Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Лабораторная работа о 8 ферзях**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**(Семестр** 2)

Выполнил студент гр. ИВТ-21-1б

Гребнев Алексей Дмитриевич

Проверил:

Яруллин Денис Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г.Пермь -2022

**Постановка задачи**

Реализовать программу, в которой реализуется алгоритм расстановки 8 ферзей на доске 8x8 так, чтобы ферзи были расставлены в каждой строке и не «били» друг друга.

**Исходные данные**

1. Создаем двумерный массив 8 на 8, который будет служить нам как шахматная доска;
2. Создаем функцию, которая будет отвечать за передвижение королевы на незанятые ячейки и проверку на то, чтобы она не срубила другую королеву;
3. И создаем третью функцию, которая будет отвечать за само расположение королевы на шахматной доске.

#include <iostream>

using namespace std;

int board[8][8];

void SetQueen(int i, int j);

void ResetQueen(int i, int j);

bool tryQueen(int i);

int main()

{

for (int i = 0; i <= 7; ++i)

{

for (int j = 0; j <= 7; j++)

{

board[i][j] = 0;

}

}

tryQueen(0);

for (int i = 0; i <= 7; ++i)

{

for (int j = 0; j <= 7; ++j)

{

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

{

cout << "Q ";

}

else

{

cout << ". ";

}

}

cout << "\n";

}

}

void SetQueen(int i, int j)

{

for (int x = 0; x <= 7; ++x)

{

++board[x][j];

++board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7)

{

++board[x][foo];

}

foo = j + i - x;

if (foo >= 0 && foo <= 7)

{

++board[x][foo];

}

}

board[i][j] = -1;

}

void ResetQueen(int i, int j)

{

for (int x = 0; x <= 7; ++x)

{

--board[x][j];

--board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7)

{

--board[x][foo];

}

foo = j + i - x;

if (foo >= 0 && foo <= 7)

{

--board[x][foo];

}

}

board[i][j] = 0;

}

bool tryQueen(int i)

{

bool result = false;

for (int j = 0; j <= 7; j++)

{

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

{

SetQueen(i, j);

if (i == 7)

{

result = true;

}

else

{

result = tryQueen(i + 1);

if (!result)

{

ResetQueen(i, j);

}

}

}

}

return result;

}









