CS 161 Recitation

Worksheet: Week 7

1. **Understanding Recursion**/Towers of Hanoi

Let's look at a classic recursive problem called the **Towers of Hanoi**. This is a game where you have 3 posts/columns and n disks of different sizes, which are initially arranged in ascending order on the 1st post/column. Your goal is to get the disks arranged on the 2nd post/column 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:

```
void towers(number_of_disks, from_post, to_post, spare_post) {
    If(number of disks is >= 1)
      Call Towers with (number_of_disks-1, from_ post, spare_ post, to_ post)
      Move the disk
      Call Towers with (number_of_disks-1, spare_ post, to_ post, from_ post)
}
```

As a group with the TAs, walk through the algorithm provided for the towers() function with a board that has 1 disk and 3 posts, then 2 disks and 3 posts, and 3 disks with 3 posts, e.g. . towers(1, 1, 2, 3);, towers(2, 1, 2, 3);, towers(3, 1, 2, 3);, etc.

Provide the example walk through for the following calls:

```
towers(1, 1, 2, 3);
towers(2, 1, 2, 3);
towers(3, 1, 2, 3);
```

2. What is the heap vs. stack? What is a memory leak? How can we avoid them?

3. What is an array? How do you access elements in an array? What is the difference in a static vs. dynamic array? What is a C-style string?