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.**

**«Ханойские башни и задача про 8 ферзей»**

Выполнил студент гр. РИС-24-2б

Трофимов Степан Степанович

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Ханойские башни**

**Анализ задачи:**

Задача о Ханойских башнях заключается в том, чтобы переместить n дисков с одного стержня на другой, используя третий стержень как вспомогательный. При этом соблюдаются следующие правила:

* Перемещать можно только один диск за раз.
* Диск нельзя класть на диск меньшего размера.
* Все диски должны в конечном итоге оказаться на целевом стержне в правильном порядке (от большего внизу к меньшему наверху).

**Объявление переменных**

* const int mn = 100; – Максимальное количество дисков (100).
* int towers[3][mn]; – Двумерный массив для хранения состояния трёх башен.
* int sizes[3] = {0, 0, 0}; – Массив, хранящий количество дисков на каждом из трёх стержней.

**Функция print\_towers(int n)**

Выводит текущее состояние башен в консоль. Каждая башня (A, B, C) представлена в виде списка дисков, упорядоченных от нижнего к верхнему.

**Функция move\_disk(int start, int end, int n)**

* Перемещает верхний диск со стержня start на стержень end.
* Уменьшает размер массива sizes[start] (удаляет диск из начальной башни).
* Добавляет диск в массив towers[end], увеличивая sizes[end].
* Выводит информацию о перемещении.

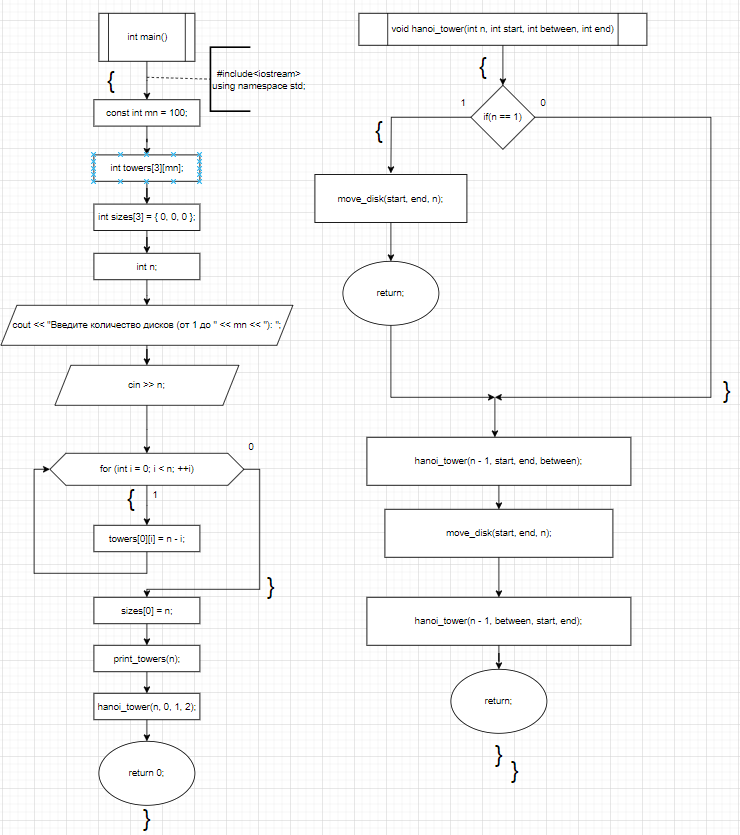
**Функция hanoi\_tower(int n, int start, int between, int end)**

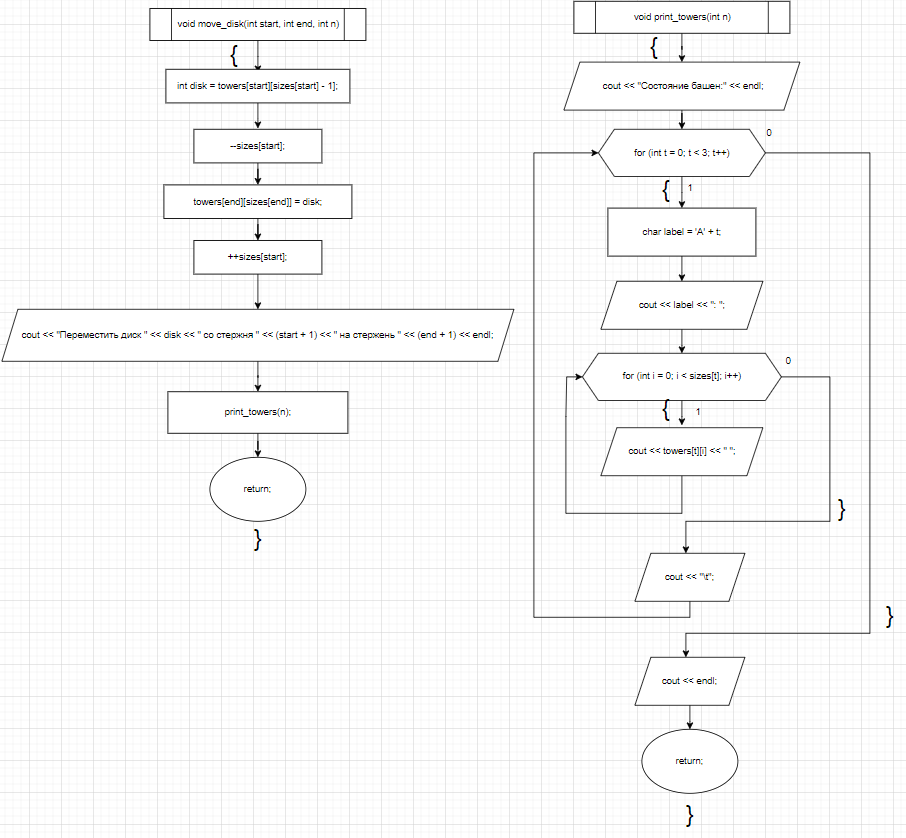
Рекурсивно решает задачу по следующему алгоритму:

1. Перемещает n-1 дисков со start на between, используя end как вспомогательный.
2. Перемещает оставшийся (самый большой) диск напрямую со start на end.
3. Перемещает n-1 дисков с between на end, используя start как вспомогательный.

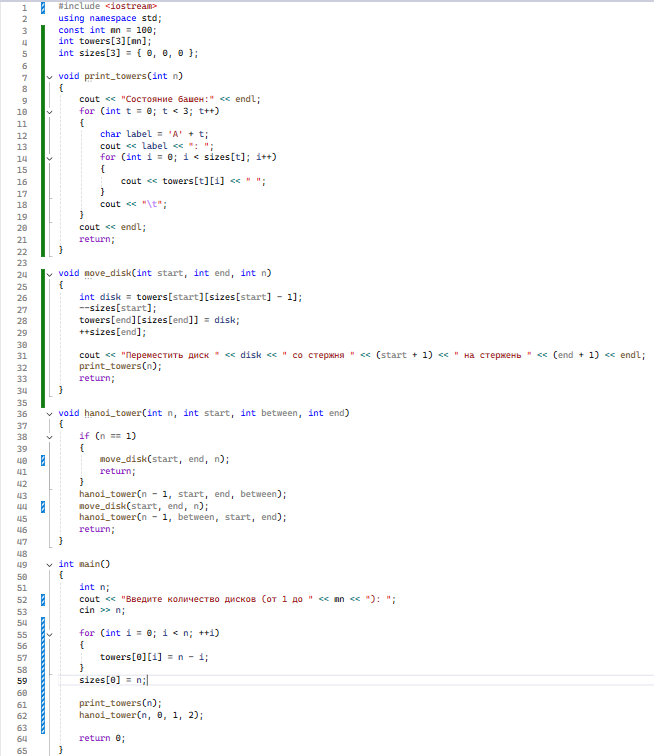
**Функция main()**

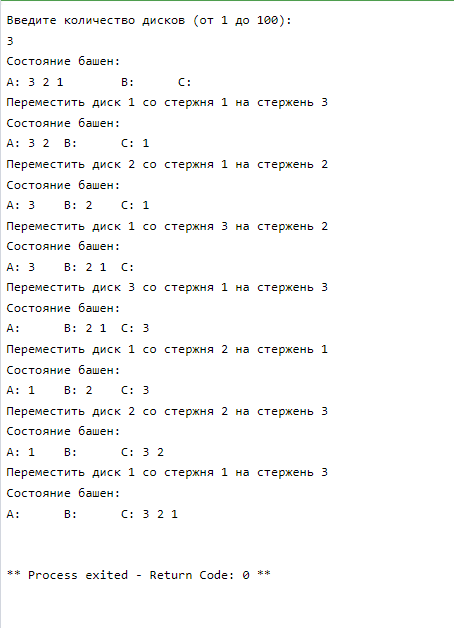
1. Запрашивает у пользователя количество дисков n.
2. Заполняет массив towers[0] числами от n до 1 (самый большой диск снизу, самый маленький сверху).
3. Вызывает print\_towers(n) для отображения начального состояния.
4. Запускает рекурсивную функцию hanoi\_tower(n, 0, 1, 2), решающую задачу.
5. Программа завершается.

**Блок-схема**

****

**Код с выводом**

****

****

**Задача про 8 ферзей**

**Анализ задачи**

Задача состоит в том, чтобы разместить 8 ферзей на шахматной доске 8×8 таким образом, чтобы они не били друг друга. Это означает, что:

* На каждой строке и в каждом столбце может быть только один ферзь.
* Ферзи не должны находиться на одной диагонали.

Цель программы – найти и вывести все возможные расстановки 8 ферзей, а также посчитать их количество.

**Способ представления доски**

* Доска представляется массивом board[n], где board[i] указывает на столбец, в котором стоит ферзь в строке i.
* Например, board[0] = 3 означает, что в первой строке ферзь стоит в 4-м столбце.

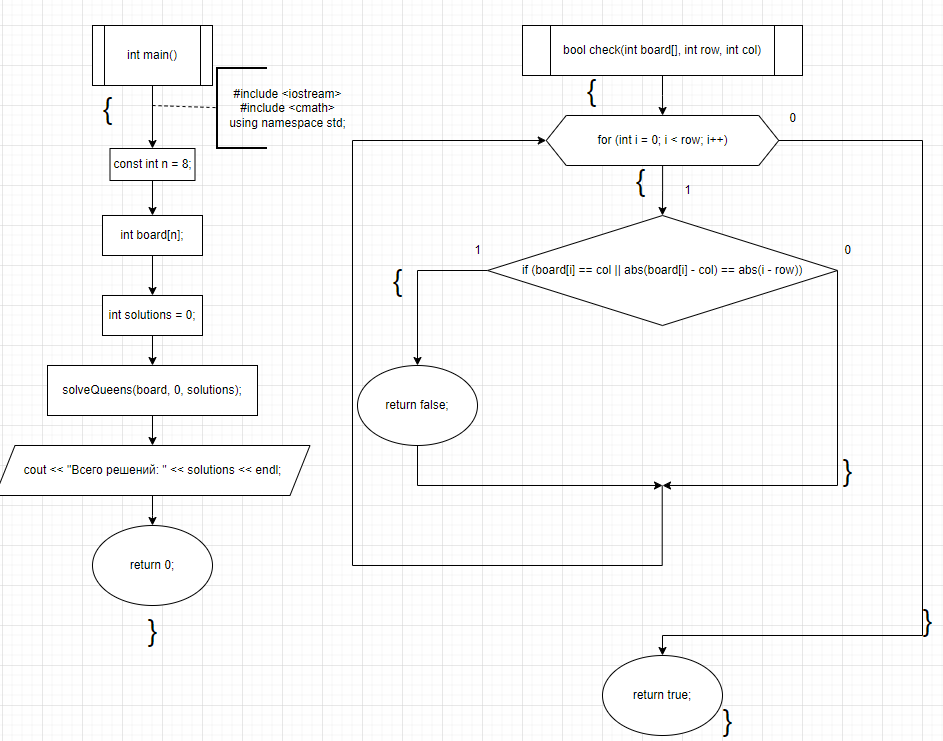
**Основные функции**

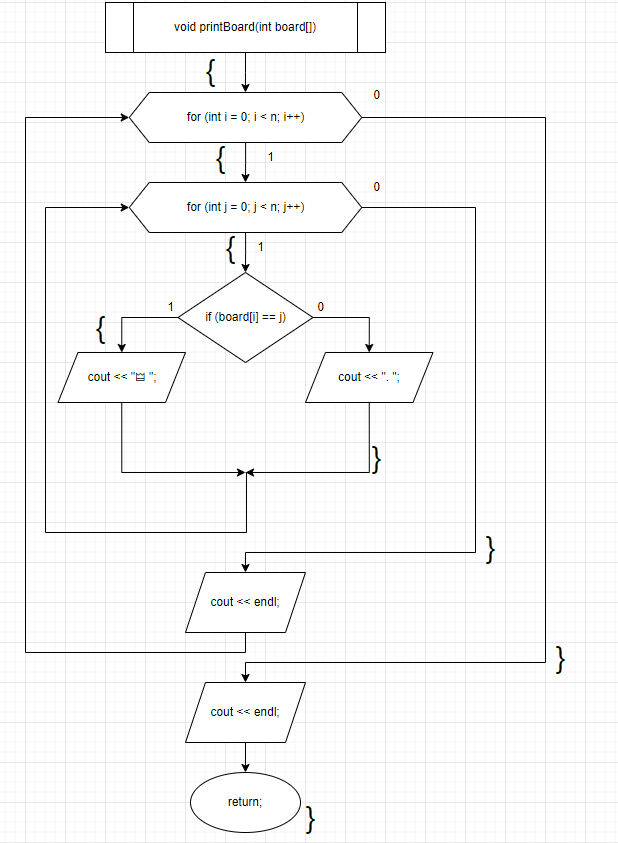
1. check(board, row, col) – проверяет, можно ли поставить ферзя в row, col.
   * Проверяется, нет ли ферзя в том же столбце.
   * Проверяется, нет ли ферзя на диагоналях.
2. solveQueens(board, row, solutions) – рекурсивно размещает ферзей:
   * Если row == n, значит, удалось расставить всех 8 ферзей, и решение выводится.
   * Перебираются все столбцы, если ферзя можно поставить (check()), ставим его и рекурсивно переходим к следующей строке.
3. printBoard(board) – печатает текущее решение, заменяя ферзей символом короны.

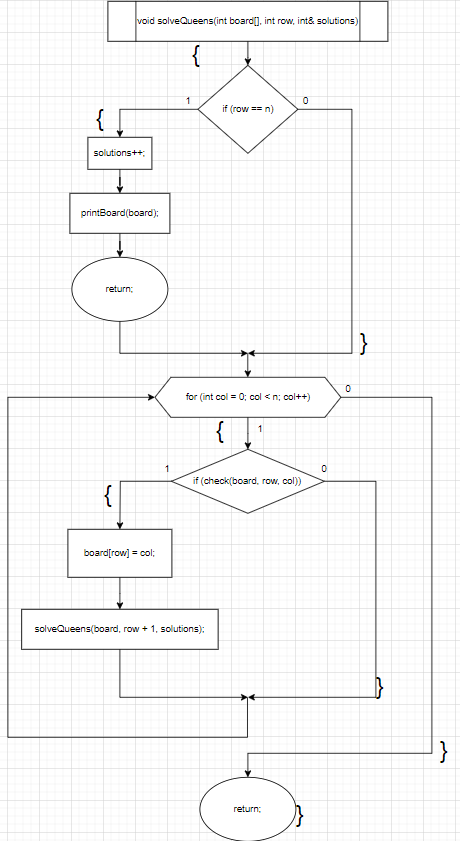
**Основная логика**

* Начинаем с row = 0 и пытаемся поставить ферзя в первый возможный столбец.
* Если удаётся поставить ферзя, переходим к следующей строке.
* Если для строки не нашлось подходящего столбца, возвращаемся на предыдущую строку и пробуем другой вариант.
* Так перебираются все возможные комбинации, находя все правильные расстановки ферзей.

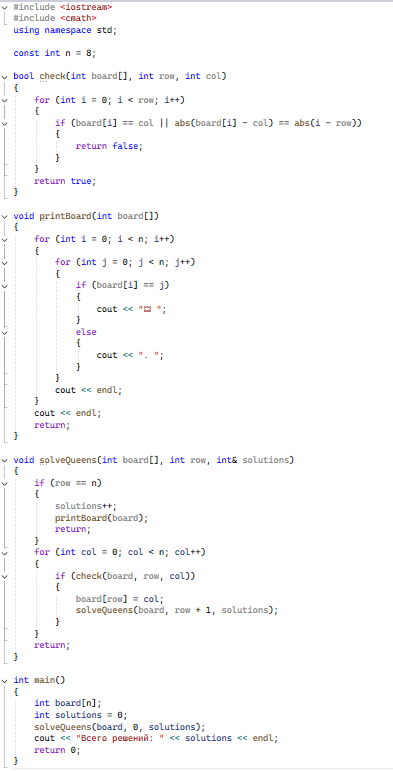
**Блок-схема**

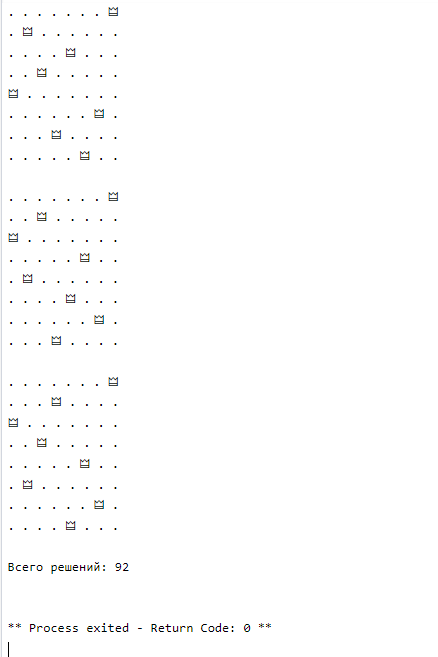
****

****

****

**Код с выводом**

****

****