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

**ОТЧЕТ**

**Тема:** 8 Ферзей

Семестр: 2

Выполнил студент РИС-23-2б:

Серебряков Егор Константинович

(дата, подпись)

Проверила:

О. А. Полякова

(дата, подпись)

Пермь 2024

**Анализ и решение**

**Постановка задачи:**

Разместить 8 ферзей на шахматной доске размера 8 на 8 так, чтобы ни один из ферзей не бил другого

**Анализ:**

1. Ставим ферзя на первую свободную ячейку в матрице
2. Отмечаем клетки, которые бьет ферзь
3. Переходим на следующую строку и повторяем шаг 2-3
4. Если в следующей строке не находится свободной ячейки, то с помощью рекурсии возвращаемся на нужное количество ходов назад и перемещаем нужного ферзя на другую свободную ячейку

**Словесный алгоритм:**

1. Создаём шаблоны функций и задаём размер шахматной доски
2. С помощью рекурсивных функций, флажков булевого типа и циклических процессов производим нахождение удачной комбинации расстановки ферзей на шахматной доске
3. С помощью функции viewfield выводим полученный результат после выполнения рекурсивных функций checkQueen и setQueen

**Код программы:**

#include <iostream>

using namespace std;

const int n = 8;

bool checkQueen(int field[n][n], int stroka, int col);

bool setQueen(int field[n][n], int stroka);

void viewfield(int field[n][n]);

int main() {

int field[n][n] = { 0 };

if (setQueen(field, 0) != 0)

{

viewfield(field);

}

return 0;

}

bool checkQueen(int field[n][n], int stroka, int col) {

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

if (field[stroka][i] || field[i][col] ||

(stroka - i >= 0 && col - i >= 0 && field[stroka - i][col - i]) ||

(stroka + i < n && col + i < n && field[stroka + i][col + i]) ||

(stroka - i >= 0 && col + i < n && field[stroka - i][col + i]) ||

(stroka + i < n && col - i >= 0 && field[stroka + i][col - i])) {

return false;

}

}

return true;

}

bool setQueen(int field[n][n], int stroka)

{

if (stroka == n) return true;

for (int col = 0; col < n; col++)

{

if (checkQueen(field, stroka, col))

{

field[stroka][col] = 1;

if (setQueen(field, stroka + 1))

{

return true;

}

field[stroka][col] = 0;

}

}

return false;

}

void viewfield(int field[n][n])

{

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

{

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

{

if (field[i][j] == 1)

{

cout << "I ";

}

else

{

cout << ". ";

}

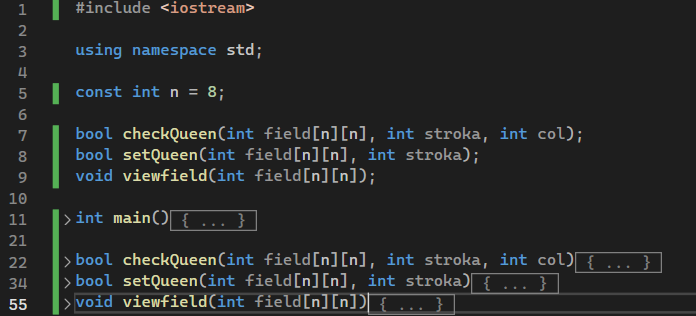
}

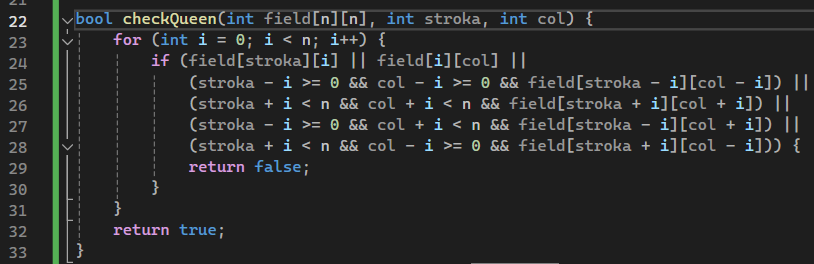
cout << endl;

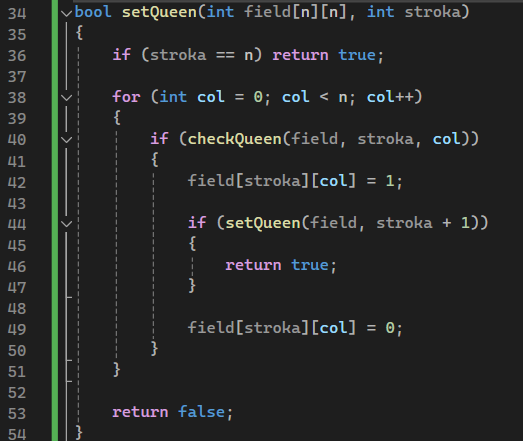
}

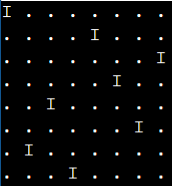
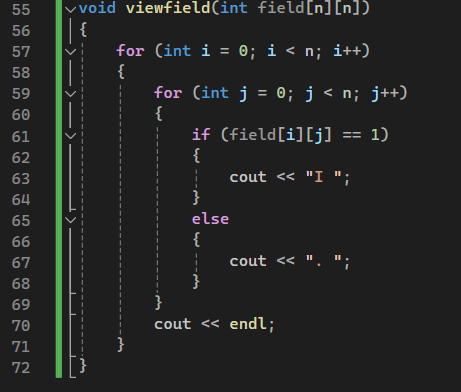
}

**Скриншоты кода:**

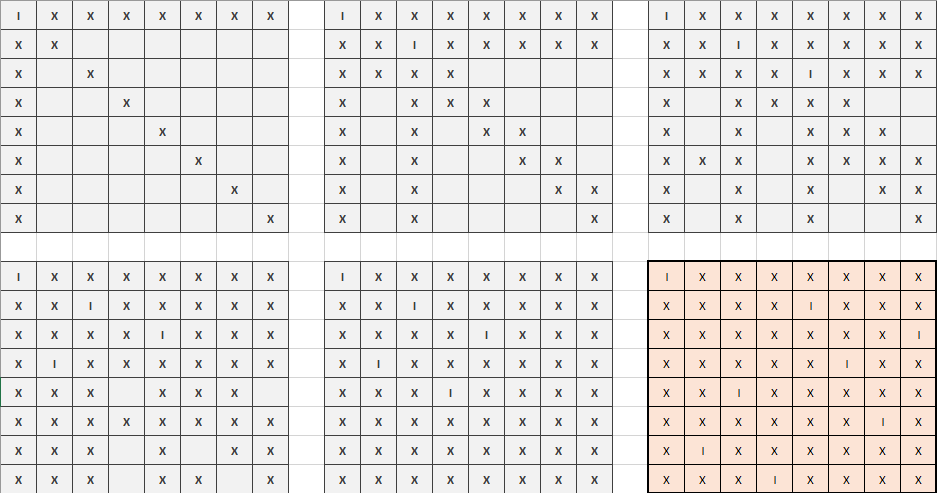
****

****

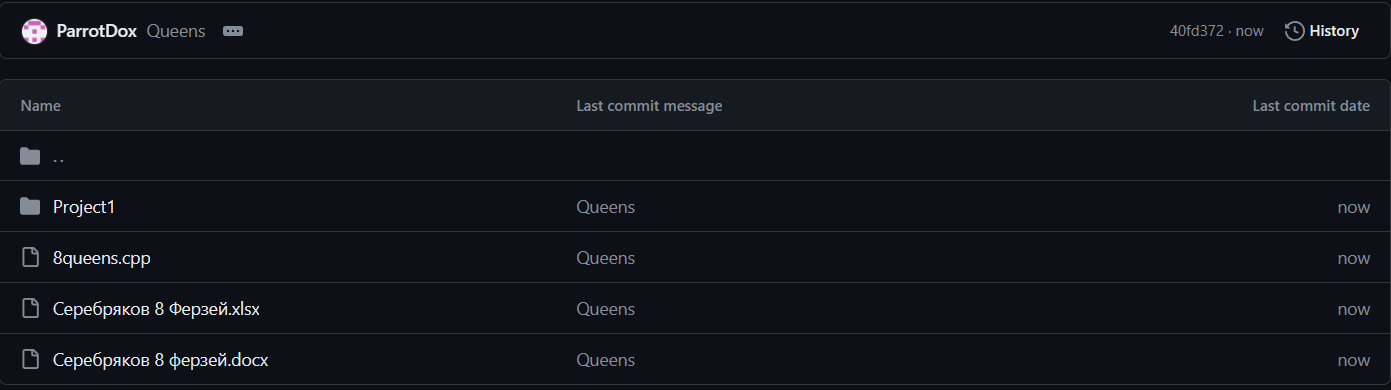
****

****

**Визуализация в Excel:**

****

**Git:**

****

**Блок-схема:**

