Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«**Информационные технологии и автоматизированные системы**»**

направление подготовки: 09.03.04 – **«**Программная инженерия»

**Лабораторная работа №3.**

**“Ханойская башня. Задача о восьми ферзях”  
Вариант 10**

Выполнил студент гр. РИС-24-2б

Бартов Игорь Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Ханойская башня**

1. **Постановка задачи:**

Имеется три стержня и несколько дисков разного диаметра, надетых на первый стержень. Требуется переместить все диски на третий стержень, соблюдая следующие правила:

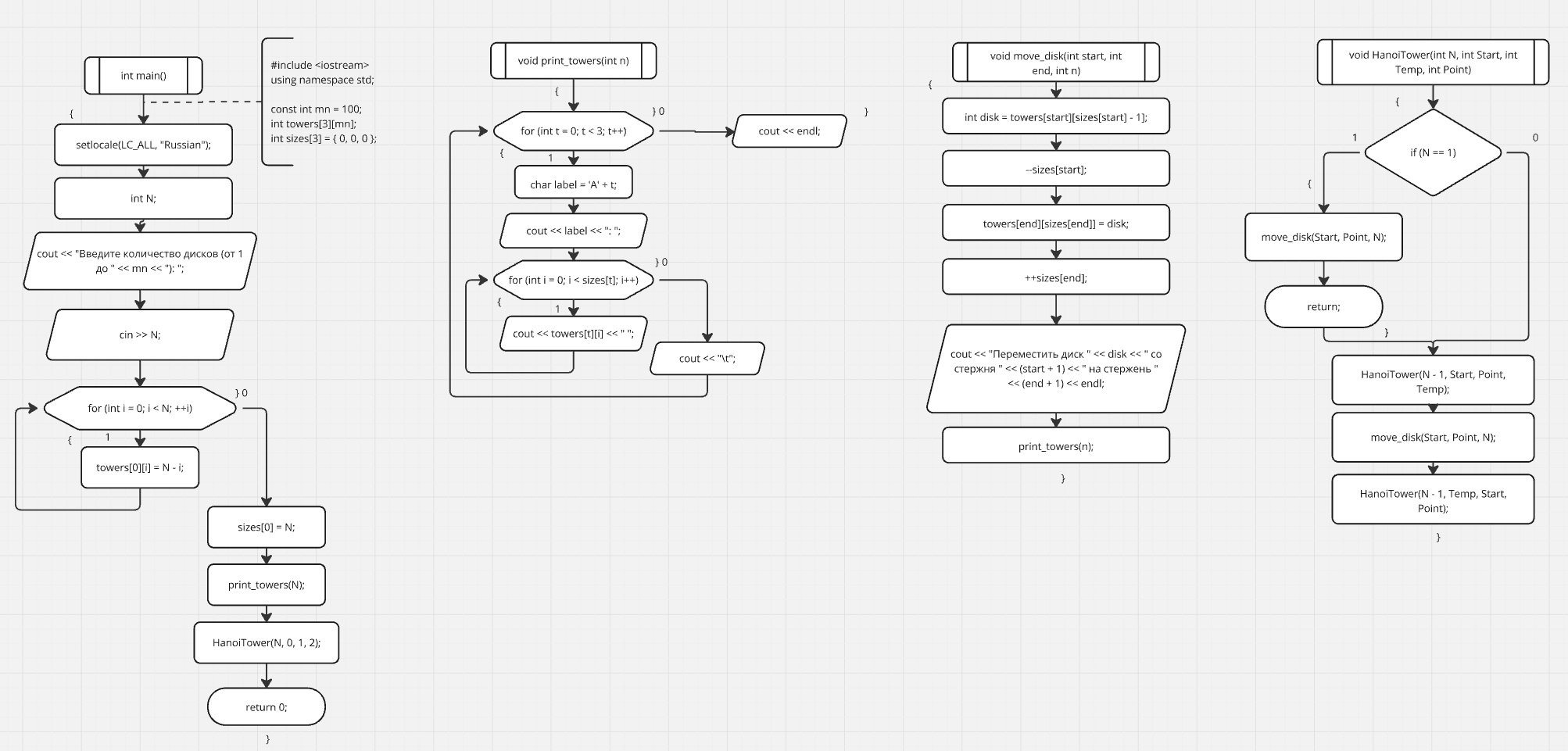
1. За один ход можно перемещать только один диск.
2. Диск можно положить только на пустой стержень или на диск большего размера.
3. Для переноса дисков разрешено использовать промежуточный (второй) стержень.

Необходимо реализовать алгоритм, позволяющий выполнить перенос за минимальное количество шагов.

### Анализ решения:

Решение задачи основывается на рекурсивном алгоритме:

1. Переносим n−1n-1n−1 верхних дисков с первого стержня на вспомогательный (второй), используя третий как промежуточный.
2. Перемещаем самый большой диск с первого стержня на третий.
3. Переносим n−1n-1n−1 дисков со второго стержня на третий, используя первый в качестве вспомогательного.
4. **Блок-схема:**



1. **Код:**  
   #include <iostream>

using namespace std;

const int mn = 100;

int towers[3][mn];

int sizes[3] = { 0, 0, 0 };

void print\_towers(int n)

{

cout << "Положение башен:" << endl;

for (int t = 0; t < 3; t++)

{

char label = 'A' + t;

cout << label << ": ";

for (int i = 0; i < sizes[t]; i++)

{

cout << towers[t][i] << " ";

}

cout << "\t";

}

cout << endl;

}

void move\_disk(int start, int end, int n)

{

int disk = towers[start][sizes[start] - 1];

--sizes[start];

towers[end][sizes[end]] = disk;

++sizes[end];

cout << "Переместить диск " << disk << " со стержня " << (start + 1) << " на стержень " << (end + 1) << endl;

print\_towers(n);

}

void HanoiTower(int N, int Start, int Temp, int Point)

{

if (N == 1)

{

move\_disk(Start, Point, N);

return;

}

HanoiTower(N - 1, Start, Point, Temp);

move\_disk(Start, Point, N);

HanoiTower(N - 1, Temp, Start, Point);

}

int main()

{

setlocale(LC\_ALL, "Russian");

int N;

cout << "Введите количество дисков (от 1 до " << mn << "): ";

cin >> N;

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

{

towers[0][i] = N - i;

}

sizes[0] = N;

print\_towers(N);

HanoiTower(N, 0, 1, 2);

return 0;

}

**Задача о восьми ферзях**

1. **Постановка задачи:**

Необходимо разместить 8 ферзей на шахматной доске размером 8×8 таким образом, чтобы ни один ферзь не угрожал другому. Это значит, что на каждой горизонтали, вертикали и диагонали должен находиться только один ферзь.

1. **Анализ решения:**

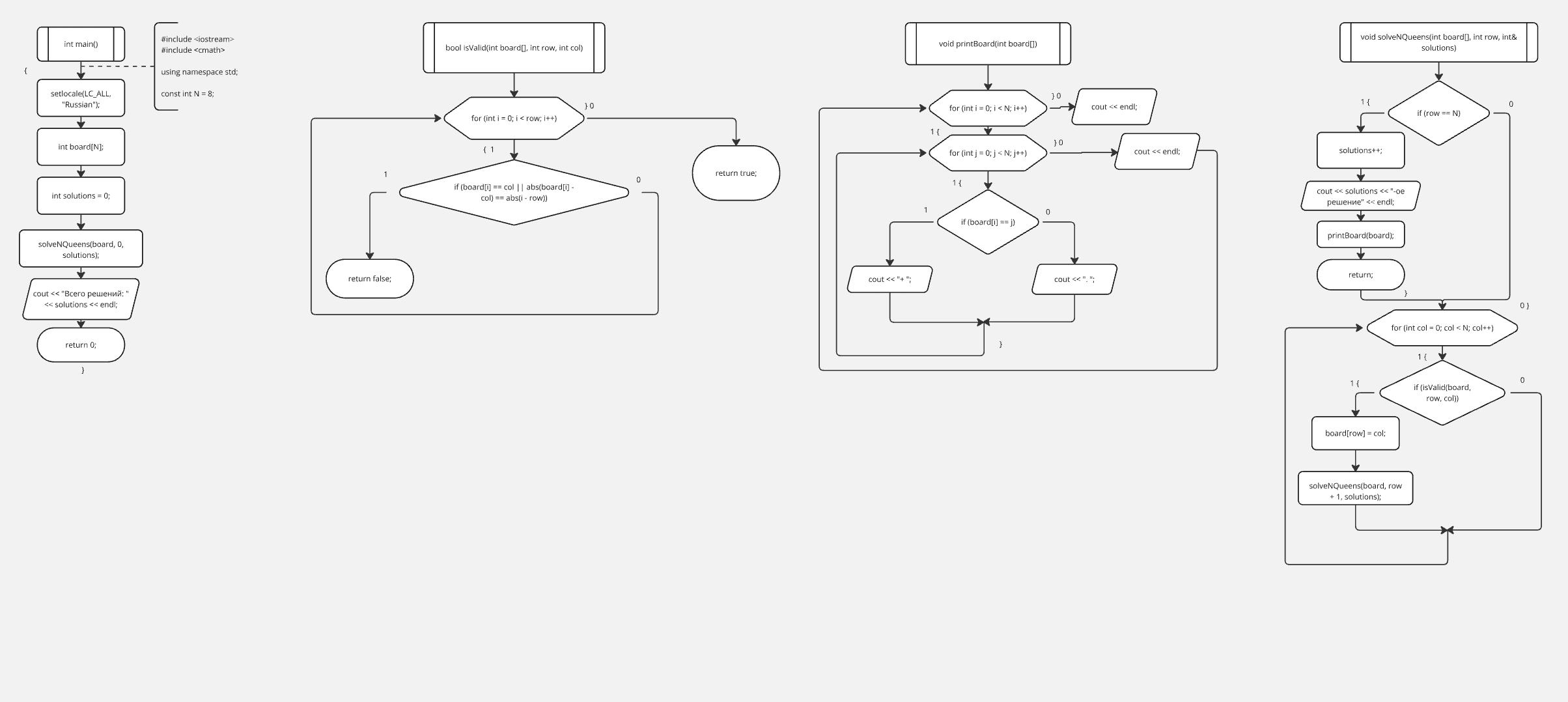
Так как ферзи могут атаковать по горизонтали, вертикали и диагонали, то при расстановке их на доске необходимо учитывать следующие условия:

1. В каждой строке должен стоять только один ферзь.
2. В каждом столбце должен стоять только один ферзь.
3. Ни один ферзь не должен находиться на одной диагонали с другим.

**3. Алгоритм решения:**

1. Размещаем первого ферзя в первой строке в первой доступной ячейке.
2. Переходим на следующую строку и пытаемся разместить там ферзя так, чтобы он не атаковал предыдущих.
3. Если невозможно безопасно разместить ферзя в текущей строке, возвращаемся на предыдущую строку и передвигаем последнего размещенного ферзя на следующую возможную позицию.
4. Повторяем процесс до тех пор, пока не расставим всех 8 ферзей.

**4. Блок-схема:**



1. **Код:**

#include <iostream>

#include <cmath>

using namespace std;

const int N = 8;

bool isValid(int board[], int row, int col) {

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

if (board[i] == col || abs(board[i] - col) == abs(i - row)) {

return false;

}

}

return true;

}

void printBoard(int board[]) {

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

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

if (board[i] == j)

cout << "+ ";

else

cout << ". ";

}

cout << endl;

}

cout << endl;

}

void solveNQueens(int board[], int row, int& solutions) {

if (row == N) {

solutions++;

cout << solutions << "-ое решение" << endl;

printBoard(board);

return;

}

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

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

board[row] = col;

solveNQueens(board, row + 1, solutions);

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int board[N];

int solutions = 0;

solveNQueens(board, 0, solutions);

cout << "Всего решений: " << solutions << endl;

return 0;

}