

**Title: Implement of Toy Problem**  
**(Tower of Hanoi)**

**Ex. No.:01**

**Reg. No.: RA2011003011334**

**Date: 27/01/2023**

**Name: Rishabh Singh Sahil**

**Aim:**

Implementation of Tower of Hanoi problem using python.

**Procedure/Algorithm:**

```
def TowerOfHanoi(n, from_rod, to_rod, aux_rod):  
    if n == 0:  
        return  
    TowerOfHanoi(n-1, from_rod, aux_rod, to_rod)  
    print("Move disk", n, "from rod", from_rod, "to rod", to_rod)  
    TowerOfHanoi(n-1, aux_rod, to_rod, from_rod)  
  
N = 3  
TowerOfHanoi(N, 'A', 'C', 'B')
```

**Program:**

## Manual Output: Manual calculation for the example you have taken:

The Tower of Hanoi is a mathematical puzzle consisting of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

Only one disk can be moved at a time.

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 rod.

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

A manual calculation for the Tower of Hanoi problem with  $n$  disks can be done by following the steps below:

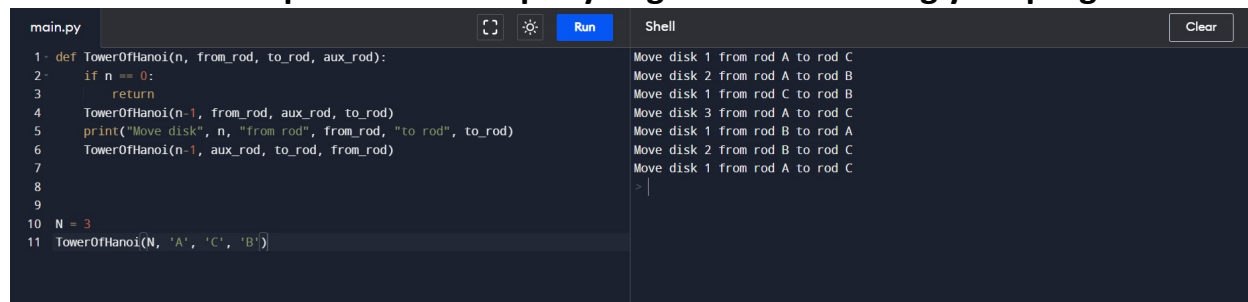
If  $n = 1$ , then only one move is required to move the disk from the source rod to the target rod.

If  $n > 1$ , then follow these steps:

- Move  $n - 1$  disks from the source rod to the auxiliary rod using the target rod as the auxiliary.
- Move the  $n$ th disk from the source rod to the target rod.
- Move the  $n - 1$  disks from the auxiliary rod to the target rod using the source rod as the auxiliary.

The total number of moves required to solve the puzzle with  $n$  disks is  $2^n - 1$ .

## Screenshot of output: Actual Output you get after executing your program:



The screenshot shows a Python IDE with a file named 'main.py'. The code defines a recursive function 'TowerOfHanoi(n, from\_rod, to\_rod, aux\_rod)' that prints the sequence of moves for 3 disks. The output in the shell window shows the following sequence of moves:

```
Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
```

## Result:

Implemented Tower of Hanoi problem successfully.

date  
10/1/2023

## Experiment - I

Aim:

Implementation of Toy problem,  
Tower of Hanoi.

Algorithm:

Step I: Move  $n-1$  disks from Source to aux

Step II: Move  $n^{\text{th}}$  disk from Source to dest

Step III: Move  $n-1$  disks from aux to dest

start

Procedure Hanoi (disk, Source, dest, aux)

IF Disk  $= 1$  Then

move disk Source to dest

ELSE

Hanoi (disk-1, Source, aux, dest)

move disk from Source to dest

Hanoi (disk-1, aux, dest, Source)

END IF

END Procedure

stop.

input

$n = 2$

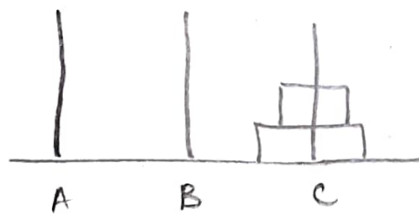
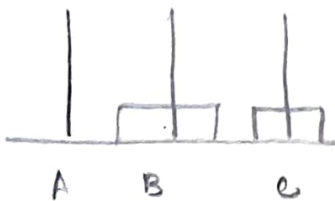
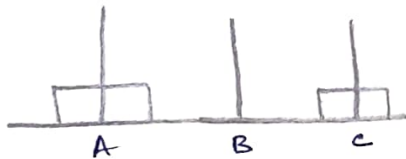
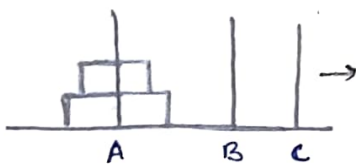
//  $N = \text{No. of Disk}$

output

Disk 2 moved from A to C

Disk 1 moved from B to C

Diagram:



## Manual Calculations:

$$\text{formula} = 2^n - 1$$

where  $n = \text{No. of Disk.}$

if  $n = 2$

$$\Rightarrow 2^2 - 1$$

$$\Rightarrow 3$$

## Result:

Hence, successfully Implemented Toy problem for  
Tower of Hanoi.



```
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void towerOfHanoi(int n, char from_rod, char to_rod,
7                 char aux_rod)
8 {
9     if (n == 0) {
10         return;
11     }
12     towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
13     cout << "Move disk " << n << " from rod " << from_rod
14         << " to rod " << to_rod << endl;
15     towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
16 }
17
18 // Driver code
19 int main()
20 {
21     int N = 3;
22
23     // A, B and C are names of rods
24     towerOfHanoi(N, 'A', 'C', 'B');
25     return 0;
26 }
27
```

```
/tmp/bf3F48N4Mr.o
Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
```