

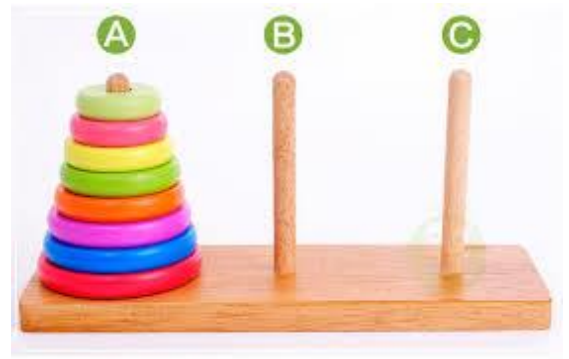
Problem 39: Tower of Hanoi

Difficulty: Hard

Originally Published: Code Quest 2016

Problem Background

The Tower of Hanoi is a mathematical game or puzzle. It was invented in 1883 by French mathematician Edouard Lucas. It consists of three pegs labeled A, B, and C and a number of disks of different sizes which can slide onto any peg.



Problem Description

The puzzle starts with the disks in a neat stack in ascending order of size on peg A, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack from peg A to peg C using peg B to help, obeying the following 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; i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

With two disks, the puzzle can be solved in three moves, with three disks, the puzzle can be solved in seven moves. The minimum number of moves required to solve a Tower of Hanoi puzzle is 2^{N-1} , where N is the number of disks.

This puzzle is well known since it touches two important topics.

- Recursive functions and stacks
- Recurrence relations

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A positive integer N representing the number of disks in the initial stack.

2
2
3

Sample Output

Your program's output should be as follows:

- The first line of each test case's output should be the number of disks in the stack.
- The next 2^{N-1} lines should be the disk movements in the form FromPeg->ToPeg. Since only the top disk can be moved, it is not necessary to print out the disk number.

```
2
A->B
A->C
B->C
3
A->C
A->B
C->B
A->C
B->A
B->C
A->C
```