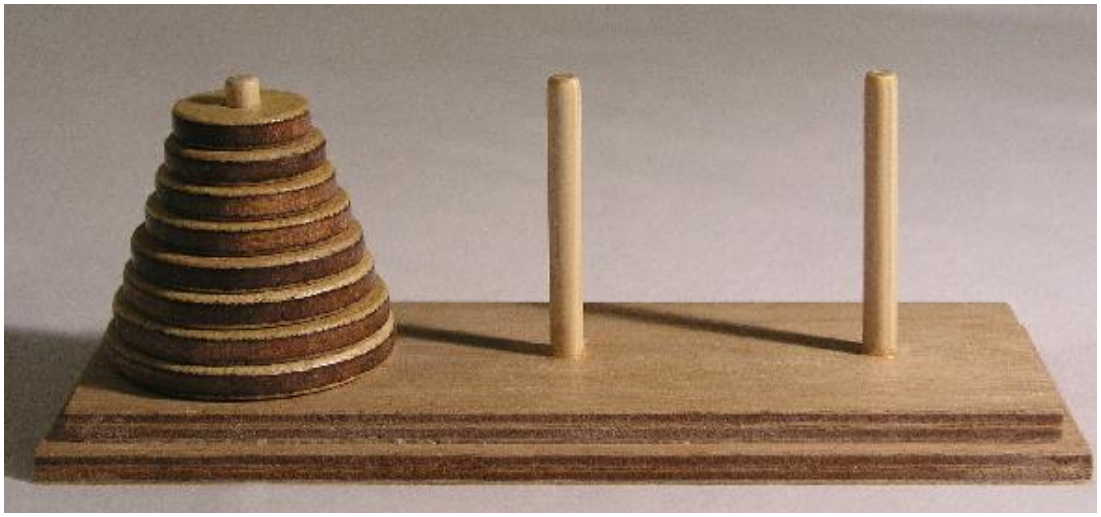
Homework Description: Tower-Of-Hanoi

The Tower of Hanoi is a mathematical puzzle. It consists of three poles and a number of disks of different sizes which can slide onto any poles. The puzzle starts with the disk in a neat stack in ascending order of size in one pole, the smallest at the top thus making a conical shape.

The puzzle has the following two rules:

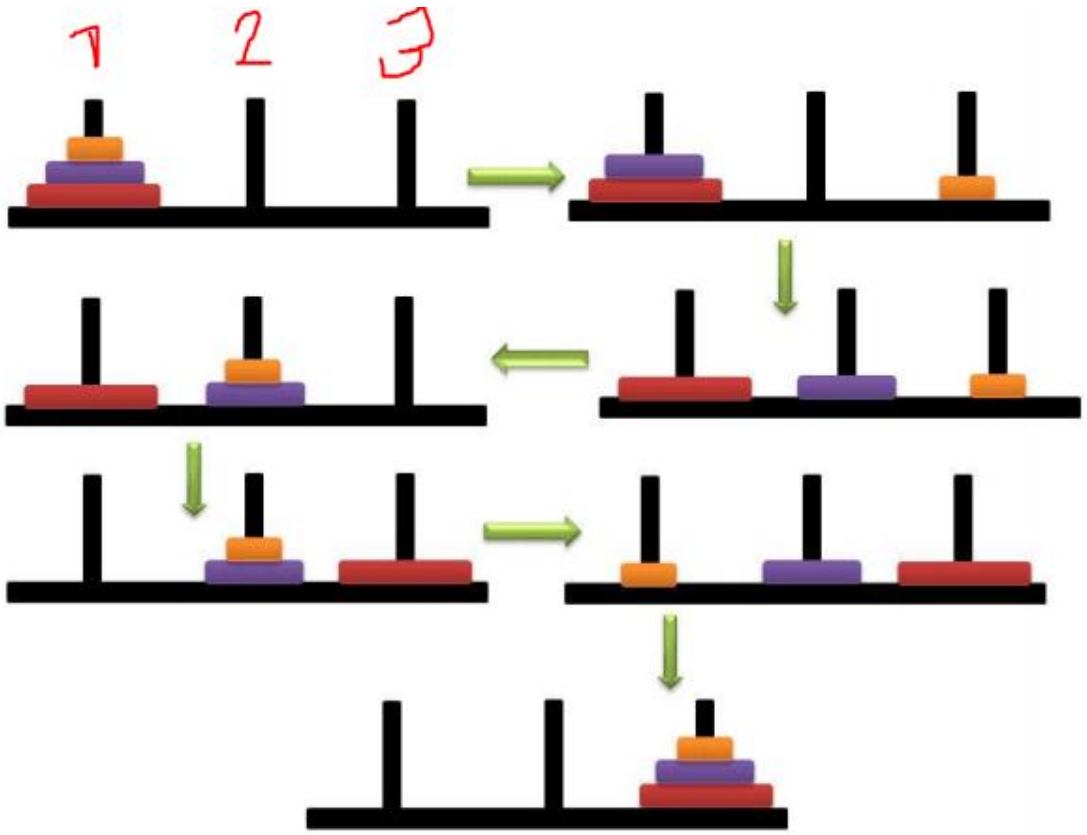
1. You can't place a larger disk onto a smaller disk
2. Only one disk can be moved at a time



```
#define STACK_BLOCK_SIZE 10

typedef struct { data_type * array; int currentsize; int maxsize} stack;

int push(stack * s, data_type d); /* the stack array will grow STACK_BLOCK_SIZE entries at a time */

data_type pop(stack * s); /* the stack array will shrink STACK_BLOCK_SIZE entries at a time */

stack * init_return(); /* initializes an empty stack */
int init(stack * s); /* returns 1 if initialization is successful */
```

Using this, implement a solution to tower-of-hanoi problem **without recursion and using stack.** Your solver should be able to take any size tower.

Example for size 3:



Example output for size 3:

```
Enter Tower size:3
Move the disk 1 from '1' to '3'
Move the disk 2 from '1' to '2'
Move the disk 1 from '3' to '2'
Move the disk 3 from '1' to '3'
Move the disk 1 from '2' to '1'
Move the disk 2 from '2' to '3'
Move the disk 1 from '1' to '3'
```