CS 506, Online Foundations of CS	**Programming Assignment 2 Towers of Hanoi**	Dr. David Nassimi

Study Module 7 (Recursive Algorithms)

This is a simple programming assignment involving **recursion**. You are to implement the Towers-of-Hanoi problem, which is an interesting puzzle that can be solved by using recursion. There are three towers: A, B, and C. Initially, n discs are stacked on tower A, with each disc smaller than the one below it. The object of the game is to move the stack of discs to tower B in the same order. (The third tower is used for intermediate storage.) There are two rules for moving the discs:

- Only one disc can be moved at a time from the top of one tower to the top of another.
- A larger disc must never be placed on top of a smaller one.

First, write a recursive algorithm to solve the problem. To analyze the time complexity, let $f(n)$ be the total number of single moves to solve the problem. Write a recurrence equation for $f(n)$ and solve the recurrence. To get a better appreciation for this time complexity, tabulate $f(n)$ for the values of n from 1 to 20.

Then, implement a program for this problem. (Input the number of discs from the keyboard, and printout each move.) Limit n to at most 6 to avoid excessive output, because the number of moves is exponential! Below is a sample output:

Towers of Hanoi Solver.
Input number of disks (max 6): 2

	Tower A	Tower B	Tower C
Initial State:	2 1		
1 (Move disc 1 from A to C):	2		1
2 (Move disc 2 from A to B):		2	1
3 (Move disc 1 from C to B):		2 1	

Your program must be in C, C++, or JAVA. Submit a hard-copy (paper copy) of your program and the produced output. Run your program for two values of n (e.g., 3 and 5).