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: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа № 2**

**«Задачи на рекурсию. Ханойская башня. 8 ферзей.»**

Выполнил студент гр. ИВТ-24-1б

Оглезнев Никита Михайлович

Проверил:

Доцент кафедры ИТАС

Д.В.Яруллин

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Ханойская башня**

1. **Постановка задачи**

Даны 1 стержень с дисками разного размера и 2 пустых стержня. Нужно переместить диски с одного стержня на другой, перекладывать можно только по одному диску за ход, складывать диски можно только меньший на больший. Реализовать программу, определяющую перестановки этих дисков с помощью наименьшего количества ходов.

Условие:

1. Даны 3 стержня;
2. Первый с n количеством дисков разного размера, в порядке возрастания сверху вниз, остальные – пустые;
3. Необходимо переложить все диски с первого стержня на другой, используя минимальное количество перестановок, учитывая, что меньший диск можно укладывать только на диск большего размера
4. В программной реализации обязателен вывод каждого хода.

Ход решения:

1. Входные данные – количество дисков на начальном стержне;
2. Условие выхода из рекурсии – когда количество дисков равно 0;
3. Шаг рекурсии: перенос n-1 дисков с начального стержня на временный, конечный стержень используется как вспомогательный; переносим самый большой диск на конечный стержень, переносим n-1 дисков с временного стержня на конечный, используя начальный стержень как вспомогательный.
4. **Блок-схема решения**

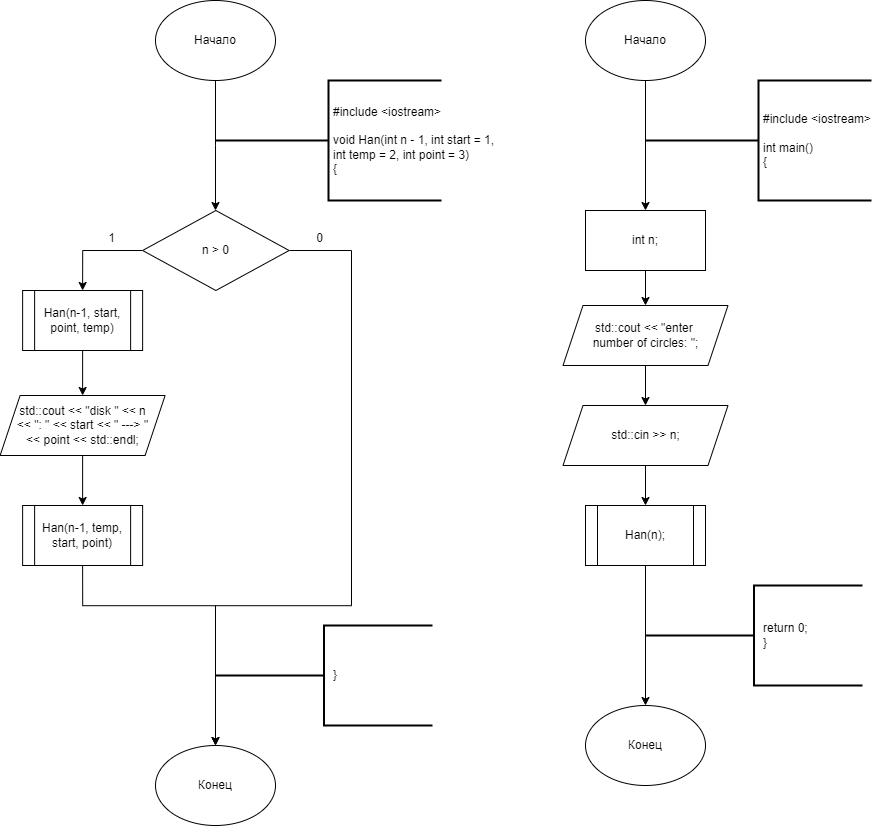
****

Рисунок 1 – Блок-схема решения задачи

1. **Код решения задачи**

#include <iostream>

void Han(int n, int start = 1, int temp = 2, int point = 3)

{

if (n > 0)

{

Han(n - 1, start, point, temp);

std::cout << "disk " << n << ": " << start << " ---> " << point << std::endl;

Han(n - 1, temp, start, point);

}

}

int main()

{

int n;

std::cout << "enter count of circles: ";

std::cin >> n;

Han(n);

return 0;

}

1. **Результат выполнения кода**

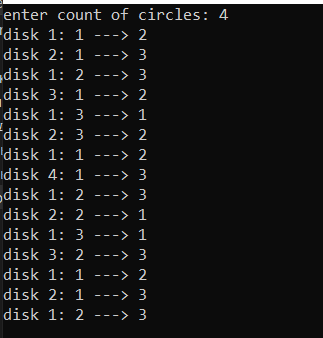
****

Рисунок 2 – Результат выполнения кода для 4-х дисков

**8 ферзей**

1. **Постановка задачи**

Реализовать алгоритм расстановки 8 ферзей на шахматной доске размером 8х8 клеток так, чтобы ферзи были расставлены в каждой строке и не «били» друг друга.

Ход решения:

1. Шахматная доска представлена двумерным массивом, пустая ячейка – 0, ячейка с ферзем – 1;
2. Рекурсия осуществлена в функции set(), входными данными является номер строки массива, начиная с 0;
3. Если номер итерируемой строки больше индекса последней массива (7), текущий массив выводится на экран, затем обнуляется;
4. Шаг рекурсии: для каждого столбца ii в текущей строке I проверяется, можно ли безопасно разместить ферзя в ячейке (i, ii), в случае успешной проверки ячейки ферзь ставится на доску, затем рекурсия обращается к следующей строке, иначе – функция переходит к следующему столбцу;
5. Проверка ячейки осуществляется от ячейки по столбцу, вправо и влево по диагонали в направлении вверх, т.к. ферзи ставятся построчно и нижние строки пустые.
6. **Блок-схема решения задачи**

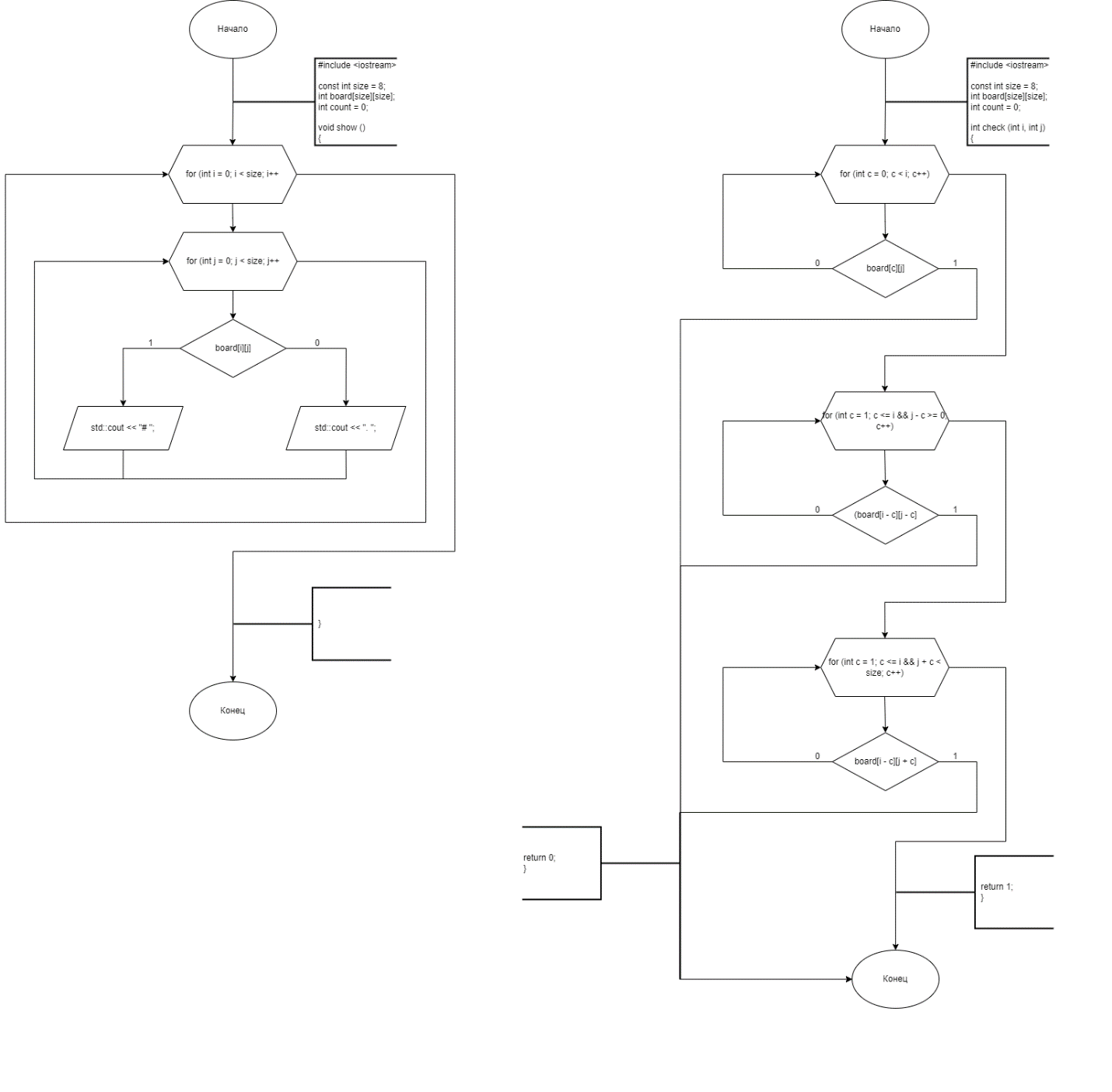
****

Рисунок 3 – Блок-схема решения задачи (начало)

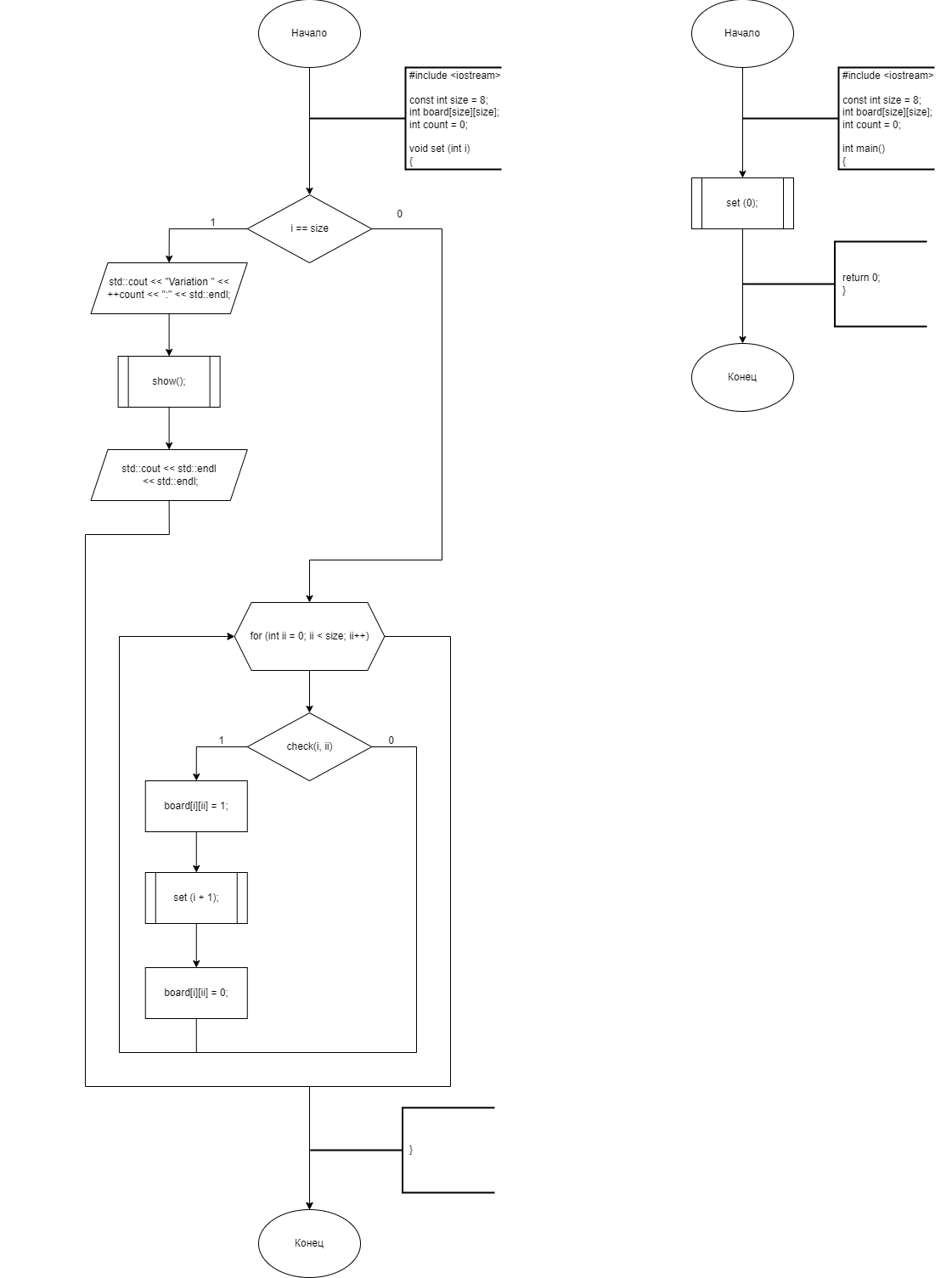


Рисунок 4 – Блок-схема решения задачи (конец)

1. **Код решения задачи**

#include <iostream>

const int size = 8;

int board[size][size];

int count = 0;

void show() // вывод доски

{

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

{

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

{

if (board[i][j]) { std::cout << "# "; }

else { std::cout << ". "; }

}

std::cout << std::endl;

}

}

int check(int i, int j) // бьется ли ячейка

{

for (int c = 0; c < i; c++) // проверка столбца

{

if (board[c][j]) { return 0; }

}

for (int c = 1; c <= i && j - c >= 0; c++) // проверка диагонали влево вверх

{

if (board[i - c][j - c]) { return 0; }

}

for (int c = 1; c <= i && j + c < size; c++) // проверка диагонали вправо вверх

{

if (board[i - c][j + c]) { return 0; }

}

return 1;

}

void set(int i) // i - номер строки

{

if (i == size)

{

std::cout << "Variation " << ++count << ":" << std::endl;

show();

std::cout << std::endl << std::endl;

return;

}

for (int ii = 0; ii < size; ii++)

{

if (check(i, ii)) // проверка строки

{

board[i][ii] = 1;

set(i + 1);

board[i][ii] = 0; // обнуление доски

}

// else - откат к предыдущей строке со смещением ферзя

}

}

int main()

{

set(0);

return 0;

}

1. **Результат выполнения кода**

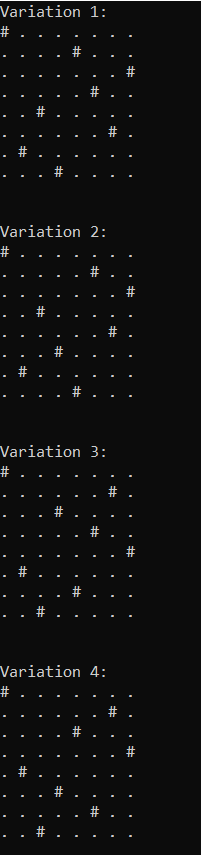


Рисунок 5 – Начало выполнения кода

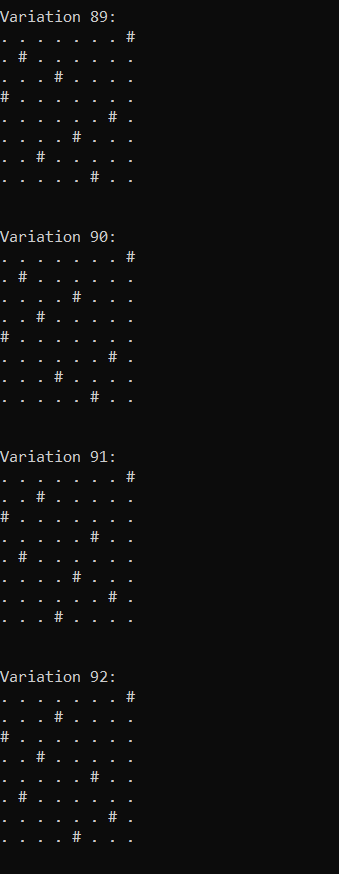


Рисунок 6 – Конец выполнения кода