Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**Отчет по лабораторной работе**

**«Ханойская башня»**

|  |
| --- |
| Выполнила студентка гр. ИВТ-24-2б  Косиненко Ксения Николаевна \_\_\_\_\_\_ |
| Проверил:  Доц. Каф. ИТАС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ольга Андреевна Полякова\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка) (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

г. Пермь, 2025

**Реализовать Ханойскую башню на C ++**

**Формулировка задачи:**

Башня Ханоя — это математическая головоломка, в которой у нас есть 3 стержня и 4 диска. Сначала все диски располагаются на первом стержне от большего диска к меньшему. Нам нужно переместить все диски от первого диска к последнему в том же порядке от большего к меньшему.

В Ханойской башне действуют следующие правила:

В один момент времени может быть перемещен только один диск.

Диск меньшего размера можно разместить над диском большего размера, но диск большего размера нельзя разместить над диском меньшего размера.

**Код программы:**

#include <iostream>

#include <clocale>

using namespace std;

const int MAX\_DiskKS = 10; // Максимальное количество дисков

void printState(int DiskksOnBar[3][MAX\_DiskKS])

{

cout << "Текущее состояние стержней:" << endl;

for (int i = 0; i < 3; ++i) {

cout << "Стержень " << (i + 1) << ": ";

for (int j = 0; j < MAX\_DiskKS; ++j) {

if (DiskksOnBar[i][j] != 0) {

cout << DiskksOnBar[i][j] << " ";

}

}

cout << endl;

}

cout << endl;

}

void hanoiTower(int n, int start, int finish, int help, int DiskksOnBar[3][MAX\_DiskKS])

{

if (n > 0)

{

hanoiTower(n - 1, start, help, finish, DiskksOnBar);

// Перемещение диска

// Находим первый свободный индекс на целевом стержне

int finishIndex = MAX\_DiskKS - 1;

while (finishIndex >= 0 && DiskksOnBar[finish - 1][finishIndex] != 0)

{

finishIndex--;

}

// Находим индекс диска на исходном стержне

int startIndex = 0;

while (startIndex < MAX\_DiskKS && DiskksOnBar[start - 1][startIndex] != n)

{

startIndex++;

}

if (startIndex < MAX\_DiskKS)

{ //Проверка на случай, если диск не найден (чтобы избежать ошибок)

// Перемещаем диск

DiskksOnBar[finish - 1][finishIndex] = n; // Помещаем диск на целевой стержень

DiskksOnBar[start - 1][startIndex] = 0; // Убираем диск с исходного стержня

cout << "Переместить диск " << n << " с стержня " << start << " на стержень " << finish << endl;

printState(DiskksOnBar); // Печатаем текущее состояние стержней

hanoiTower(n - 1, help, finish, start, DiskksOnBar);

}

else

{

cerr << "Ошибка: Диск " << n << " не найден на стержне " << start << endl;

}

}

}

int main()

{

int Disk;

setlocale(LC\_ALL, "ru");

cout << "Введите количество дисков: ";

cin >> Disk;

// Инициализация состояния стержней

int DiskksOnBar[3][MAX\_DiskKS] = { 0 }; // 0 означает, что диск отсутствует

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

{

DiskksOnBar[0][MAX\_DiskKS - 1 - i] = i + 1; // Заполняем первый стержень дисками

}

cout << "Решение Ханойской башни с " << Disk << " дисками:" << endl;

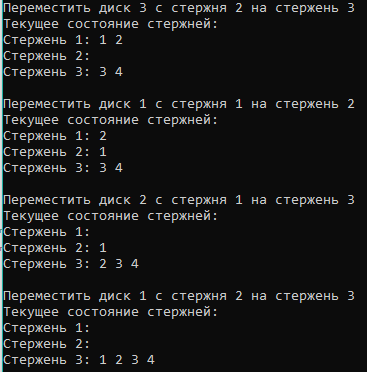
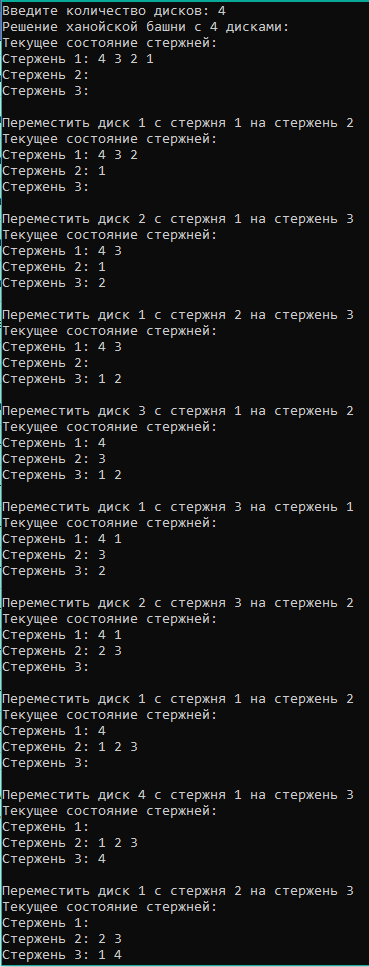
printState(DiskksOnBar); // Печатаем начальное состояние

hanoiTower(Disk, 1, 3, 2, DiskksOnBar); // Перемещение дисков с стержня 1 на стержень 3, используя стержень 2 в качестве вспомогательного

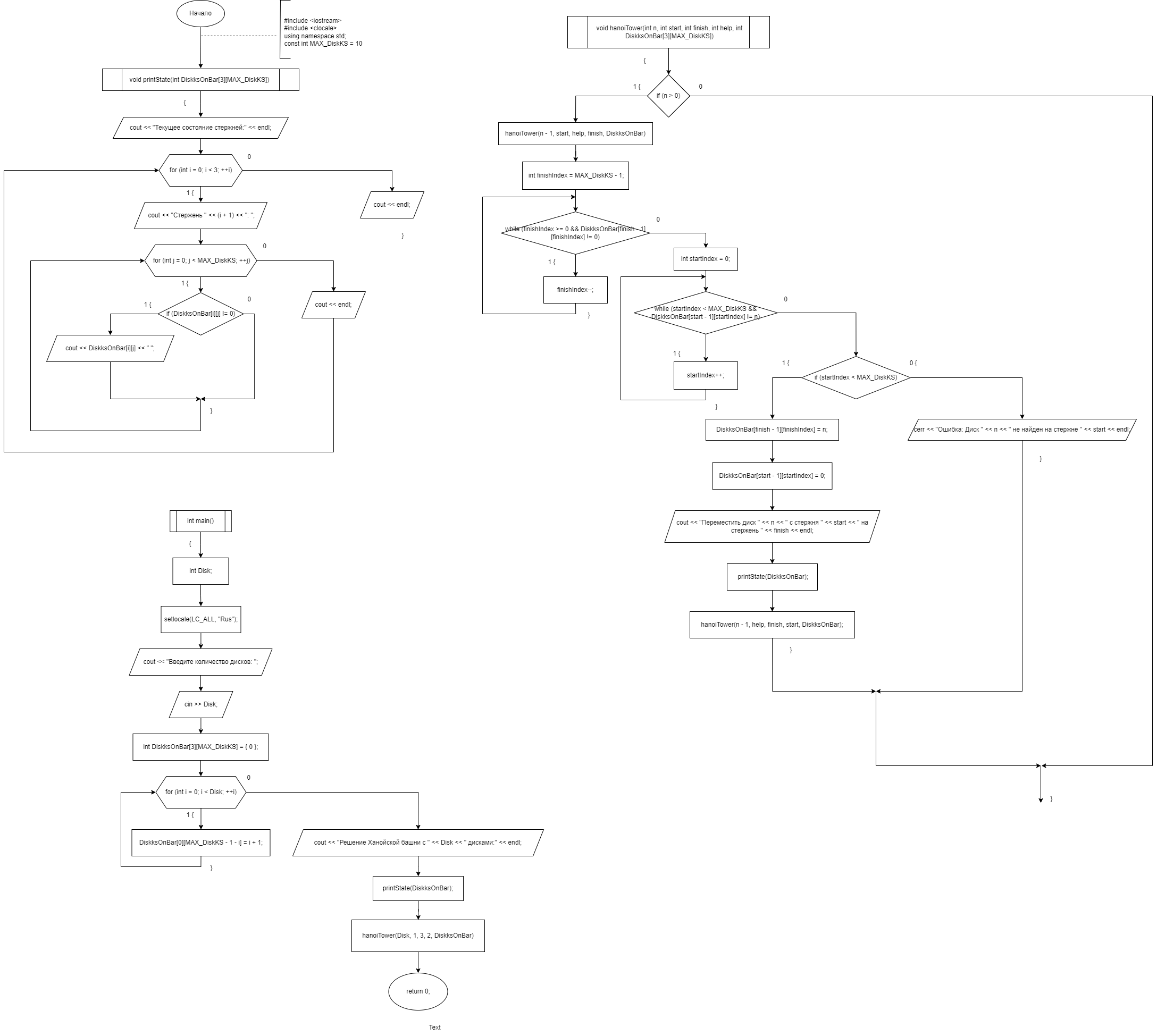
return 0;

}

**Вывод программы:**



**Блок-схема кода программы:**



**GitHub**

<https://github.com/kitmammaw/->

