More Recursion

1. What is wrong with these methods?

a)
```
public static double bad (double a, double b) {
   a = a / 2;
   b = b*2;
   return bad(a, b);
}
```

b)
```
public static int badToo (int n) {
   if (n < 1)
      return 0;
   else if (n==1)
      return 5;
   else
      return 2 * badToo(n+1) + 3;
}
```

2. **TowerOfHanoi.java**: In the Towers of Hanoi problem, the number of moves grows rapidly as the number of disks increases. By following the algorithm given in class, write a recursive method `countMoves` that takes in the size of the tower and return the number of moves.

3. **PrintRow.java**:
   a) Give a recursive description of the process of printing a row containing n asterisks.
   b) Complete the definition of a recursive method `printRow` whose header is shown below. The method should print a line containing a row of `n` asterisks. If `n` is less than one, the method should print nothing.
   ```
   public static void printRow (int n)
   ```

4. **PrintTriangle.java**: Suppose that the following pattern is called a *5-triangle*
   ```
   *
   **
   ***
   ****
   *****
   ```
   a) Give a recursive description of the process of printing an *n-triangle*
   b) Write a recursive method `printTriangle` with a single `int` parameter `n`. The method should print a triangle of asterisks like the one shown here but containing `n` rows. If `n` is less than one, the method should print nothing.

5. **LargestOfList.java**:
   a) Write a recursive method with the following header to determine the largest value of a list of values. The parameter `list` is the array

represent the list of integers, `start` represents the index of the start of the list and `end` represents the index of the end of the list.

```
largest(int[] list, int start, int end)
```

b) Write a wrapper method with the following header for the recursive method above.

```
largest(int[] list)
```

6. **BinarySearch.java**: Write recursive method to implement the binary search algorithm. Given an array of integers, and a target integer, the method should return the location (index) of the target in the array.