```
1. #include <stdio.h>
2. void recursive_towerofHanoi(int n, int from, int to, int aux) {
3.   if(n==0)
4.   return;
5.   else
6.   recursive_towerofHanoi(n-1, from, aux, to);
7.   printf("\n Disk %d is moved from tower %d to %d\n", n, from, to);
8.   recursive_towerofHanoi(n-1, aux, to, from);
9. }
}
int main()
{
  int n;
  printf("\n Enter no. of disks");
  scanf("%d", &n);
  recursive_towerofHanoi(n, 1, 3, 2);
  return 0;
}
```

Claim: The recursive_towerofHanoi(n, from, to, aux) correctly computes all the valid moves for moving n disks from the 'from' tower to the 'to' tower via the 'aux' tower s.t. 1) Only one disk can be moved at a time 2) No disk can be placed on top of a disk with a smaller diameter. The disks are in order of decreasing diameter as one scans up the tower.

Proof: The proof is by mathematical induction on n, the no. of disks

Base Case: n=0. If we have 0 disks, there is nothing to move, so we correctly return void in line 4.

I.H.: For an arbitrary n=c, the recursive_towerofHanoi() function correctly computes the valid moves for moving c disks from the 'from' tower (initially 1) to the 'to' tower (initially 3) via the 'aux' tower (initially 2) maintaining all the given condns.