```c
//
//  Stack.h
//  Tower of Hanoi
//
//  Created by Enrique Perez on 3/28/24.
//

#include "Stack.h"
#include "Node.h"
#include <stdio.h>
void discSetup(struct Stack * stack, int i);

int main(void) {
    int numOfDisks = -1;
    while (numOfDisks < 3) {
        printf("Enter number of disks: ");
        scanf("%d", &numOfDisks);
    }

    struct Stack *peg1 = create_stack(numOfDisks);
    Stack *peg2 = create_stack(numOfDisks);            // no need for "struct" when using typedef
    Stack *peg3 = create_stack(numOfDisks);
    List* stack_list = create_list();        // future development using nodes instead of an array.20

    // 03/30/24: creating an array of stacks pointers to cycle through pegs.
    struct Stack *stacks[3];  // current stack at stacks[cur]
    stacks[0] = peg1;
    stacks[1] = peg2;
    stacks[2] = peg3;

    // 03/28/24
    discSetup(peg1, numOfDisks);

    // 03/30/24 precondition: there are at least 3 disks on peg1
    int cur = 0;
    int from = 0;
    unsigned long moves = 0;
    do{
        while(is_empty(stacks[cur])) cur = (cur+1)%3;
        from = cur;
        int disc = pop(stacks[cur]);        // B
        cur = (cur+1)%3;
        while(!is_empty(stacks[cur]) && (disc > peek(stacks[cur]))) cur = (cur+1)%3;
        if (from != cur){ //if equal, that means the disc didn't move. commented out for effeciency.
            if (is_empty(stacks[cur])) printf("disc %d moved from peg%d to peg%d (empty peg)\n", disc, from+1, cur+1);
            else printf("disc %d moved from peg%d to peg%d (top disc %d)\n", disc, from+1, cur+1, peek(stacks[cur]));
            moves++;
            if (moves%1000000 == 0) printf("%lu moves\n", moves);
        }
        push(stacks[cur], disc);
        cur = (cur+1)%3;
    }while(!is_full(peg2) && !is_full(peg3));
    //printf("breakpoint, check if discs moved in order.");

    printf("peg2: ");
    while(!is_empty(peg2)) printf("%d, ", pop(peg2));
    printf("\npeg3: ");
    while(!is_empty(peg3)) printf("%d, ", pop(peg3));
    printf("\n%lu moves completed.\n", moves);

  return 0;
}

void discSetup(struct Stack* stack, int numOfDisks){
    for (int i=numOfDisks; i > 0; i--){
        push(stack, i);
    }
}
```