

Here we are implementing Towers of Hanoi program in a non recurring program using stacks

```
#include <stdio.h>
```

```
struct State {  
    int n;  
    char from_peg;  
    char to_peg;  
    char aux_peg;  
};
```

```
struct State stack[100];  
int top = -1;
```

```
void push(struct State s) {  
    stack[++top] = s;  
}
```

```
struct State pop() {  
    return stack[top--];  
}
```

```
int is_empty() {  
    return top == -1;  
}
```

```
void towersOfHanoiIterative(int n, char from_peg, char to_peg, char aux_peg) {
```

```
    push((struct State){n, from_peg, to_peg, aux_peg});
```

```
    while (!is_empty()) {  
        struct State current = pop();
```

```
        if (current.n == 1) {  
            printf("Move disk 1 from peg %c to peg %c\n", current.from_peg, current.to_peg);  
        } else {
```

```
            push((struct State){current.n - 1, current.aux_peg, current.to_peg, current.from_peg});
```

```
        printf("Move disk %d from peg %c to peg %c\n", current.n, current.from_peg,
current.to_peg);

        push((struct State){current.n - 1, current.from_peg, current.aux_peg, current.to_peg});
    }
}

int main() {
    int num_disks;
    printf("Enter the number of disks: ");
    scanf("%d", &num_disks);

    towersOfHanoiIterative(num_disks, 'A', 'C', 'B');
    return 0;
}
```

main.c

Output



```
Enter the number of disks: 3
Move disk 3 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg A to peg C
Move disk 1 from peg C to peg B
Move disk 2 from peg B to peg C
Move disk 1 from peg B to peg A
Move disk 1 from peg A to peg C
```

=== Code Execution Successful ===