```
/*
    Author: Autumn C. Spaulding
    Creation Date: 24 August 2000

    *********************************************************************

                Towers of Hanoi

    *********************************************************************

    Description:  This is a solution to the Towers of Hanoi Puzzle.
    The program asks the user for the number of disks (and does minimal
    error checking), then solves the Towers of Hanoi Problem for that
    number of disks.

    The Towers of Hanoi Problem: You have a set of three pegs (peg 1,
    peg 2, and peg 3), on one of which there are n disks.  Transfer all of
    the pegs from peg 1 to peg 3 while moving only one at a time, and with
    the requirement that a disk must be smaller than any that it is placed
    on.

    *       _|_          |           |            << START
    *      __|__         |           |
    *     ___|___        |           |
    *    ____|____       |           |
    *    _________________________________


    *        |           |           |
    *        |           |           |            << VALID
    *        |           |          _|_
    *    ____|____    ___|___      __|__
    *    _________________________________


    *        |           |           |
    *        |           |           |            << INVALID
    *        |       ____|____      _|_
    *        |        ___|___      __|__
    *    _________________________________

The output should look like:

The Towers of Hanoi Puzzle
How many disks?  n < 12
4
Move from 1 to 2
Move from 1 to 3
Move from 2 to 3
Move from 1 to 2
Move from 3 to 1
Move from 3 to 2
Move from 1 to 2
Move from 1 to 3
Move from 2 to 3
Move from 2 to 1
Move from 3 to 1
Move from 2 to 3
Move from 1 to 2
Move from 1 to 3
Move from 2 to 3
Problem Solved!

*/
```

```
    *****************************  C++ Code  ******************************

#include <iostream.h>
void hanoi(int numDisks, int startPost, int endPost, int freePost);

int main()
{
    cout << "The Towers of Hanoi Puzzle" << endl;
    int numDisks;
    do
    {
        cout << "How many disks?  n <= 12" << endl;
        cin >> numDisks;
    }while (numDisks > 12 || numDisks < 1);

    hanoi(numDisks, 1, 3, 2);
    cout << "Problem solved!" << endl;

    return 1;
}

void hanoi(int numDisks, int startPost, int endPost, int freePost)
{
    if (numDisks == 1)
        cout << "Move from " << startPost << " to " << endPost << endl;
    else
    {
        hanoi(numDisks-1, startPost, freePost, endPost);
        hanoi(1, startPost, endPost, freePost);
        hanoi(numDisks-1, freePost, endPost, startPost);
    }
}

    *****************************  Java Code  ******************************

public class HanoiApp
{
    public static void main(String args[])
    {
        System.out.println( "The Towers of Hanoi Puzzle" );
        Keyboard getInput = new Keyboard();
        int numDisks = getInput.readInt("How many disks? n <= 12  ", 1, 12, 6);

        TowersOfHanoi hanoiGame = new TowersOfHanoi();
        hanoiGame.hanoi(numDisks, 1, 3, 2);

        System.out.println("Problem Solved!");
    }
}

class TowersOfHanoi
{
    void hanoi(int numDisks, int startPost, int endPost, int freePost)
    {
        if (numDisks == 1)
            System.out.println("Move from " + startPost + " to " + endPost);
        else
        {
            hanoi(numDisks-1, startPost, freePost, endPost);
            hanoi(1, startPost, endPost, freePost);
            hanoi(numDisks-1, freePost, endPost, startPost);
        }
    }
}
```