**Program 7**

**Write a Program to Implement Tower of Hanoi using Python.**



**Description:**

The Tower of Hanoi is a classic mathematical puzzle that involves moving a set of disks from one rod to another, adhering to specific rules. In this Python program, we’ll explore how to solve the Tower of Hanoi using recursion, a fundamental programming technique that allows us to break down this complex problem into simpler, manageable sub-problems. The program will demonstrate the sequence of moves required to transfer the disks between the rods efficiently, providing a clear example of recursive problem-solving in action. This educational example not only enhances understanding of recursion but also offers insight into algorithmic thinking for solving puzzles.

## What are the rules of the Tower of Hanoi?

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.

**Note**: Transferring the top n-1 disks from the source rod to the Auxiliary rod can again be thought of as a fresh problem and can be solved in the same manner.

## Tower of Hanoi Using Recursion.

The function [recursively](https://www.geeksforgeeks.org/introduction-to-recursion-2/)breaks down the problem of moving n disks into smaller problems of moving n-1 disks. It alternates the roles of the rods (source, destination, auxiliary) in each recursive call to facilitate the step-by-step transfer of disks according to the rules of the Tower of Hanoi puzzle. The rules are that you can only move one disk at a time and a larger disk may not be placed on top of a smaller disk.

**SOURCE CODE :**

# Recursive Python function to solve the tower of hanoi

def TowerOfHanoi(n , source, destination, auxiliary):

if n==1:

print ("Move disk 1 from source",source,"to destination",destination)

return

TowerOfHanoi(n-1, source, auxiliary, destination)

print ("Move disk",n,"from source",source,"to destination",destination)

TowerOfHanoi(n-1, auxiliary, destination, source)

# Driver code

n = 4

TowerOfHanoi(n,'A','B','C')

# A, C, B are the name of rods

**OUTPUT :**

Move disk 1 from source A to destination C

Move disk 2 from source A to destination B

Move disk 1 from source C to destination B

Move disk 3 from source A to destination C

Move disk 1 from source B to destination A

Move disk 2 from source B to destination C

Move disk 1 from source A to destination C

Move disk 4 from source A to destination B

Move disk 1 from source C to destination B

Move disk 2 from source C to destination A

Move disk 1 from source B to destination A

Move disk 3 from source C to destination B

Move disk 1 from source A to destination C

Move disk 2 from source A to destination B

Move disk 1 from source C to destination B