3) (10 pts) DSN (Stacks)

Suppose we have implemented a stack using a linked list. The structure of each node of the linked list is shown below. The stack structure contains a pointer to the head of a linked list and an integer, size, to indicate how many items are on the stack.

```
typedef struct node {
    int num;
    struct node* next;
} node;

typedef struct stack {
    struct node *top;
    int size;
} stack;
```

The generalized Towers of Hanoi game can be represented by **`numTowers`** stacks of integers, where the values in each stack represent the radii of the disks from the game for the corresponding tower. Recall that a valid move involves taking a disk at the top of one stack and placing it on the top of another stack, so long as that other stack is either empty or the disk currently at the top of the other stack is bigger than the disk about to be placed on it. Complete the function below so that it takes in an array of stacks representing the contents of the towers in Towers of Hanoi and prints out all of the valid moves that could be made from that state, but doesn't move anything. You may assume that the array of stacks passed into the function represent a valid state in a Towers of Hanoi game, where the value stored in the stack is the corresponding disk radius and the disk radii range from 1 to n, for some positive integer n. Assume that you have access to the following functions that involve a stack and that they work as described:

```
// Returns the value stored at the top of the stack pointed to by s. If stack pointed to by s is empty, a
// random value is returned.
int peek(stack *s);

// Returns 1 if the stack pointed to by s is empty, and 0 otherwise.
int isEmpty(stack *s);

void printValidMoves(stack towers[], int numTowers) {

    for (int i=0; i<numTowers; i++) {
        for (int j=0; j<numTowers; j++) {

            if ( isEmpty(&towers[i]) ) continue;

            if ( isEmpty(&towers[j]) || peek(&towers[i]) < peek(&towers[j]) )

                printf("Valid Move from tower %d to tower %d.\n", i, j);
        }
    }
}
```

Grading: 3 pts first slot, 3 pts second slot, 4 pts last slot