**Practical No. 8**

**Aim:** Write a recursive function for Tower of Hanoi problem.

**Theory:**

The **Tower of Hanoi** is a mathematical puzzle consisting of three rods (pegs) and a number of disks of different sizes.

1. All disks are initially stacked on one rod in decreasing size order (largest at the bottom, smallest at the top).
2. The objective is to move all disks to another rod, following these rules:
   1. Only one disk can be moved at a time.
   2. A disk can only be moved if it is the topmost disk on a rod.
   3. No disk may be placed on top of a smaller disk.

Recursion is used because the problem can be broken into smaller sub-problems:

1. Move n-1 disks to an auxiliary rod.
2. Move the largest disk to the destination rod.
3. Move n-1 disks from the auxiliary rod to the destination rod.

**Algorithm:**

1. Start.
2. Define a recursive function towerOfHanoi(n, source, auxiliary, destination).
3. If n == 1, move the single disk directly from source to destination.
4. Otherwise:
   * Recursively move n-1 disks from source to auxiliary using destination.
   * Move the remaining disk from source to destination.
   * Recursively move n-1 disks from auxiliary to destination using source.
5. End.

**Program:**

#include <stdio.h> // Include standard input-output header

// Recursive function to solve Tower of Hanoi

void towerOfHanoi(int n, char source, char auxiliary, char destination) {

if (n == 1) { // Base case: only one disk

printf("Move disk 1 from %c to %c\n", source, destination); // Direct move

return; // End recursion

}

// Step 1: Move n-1 disks from source to auxiliary

towerOfHanoi(n - 1, source, destination, auxiliary);

// Step 2: Move the nth (largest) disk from source to destination

printf("Move disk %d from %c to %c\n", n, source, destination);

// Step 3: Move n-1 disks from auxiliary to destination

towerOfHanoi(n - 1, auxiliary, source, destination);

}

int main() {

int n; // Number of disks

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

scanf("%d", &n); // Take user input for number of disks

printf("The sequence of moves:\n");

towerOfHanoi(n, 'A', 'B', 'C'); // Call recursive function with rods A, B, C

return 0; // End of program

}

**Output (Example run with 3 disks):**

Enter the number of disks: 3

The sequence of moves:

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

**Conclusion:**

* The Tower of Hanoi problem was solved successfully using **recursion**.
* The program demonstrated how a complex problem can be broken down into smaller, simpler sub-problems.
* Students learned the importance of base cases in recursion and how recursive calls work step by step to produce the solution.