## UNIVERSITY INSTITUTE OF COMPUTING

**PROJECT REPORT ON**

## TOWER OF HANOI

Program Name: BCA

Subject Name/Code: Data Structures(23CAT-201)

**Submitted by: Submitted to:**

Name: Shivam Koshyari Name: Mrs. Shilpi Mittal

UID: 23BCA10050 Designation: Co-ordinator

Section: BCA – 3 “A”

Group: 1

# ABSTRACT

### Introduction:

The **Tower of Hanoi** is a classic mathematical puzzle that demonstrates recursive problem-solving. Originally developed by French mathematician Édouard Lucas in 1883, the puzzle consists of three rods and a number of disks of different sizes that can slide onto any rod. The disks start stacked in descending size order on one rod, and the objective is to move the entire stack to another rod following a few simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or an empty rod.
3. No disk may be placed on top of a smaller disk.

### Technique:

The application is developed using C++ with a focus on Function Recursion. It employs the following techniques:

1. *Recursion* : *Recursion* in C++ is a process in which a function calls itself within its definition, with a base case to end the recursive calls and prevent infinite loops.
2. Linked List Data Structure: Linked lists are used for dynamic storage of user accounts, allowing eﬃcient addition, searching, and deletion operations. This structure is ideal for managing user data without the need for ﬁxed-size arrays.

### System Configuration:

* OS: Windows 10 or Linux
* Processor: Intel Core i3 (minimum);
* RAM: 4 GB (minimum);
* Development Environment: Any C++ IDE (e.g., Visual Studio, GCC Compiler)

# SUMMARY

The **Tower of Hanoi** project involves solving the classic puzzle where disks of different sizes are moved between three rods, following specific rules:

**Objective**: Move all disks from the starting rod to the target rod.

**Rules**:

* Only one disk can be moved at a time.
* A larger disk cannot be placed on top of a smaller disk.
* Disks must be moved to one of the three rods only.

**Steps**:

* Select the number of disks.
* Follow the algorithm or recursive steps to move disks between rods.
* Track the number of moves required to complete the puzzle (optimal solution for *n* disks is 2  
  2^n - 12n−1 moves).

The project typically includes options to:

* Visualise the solution steps.
* Practice solving the puzzle manually or using a guided approach.

**PROCESS:**

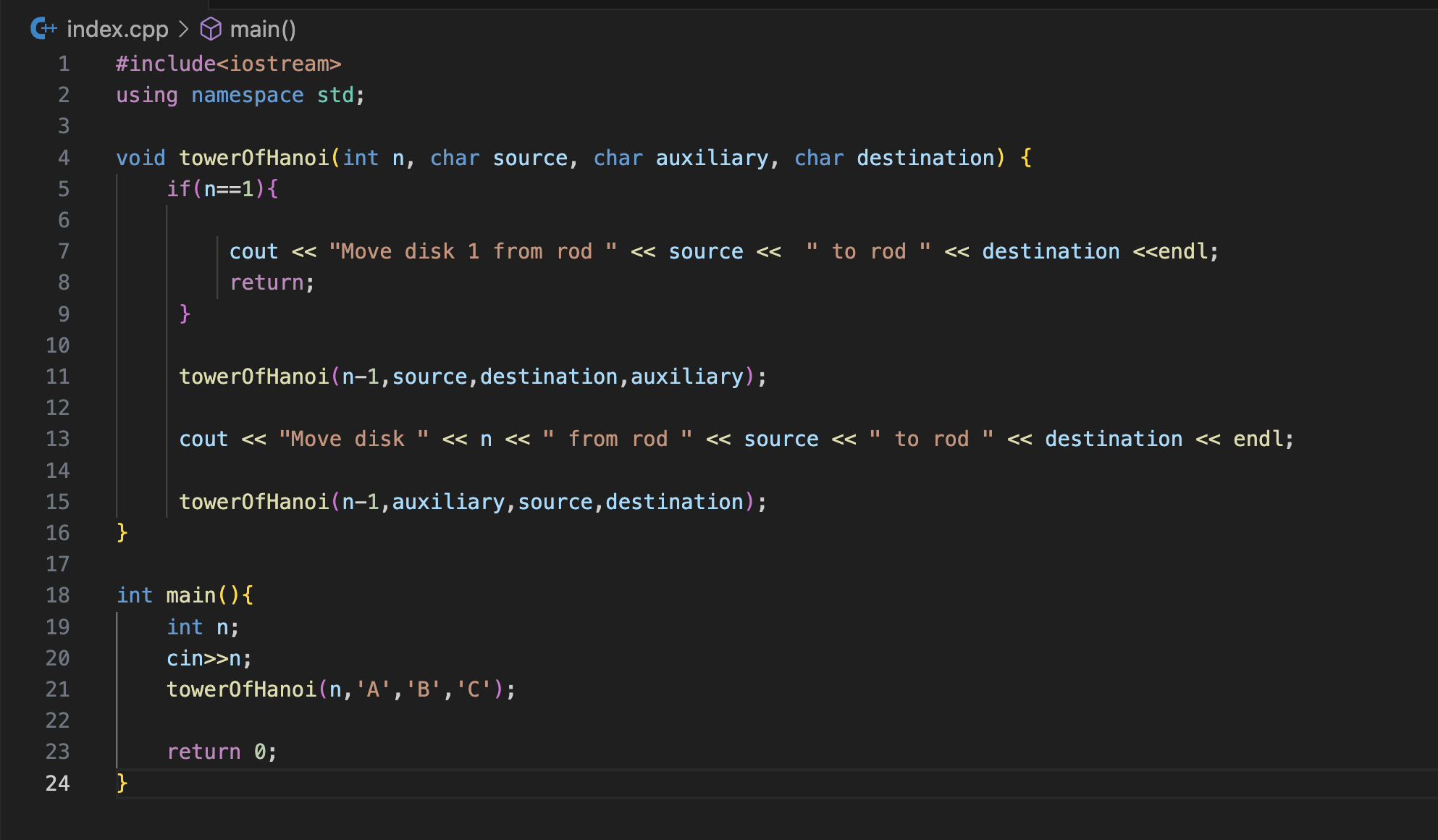
1. **Input Number of Disks** (n)
2. **Check if n is 1**

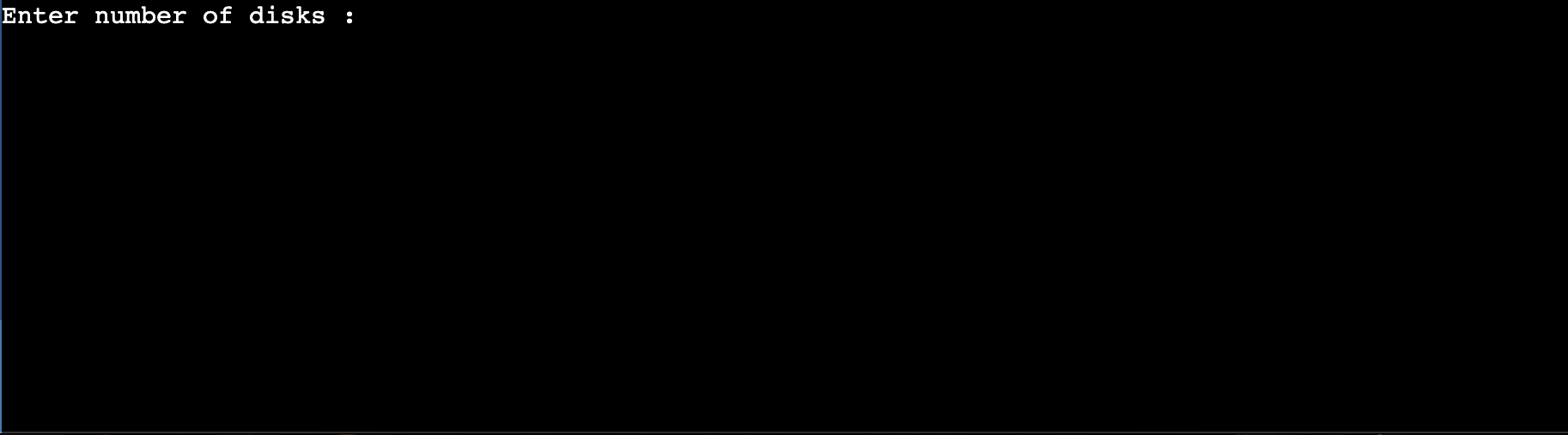
* If **Yes**: Move the disk directly from the starting rod to the target rod.
* If **No**:
  + **Step A**: Move the top n-1 disks from the starting rod to the auxiliary rod.
  + **Step B**: Move the nth (largest) disk from the starting rod to the target rod.
  + **Step C**: Move the n-1 disks from the auxiliary rod to the target rod.

1. **Repeat until all disks are on the target rod**

### Process:mewmew nigga.png

Output:

Main Code :

Input:

Result: