# Tower of Hanoi using Recursion in C

## A Project Report

**WIN SEM 2020-21**

## CSE2014

*Submitted by*

**Gelle Hruthesh Reddy***(20BCB7031)*

**Gogireddy Sai Nithin Reddy***(20BCB7027)*


**VIT-AP UNIVERSITY**

**AMARAVATI**

**ANDHRA PRADESH, INDIA**

**2021**

**Objective:**

To study the problem of Tower of Hanoi using Recursion in C

**Theory:**

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

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 i.e., a disk can only be moved if it is the uppermost disk on a stack.

3.No disk may be placed on top of a smaller disk.

**Algorithm:**

As per theory, our objective is to move all the disks placed on pole A to C. We need to move n-1 disks from A to B. After moving we can move largest disk to C. Then we can move disks from B to C. If there is only one disk, then we can directly move to C. If we have more disks, we need to use recursion by calling a function with n-1 disks, from rod, auxiliary rod, to rod. Disk is moved from rod to to rod. Again, calling function with n-1, Auxiliary rod, to rod, from rod. And process continues until it satisfies.

**Code:**

```
#include <stdio.h>


void tower(int n,char x,char y,char z);

int main()

{

   int n;

   printf("Enter the number of disks: ");

   scanf("%d",&n);

   tower(n,'A','B','C');

   return 0;

}

void tower(int n,char x,char y,char z)

{

   if (n==1)

   {
```

```
        printf("\nDisk 1 is moved from rod %c to rod %c",x,y);
        return;
    }
    tower(n-1,x,z,y);
    printf("\nDisk %d is moved from rod %c to rod %c",n,x,y);
    tower(n-1,z,y,x);
}
```

**Output:**

```
Enter the number of disks: 5

Disk 1 is moved from rod A to rod B
Disk 2 is moved from rod A to rod C
Disk 1 is moved from rod B to rod C
Disk 3 is moved from rod A to rod B
Disk 1 is moved from rod C to rod A
Disk 2 is moved from rod C to rod B
Disk 1 is moved from rod A to rod B
Disk 4 is moved from rod A to rod C
Disk 1 is moved from rod B to rod C
Disk 2 is moved from rod B to rod A
Disk 1 is moved from rod C to rod A
Disk 3 is moved from rod B to rod C
Disk 1 is moved from rod A to rod B
Disk 2 is moved from rod A to rod C
Disk 1 is moved from rod B to rod C
Disk 5 is moved from rod A to rod B
Disk 1 is moved from rod C to rod A
Disk 2 is moved from rod C to rod B
Disk 1 is moved from rod A to rod B
Disk 3 is moved from rod C to rod A
Disk 1 is moved from rod B to rod C
Disk 2 is moved from rod B to rod A
Disk 1 is moved from rod C to rod A
Disk 4 is moved from rod C to rod B
Disk 1 is moved from rod A to rod B
Disk 2 is moved from rod A to rod C
Disk 1 is moved from rod B to rod C
Disk 3 is moved from rod A to rod B
Disk 1 is moved from rod C to rod A
Disk 2 is moved from rod C to rod B
Disk 1 is moved from rod A to rod B

...Program finished with exit code 0
Press ENTER to exit console.
```

**Result:**

After executing the code, we have chosen 5 disks and the puzzle can be played with any $n^{th}$ number of disks. The minimal number of moves required to solve a tower of Hanoi puzzle is $2^n - 1$ and it showed the correct result with obeying the rules of Tower of Hanoi.