TOWER OF HANOI

The Tower of Hanoi is a classic problem that involves moving a tower of disks from one peg to another, subject to the following constraints:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty peg.
3. No disk may be placed on top of a smaller disk.

Algorithms

TOH(n, src, dest, aux)

1. Start
2. if (n=1)
   move disk 1 from src to dest
   else
   i. TOH(n-1, src, aux, dest)
   ii. move n disk from src to dest
   iii. TOH(n-1, aux, dest, src)
3. Stop

IMPLEMENTATION

```c
#include <stdio.h>

void towerOfHanoi(int n, char src, char dest, char aux) {
    if (n == 1) {
        printf("Move disk 1 from %c to %c\n", src, dest);
        return;
    }

    towerOfHanoi(n - 1, src, aux, dest);
    printf("Move disk %d from %c to %c\n", n, src, dest);
    towerOfHanoi(n - 1, aux, dest, src);
}

int main() {
    int n;

    printf("Enter the number of disks: ");
    scanf("%d", &n);

    printf("\nTower of Hanoi Steps:\n");
    towerOfHanoi(n, 'A', 'B', 'C');
```

```
    return 0;
}
```

OUTPUT

```
Enter the number of disks: 4

Tower of Hanoi Steps:
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
```