Exercise #7 (No computers needed)
Due Friday, 11/13/2015, at 11:59pm

New Terms
In your own words define the following: Recursion, Heap vs. Stack, Array

2-D Arrays This Week
We have seen 1-d arrays, but we will learn about multi-dimensional arrays this week. Since Assignment #5 requires the use of a 2-d array, begin discussing what it is and how you use it!

How do you declare a 1-d array of characters?
How do you access an element in a 1-d array?
How do you pass a 1-d array to a function, i.e. what is the name by itself?

How do you declare a 2-d array of characters?
How do you access an element in a 2-d array?
How do you pass a 2-d array to a function, i.e. what is the name by itself?

As a group, determine the declaration and initialization of the 2-d board needed in Assignment #5. Write the definition of the initialize_board() function and provide an example call to the function.

Understanding Recursion, Another example:
Let’s look at a classic recursive problem called the Towers of Hanoi. This is a game where you have 3 posts and n disks of different sizes, which are initially arranged in ascending order on the 1st post. Your goal is to get the disks arranged on the 2nd post in ascending order using these following rules:

- You can only move one disk at a time.
- You cannot put a larger disk on top of a smaller disk.

First, begin by writing the steps on a piece of paper that represents the moves among the posts. For instance, with three disks, the smallest disk from the 1st post will be moved to the second post, i.e. 1 -> 2. Then, the 2nd disk will be moved to the 3rd post, i.e. 1->3, etc.

Write the steps for the base case, n = 1 disks, n = 2 disks, and n = 3 disks. You should notice that you have \(2^n - 1\) moves for each of these cases. Also, note any pattern that you see, i.e. when do you see the base case.

Here is an outline of the recursive towers function:

```c
void towers(int disks, int b[][3], int from_col, int to_col, int spare) {
    if(number of disks is >= 1) {
        Call Towers with (disks-1, b, from_col, spare, to_col)
        Move the disk
        Print the board
        Call Towers with (disks-1, b, spare, to_col, from_col)
    }
}
```
As a group with the TAs, walk through the algorithm provided for the towers() function with a 3 x 3 board, b. Remember, it just takes example input arguments to follow what the algorithm would do, e.g. towers(2, b, 1, 2, 3);, towers(3, b, 1, 2, 3);, etc.

Provide the example walk through for the following calls:

- towers(1, b, 1, 2, 3);
- towers(2, b, 1, 2, 3);
- towers(3, b, 1, 2, 3);

For take-home exercises completed in peer-led groups, each student must participate and write answers to each of the questions on his/her own paper to show for credit. Your 1-2-3 grade will be based on the completion/understanding shown on your own piece of paper shown to your TA for a grade before leaving the group session!!!

For take-home exercises completed on your own, turn in your work electronically using the TEACH website!!!