8. Problem Statement: Tower of Hanoi

**Problem Analysis:**

Tower of Hanoi, is a mathematical puzzle which consists of three towers (pegs) and more than one rings. These rings are of different sizes and stacked upon in an ascending order, i.e. the smaller one sits over the larger one. There are other variations of the puzzle where the number of disks increase, but the tower count remains the same.

The mission is to move all the disks to some another tower without violating the sequence of arrangement. A few rules to be followed for Tower of Hanoi are −

• Only one disk can be moved among the towers at any given time.

• Only the "top" disk can be removed.

• No large disk can sit over a small disk.

**Algorithm:**

TowersOfHanoi(n, x, y, z)

*// Move the top n disks from tower x to tower y.*

{

if(n ≥ 1) then

{

TowersOfHanoi (n-1, x, z, y);

write (“move top disk from tower”, x, “to top of tower”, y);

TowersOfHanoi (n-1, z, y, x);

}

}

**Source Code:**

#include<iostream>

using namespace std;

void TOH(int n, char frompeg, char topeg, char auxpeg)

{

if(n>=1)

{

TOH(n-1, frompeg, auxpeg, topeg);

cout<<"Move "<<n<< " disk from "<<frompeg<< " to " <<topeg<<endl;

TOH(n-1, auxpeg, topeg, frompeg);

}

}

int main()

{

int n;

cout<<"Enter the number of disks : ";

cin>>n;

cout<<"The sequence of moves involved in the Tower of Hanoi are :\n";

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

return 0;

}

**Sample Input:**

Enter the number of disks : 3

**Sample Output:**

The sequence of moves involved in the Tower of Hanoi are :

Move 1 disk from A to C

Move 2 disk from A to B

Move 1 disk from C to B

Move 3 disk from A to C

Move 1 disk from B to A

Move 2 disk from B to C

Move 1 disk from A to C