

Write a Prolog Program to implement Towers of Hanoi.

AIM

To write a Prolog program to implement the Towers of Hanoi problem using recursion.

ALGORITHM

1. Start the program.
2. If there is only **1 disk**, move it directly from the source peg to the target peg.
3. If there are **$N > 1$ disks**:
 - Move $N-1$ disks from the source peg to the auxiliary peg.
 - Move the remaining largest disk from the source peg to the target peg.
 - Move the $N-1$ disks from the auxiliary peg to the target peg.
4. Repeat recursively until all disks are moved to the target peg.
5. Stop.

```
% hanoi(N, Source, Target, Auxiliary).  
% N = number of disks  
% Source = starting peg  
% Target = destination peg  
% Auxiliary = helper peg  
hanoi(1, Source, Target, _) :-  
    write('Move disk 1 from '), write(Source),  
    write(' to '), write(Target), nl.  
hanoi(N, Source, Target, Auxiliary) :-  
    N > 1,  
    N1 is N - 1,  
    hanoi(N1, Source, Auxiliary, Target),  
    write('Move disk '), write(N),  
    write(' from '), write(Source),  
    write(' to '), write(Target), nl,  
    hanoi(N1, Auxiliary, Target, Source).
```

```
?-  
% c:/Users/gayathri/Downloads/hanoi.pl compiled 0.00 sec, 2 clauses  
?- hanoi(3, left, right, middle).  
Move disk 1 from left to right  
Move disk 2 from left to middle  
Move disk 1 from right to middle  
Move disk 3 from left to right  
Move disk 1 from middle to left  
Move disk 2 from middle to right  
Move disk 1 from left to right  
true ■
```

RESULT:

The program successfully solves the Towers of Hanoi problem by displaying the sequence of moves required to transfer N disks from the source peg to the target peg using an auxiliary peg.