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 ПО**

**ЛАБОРАТОРНОЙ РАБОТЕ**

**Тема:** 8 ферзей

Выполнила студентка РИС-23-2б:

Куклина Юлия Витальевна

Проверила: доцент кафедры ИТАС

О.А. Полякова

2024

**Задача**

Расставить на стандартной 64-х клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого. Реализовать решение задачи на языке C++.

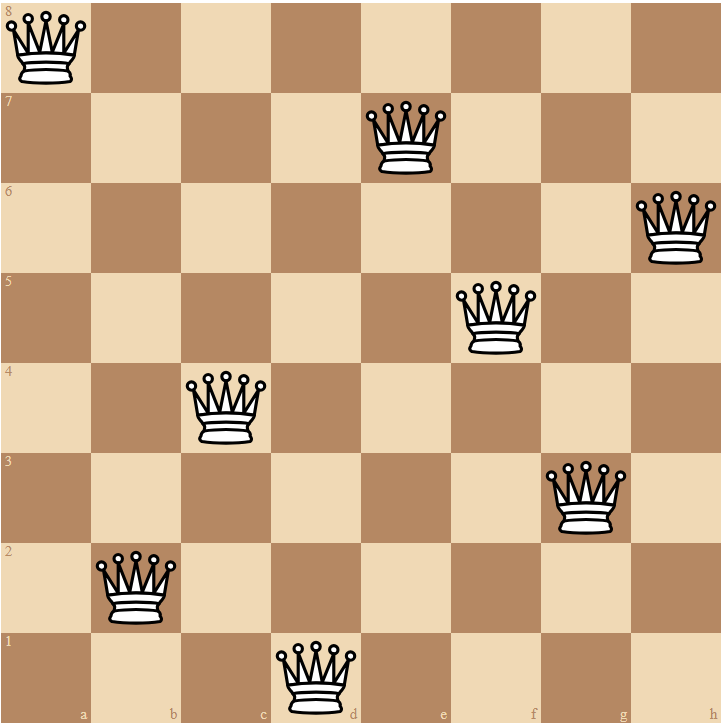
**Анализ задачи**

1. int board[8][8] - создает двумерный массив с именем board размером 8x8. Массив используется для хранения информации о расположении ферзей на шахматной доске.

2. Функции reQueen и setQueen служат для удаления и добавления ферзей на доску соответственно.

reQueen принимает два параметра: координаты ячейки, из которой нужно удалить ферзя. Затем функция проходит по всем ячейкам доски и уменьшает их значения на 1, кроме ячейки, из которой был удален ферзь, которой присваивается значение 0.

setQueen работает аналогично, только вместо уменьшения значений ячеек она увеличивает их на 1. Эта функция также принимает два параметра - координаты ячейки, куда нужно поместить нового ферзя.

****

**Код**

#include <iostream>

using namespace std;

int board[8][8];

void reQueen(int i, int j)

{

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

{

--board[x][j]; // -горизонталь

--board[i][x]; // -диагональ

int k;

k = j - i + x; // диагональ №1

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

--board[x][k];

k = j + i - x; // диагональ №2

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

--board[x][k];

}

board[i][j] = 0; // пустая клетка

}

void setQueen(int i, int j)

{

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

{

++board[x][j]; // +горизонталь

++board[i][x]; // +диагональ

int k;

k = j - i + x; // диагональ №1

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

++board[x][k];

k = j + i - x; // диагональ №2

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

++board[x][k];

}

board[i][j] = -1; // ферзь стоит

}

bool tryQueen(int i) // попытка поставить ферзя

{

bool result = false;

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

{

if (board[i][j] == 0) // если клетку не бьют

{

setQueen(i, j); // ставим ферзя

if (i == 7) // если все ферзи стоят

result = true; // попытка удалась

else

{

if (!(result = tryQueen(i + 1))) // иначе ставим на следующий столбец

reQueen(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; // пустая доска

tryQueen(0); // начинаем с первой клетки

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

{

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

{

if (board[i][j] == -1) // на клетке ферзь

cout << " " << (char)3 << " ";

else

cout << " . "; // пустая клетка

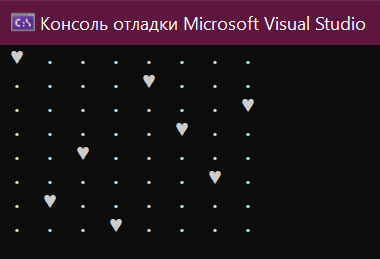
}

cout << endl;

}

}

**Работа кода**

****

**Блок-схема**

