Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

Лабораторная работа по теме

“Восемь ферзей.”

Выполнил: Студент группы РИС-23-3б

Асташин Дмитрий Алексеевич

Проверила: Доцент кафедры ИТАС

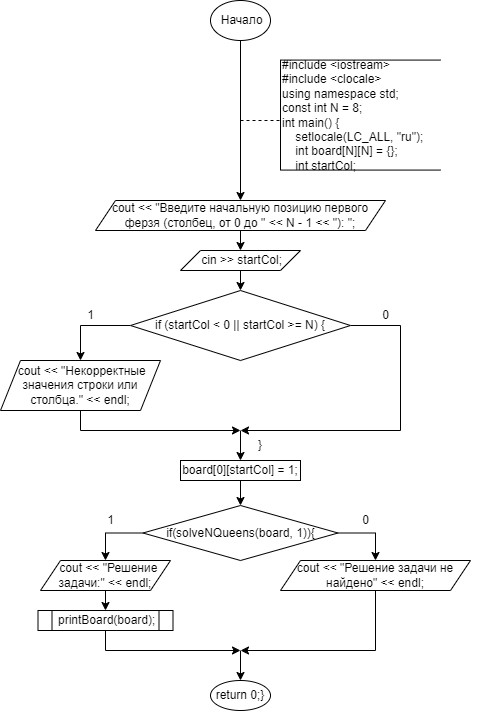
О. А. Полякова

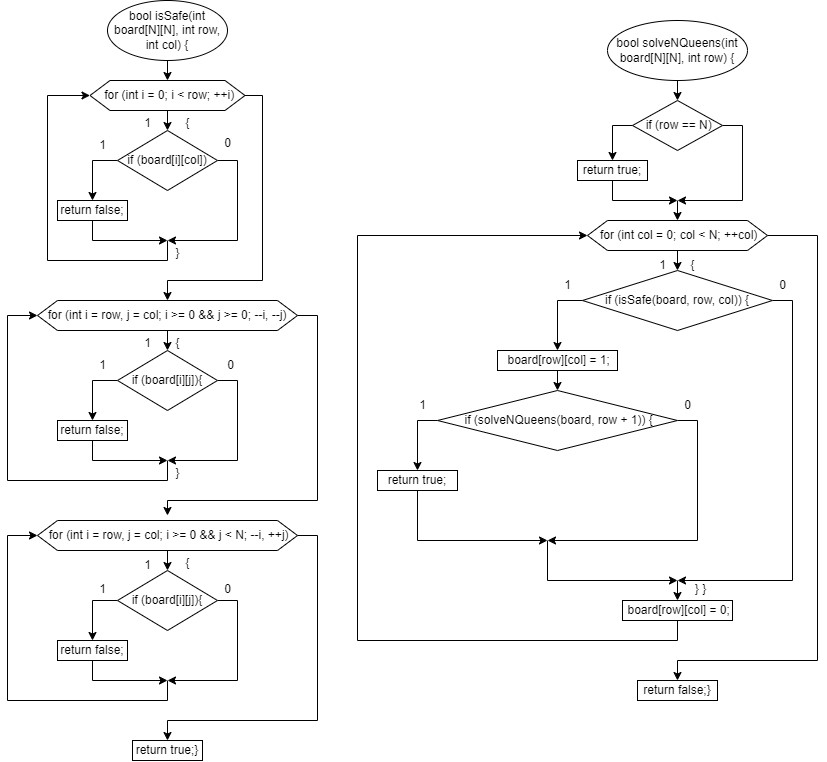
2024

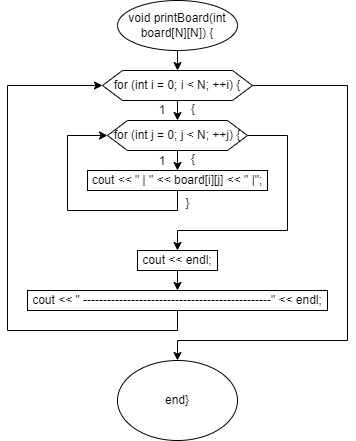
1. Анализ:

* Доска – двумерный массив(8 строк, 8 столбцов), заполненный нулями;
* Функция printBoard() – рисует доску перебором всех элементов двумерного массива с помощью вложенных циклов;
* Функция isSafe() проверяет можно ли поместить ферзя в текущую позицию. Проверка идет по вертикали, и двум диагоналям. Если Будет найдена единица(ферзь), то функция выводит false, иначе true;
* Рекурсивная функция solveNQueens() расставляет ферзей на доске. С помощью цикла for ставим ферзя в каждую строку, проверяем позицию с помощью isSafe() и отрисовываем доску.

1. Блок-схема:







1. Код:

#include <iostream>

#include <clocale>

using namespace std;

const int N = 8; // Размер доски

// Функция для отображения доски с расположением ферзей

void printBoard(int board[N][N]) {

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

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

cout << " | " << board[i][j] << " |";

}

cout << endl;

cout << " -----------------------------------------------" << endl;

}

}

// Функция для проверки, можно ли поместить ферзя в указанную позицию

bool isSafe(int board[N][N], int row, int col) {

// Проверяем вертикали

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

if (board[i][col]) {

return false;

}

// Проверяем верхнюю левую диагональ

for (int i = row, j = col; i >= 0 && j >= 0; --i, --j)

if (board[i][j]) {

return false;

}

// Проверяем верхнюю правую диагональ

for (int i = row, j = col; i >= 0 && j < N; --i, ++j)

if (board[i][j]) {

return false;

}

return true;

}

// Рекурсивная функция

bool solveNQueens(int board[N][N], int row) {

// Если все ферзи были размещены, возвращаем true

if (row == N) {

return true;

}

// Пытаемся поместить ферзя в каждый столбец текущей строки

for (int col = 0; col < N; ++col) {

// Проверяем, можно ли поместить ферзя в эту позицию

if (isSafe(board, row, col)) {

// Помещаем ферзя в текущую позицию

board[row][col] = 1;

printBoard(board);

cout << endl << endl;

// Рекурсивно размещаем оставшихся ферзей

if (solveNQueens(board, row + 1)) {

return true;

}

// Если размещение ферзя в текущей позиции не приводит к решению, отменяем изменения

board[row][col] = 0;

}

}

// Если не удалось разместить ферзя в текущей строке, возвращаем false

return false;

}

int main() {

setlocale(LC\_ALL, "ru");

int board[N][N] = {}; // Инициализируем пустую доску

int startCol;

cout << "Введите начальную позицию первого ферзя (столбец, от 0 до " << N - 1 << "): ";

cin >> startCol;

// Проверяем, введены ли корректные значения

if (startCol < 0 || startCol >= N) {

cout << "Некорректные значения строки или столбца." << endl;

}

// Размещаем первого ферзя на доске

board[0][startCol] = 1;

// Решаем задачу о восьми ферзях

if (solveNQueens(board, 1)) { // Начинаем со второй строки

cout << "Решение задачи:" << endl;

printBoard(board);

}

else {

cout << "Решение задачи не найдено" << endl;

}

return 0;

}