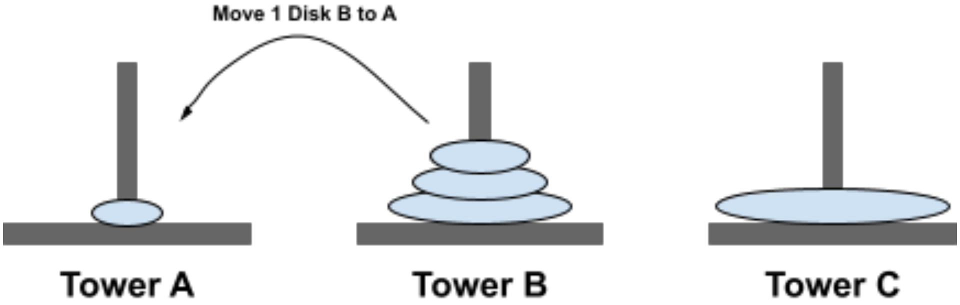
University Ferhat Abess-Sétif 1
Faculty of sciences
Computer sciences Departement
Module: ASD3
Level: 2nd-year Computer Science Engineering Students **Dr. Lakehal Abderrahim**

TP 1

Objective: Implement and understand the principle of recursion through a classic example.

Instructions: Type the source code for each section, write a complete `int main()` to test it, and correct all lexical, syntax, semantic, runtime, and logic errors. Finally, test your program with various input sizes and discuss the program's behavior as the input size (*n*) becomes large.

A. Tower of Hanoi



Problem Statement

You are given **three pegs** (or rods):

- **Source (A)**
- **Auxiliary (B)**
- **Destination (C)**

You also have **n disks** of different sizes that can slide onto any peg. The puzzle starts with the disks neatly stacked in **increasing size order** on the **source peg**, smallest on top.

Goal:

Move all the disks from the **source peg (A)** to the **destination peg (C)** following these **rules**:

1. You can move **only one disk at a time**.
2. Each move consists of taking the **top disk** from one stack and placing it on **top of another stack**.
3. **No disk** may be placed on **top of a smaller disk**.

To do:

Develop a recursive C++ program that simulates the movement of disks in the Tower of Hanoi problem.

Hints:

Consider N disks numbered from 1 to N and three pegs labeled A, B, and C.

Identify the **base case** and the **recursive case** in this program.

To move `n` disks from **A** → **C** using **B** as auxiliary:

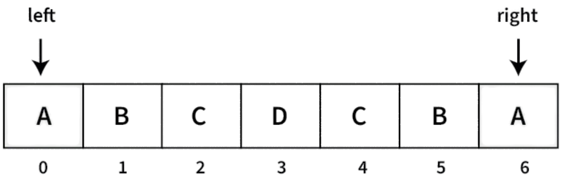
1. Move the top `n-1` disks from **A** → **B** using **C** as auxiliary.
2. Move the **remaining largest disk** from **A** → **C**.
3. Move the `n-1` disks from **B** → **C** using **A** as auxiliary.

To visualize the execution trace, print each movement in the format: `Move disk 1 from A -> C`**.**

B. Palindrome Check

Problem Statement

We need to check if a string is a palindrome (i.e., it reads the same forward and backward) using recursion.



To do:
write a recursive C++ function named (`isPalindrome`) that determines whether a given string is a palindrome or not:

```
bool isPalindrome(string s, int left, int right);
```