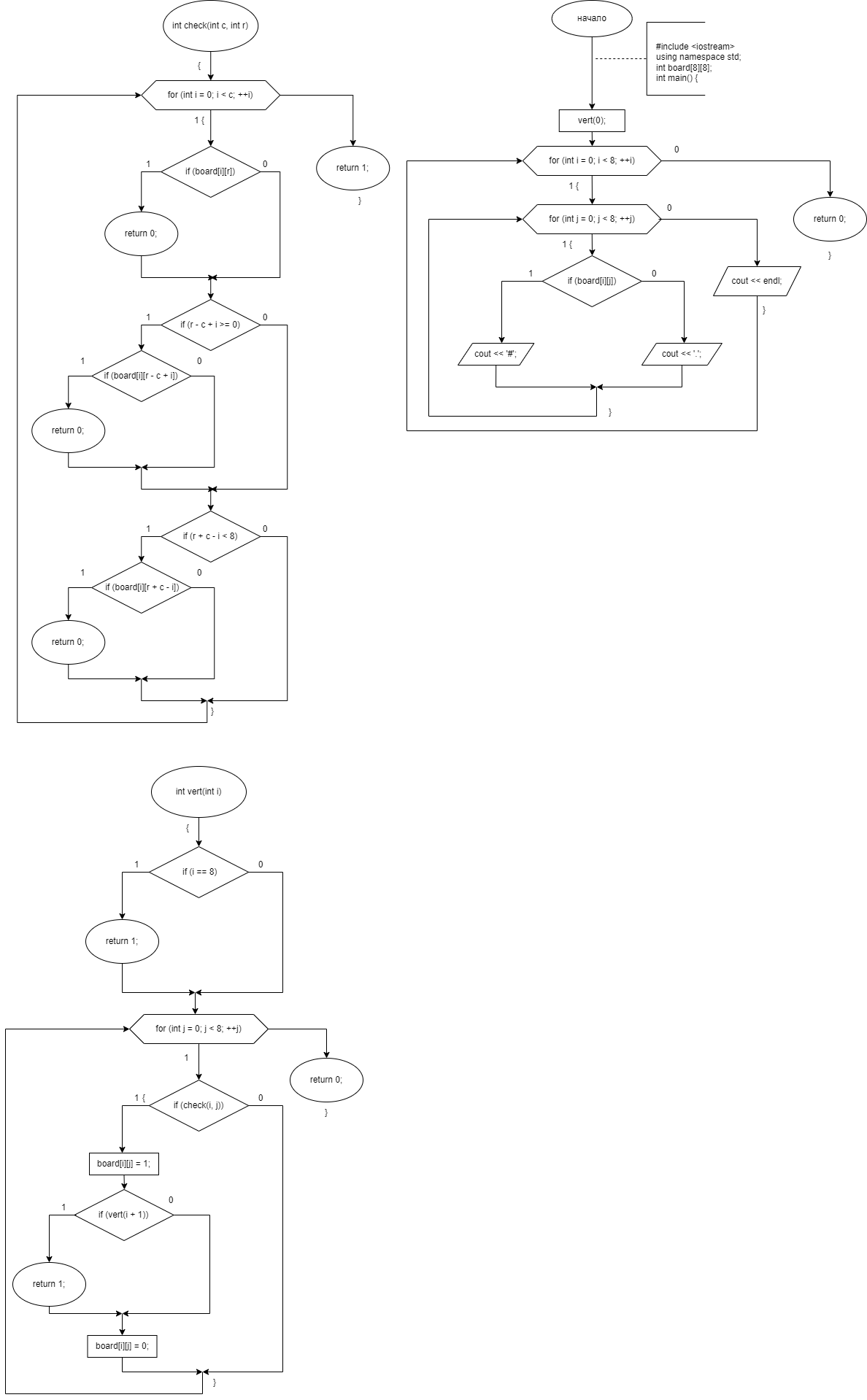
Необходимо расставить 8 ферзей на шахматной доске так, чтобы они не атаковали друг друга.

**Анализ задачи:**

Два ферзя не могут стоять на одной вертикали.

Создается рекурсивная функция vert(i)  установки ферзя в i-й вертикали. Эта функция поочередно проверяет все клетки, и если можно, то ставит ферзя и рекурсивно вызывает vert(i+1). Если алгоритм дошел до 8-й вертикали (несуществующей), то все фигуры расставлены. Если фигуру установить нельзя, то возвращается 0, и на предыдущей вертикали происходит новая попытка поставить ферзя.

**Блок схема:**

Рисунок 1 – блок схема

**Код программы:**

#include <iostream>

using namespace std;

int board[8][8];

int check(int c, int r) // Проверка, что можно ставить в клетку [r][c]

{

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

{

// Проверка горизонтали

if (board[i][r]) return 0;

// Проверка диагонали

if (r - c + i >= 0) if (board[i][r - c + i]) return 0;

if (r + c - i < 8) if (board[i][r + c - i]) return 0;

}

return 1;

}

int vert(int i) // Установка i-й вертикали

{

if (i == 8) return 1; // Фигуры расставлены, алгоритм завершен

for (int j = 0; j < 8; ++j) // Поиск места по ввертикали

if (check(i, j)) // Если можно -

{

board[i][j] = 1; // Фигура устанавливается

if (vert(i + 1)) return 1; // Получилось поставить

board[i][j] = 0; // Не получилось - удаление ферзя

}

return 0;

}

int main()

{

vert(0);

// Вывод результата

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

{

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

if (board[i][j]) cout << '#';

else cout << '.';

}

cout << endl;

}

return 0;

}

**Результат работы алгоритма:**

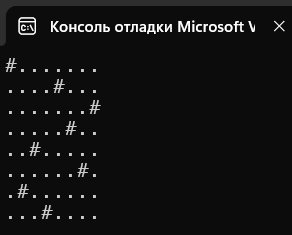
****

Рисунок 2 – результат работы алгоритма для поля размером 8х8



Рисунок 3 – результат работы алгоритма для поля размером 4х4

**Визуализация работы алгоритма:**

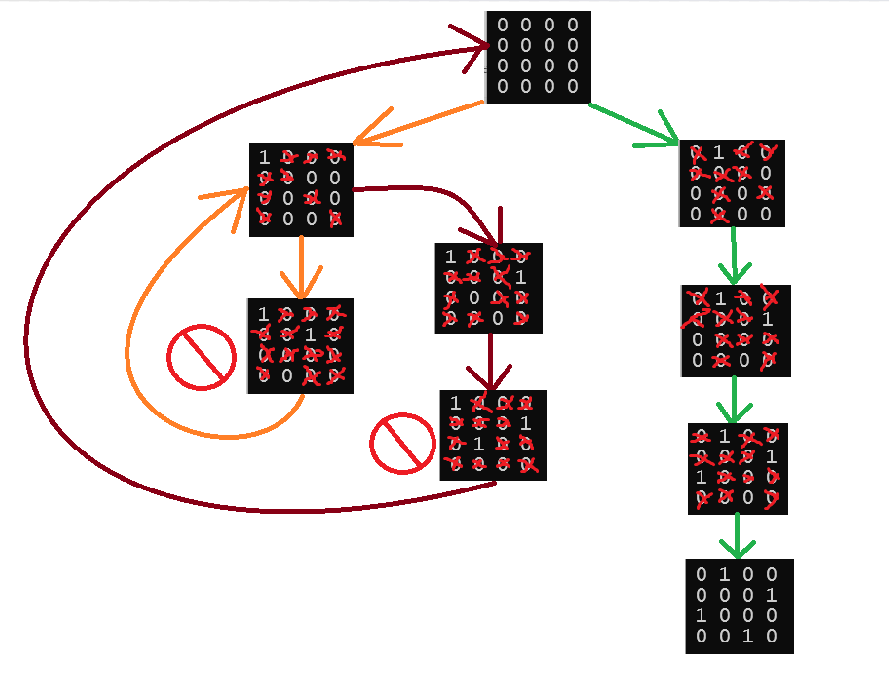
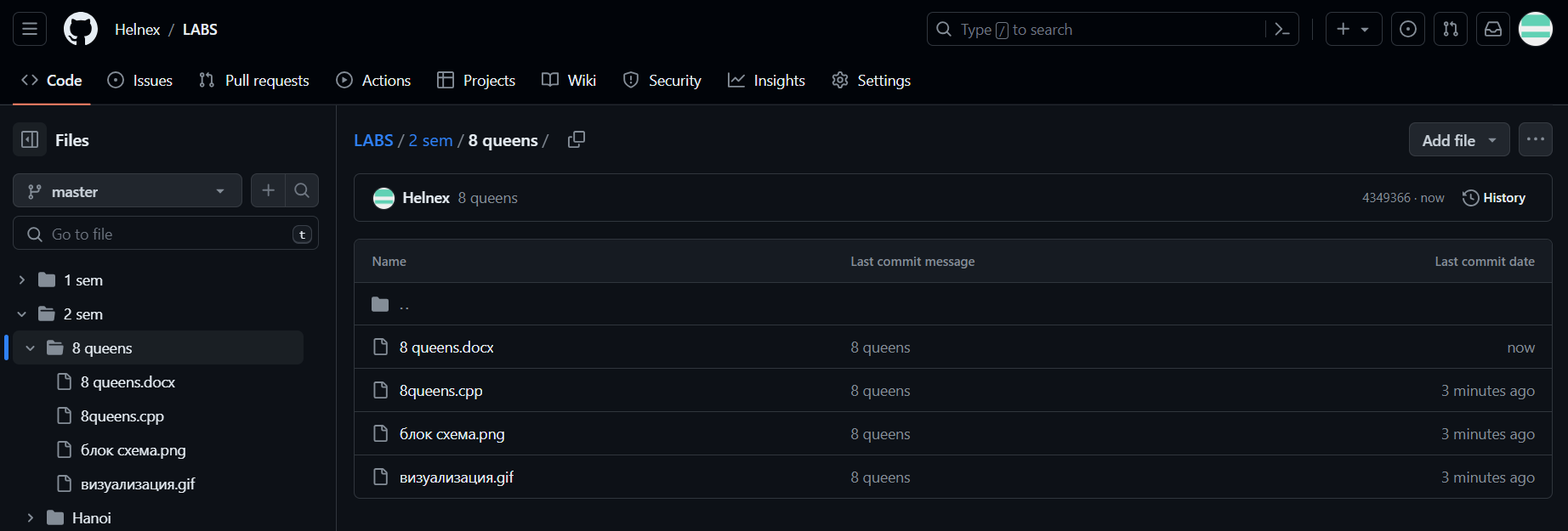
Рассмотрим работу алгоритма на примере доски 4х4, на которой нужно расставить 4 фигуры.

Рисунок 4 – визуализация работы алгоритма для поля размером 4х4

**Скриншот с Github**

****