**1.Thuật toán đệ quy là** một phương pháp lập trình trong đó một hàm tự gọi chính nó để giải quyết một bài toán. Trong C++, việc sử dụng đệ quy giúp giải quyết các bài toán phức tạp bằng cách chia nhỏ chúng thành các vấn đề nhỏ hơn và tương tự như nhau.

Ví dụ:

// Hàm đệ quy tính giai thừa : n!

**int factorial(int n)**

**{**

**if (n <= 1)**

**{**

**// Điều kiện dừng**

**return 1;**

**}**

**else**

**{**

**return n \* factorial(n - 1);**

**// Gọi đệ quy**

**}**

**}**

// Hàm đệ quy tính số

**Fibonacci int fibonacci(int n)**

**{**

**if (n <= 1)**

**{**

**return n;**

**}**

**else**

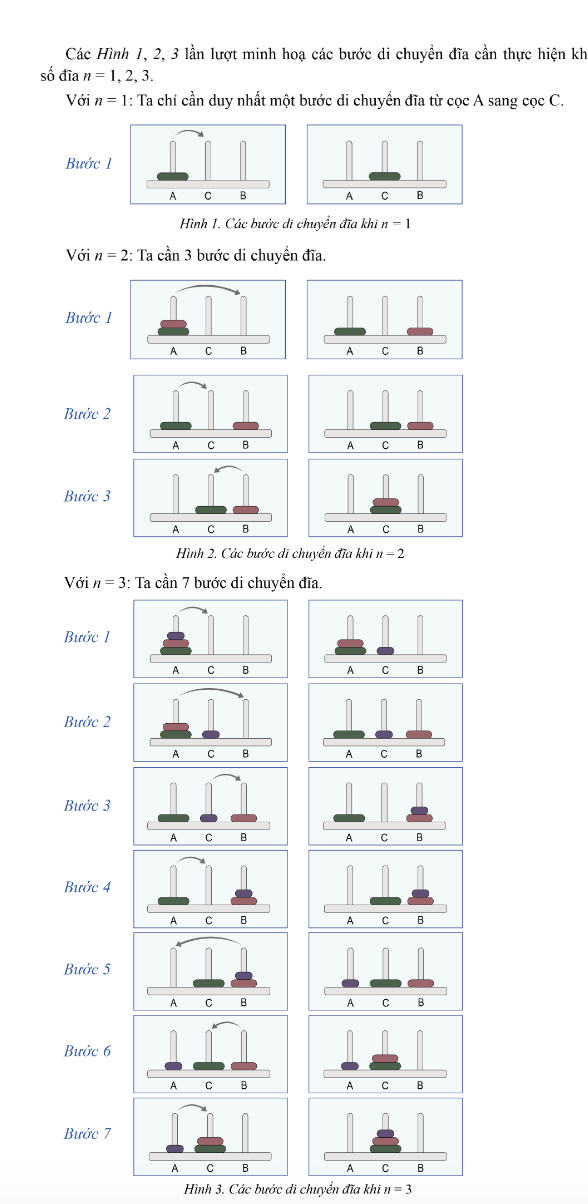
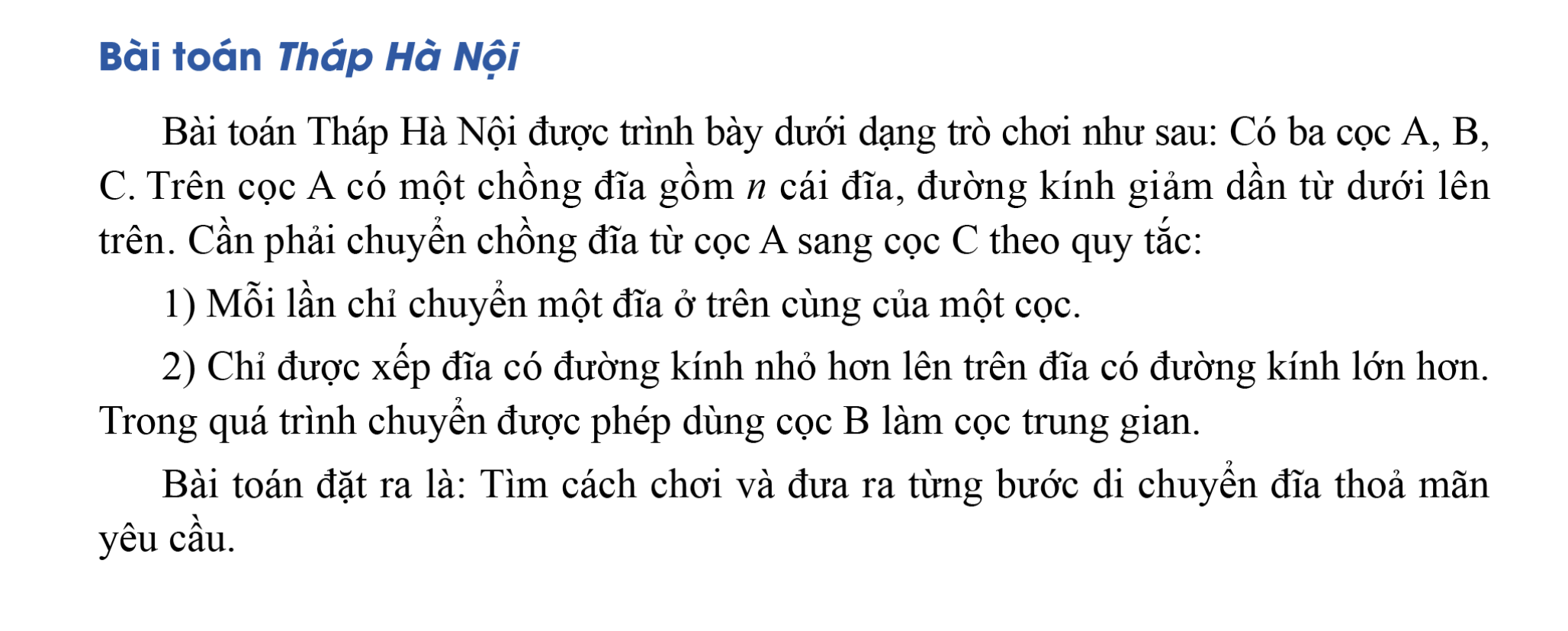
**{**

**return fibonacci(n - 1) + fibonacci(n - 2);**

**}**

**}**

BÀI 1



#include<bits/stdc++.h>

using namespace std;

// Hàm đệ quy giải bài toán Tháp Hà Nội

void thap(int n, char a, char c, char b) {

if (n == 1) {

// Nếu chỉ còn 1 đĩa, di chuyển nó từ cột từ (a) sang cột đến (c)

cout << "Di chuyen dia 1 tu cot " << a << " sang cot " << c << endl;

return;

}

// Di chuyển n-1 đĩa từ cột từ (a) sang cột phụ (b) bằng cột đến (c)

thap(n - 1, a, b, c);

// Di chuyển đĩa lớn nhất từ cột từ (a) sang cột đến (c)

cout << "Di chuyen dia " << n << " tu cot " << a << " sang cot " << c << endl;

// Di chuyển n-1 đĩa từ cột phụ (b) sang cột đến (c) bằng cột từ (a)

thap(n - 1, b, c, a);

}

int main() {

int n; // Số lượng đĩa

cout << "Nhap so dia: ";

cin >> n;

cout << "Cac buoc di chuyen dia tu cot A sang cot C:" << endl;

thap(n, 'A', 'C', 'B'); // 'A' là cột nguồn, 'C' là cột đích, 'B' là cột phụ

return 0;

}

BÀI 2

