AI PRACTICAL NO. 8

Girish Nhavkar 9560 TE COMPS A

1. Prolog for tower of Hanoi

```
% hanoi(+N, +A, +B, +C, -Moves)
% N: number of disks
% A: the name of the source peg
% B: the name of the auxiliary peg
% C: the name of the destination peg
% Moves: list of moves to solve the Tower of Hanoi problem
hanoi(0, _, _, _, []).
hanoi(N, A, B, C, Moves) :-
    N > 0,
    N1 is N - 1,
    hanoi(N1, A, C, B, Moves1),
    append(Moves1, [(A,C)], Moves2),
    hanoi(N1, B, A, C, Moves3),
    append(Moves2, Moves3, Moves).

% Example usage:
% ?- hanoi(3, left, middle, right, Moves).
% Moves = [(left, right), (left, middle), (right, middle), (left, right), (middle, left), (middle, right),
(left, right)].

% ?- hanoi_moves(4, Count).
% Count = 15
```

% ?- between(1, 5, N), hanoi(N, left, middle, right, Moves), writeln(Moves).

% [(left,right)]

%Moves = [(left,right)],

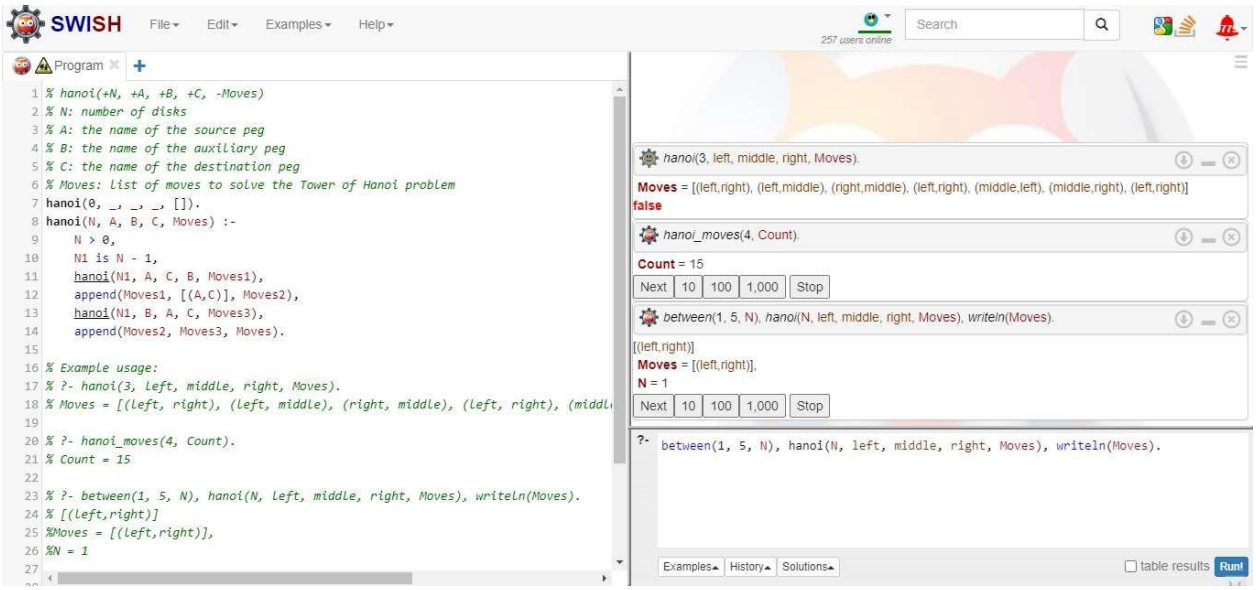
%N = 1

% If you only want to count the number of moves without storing them:

% hanoi_moves(+N, -Count)

```
hanoi_moves(N, Count) :-
    hanoi(N, _, _, _, Moves),
    length(Moves, Count).
```

OUTPUT:



2. Prolog for N- Queen

% render solutions nicely.

:- use_rendering(chess).

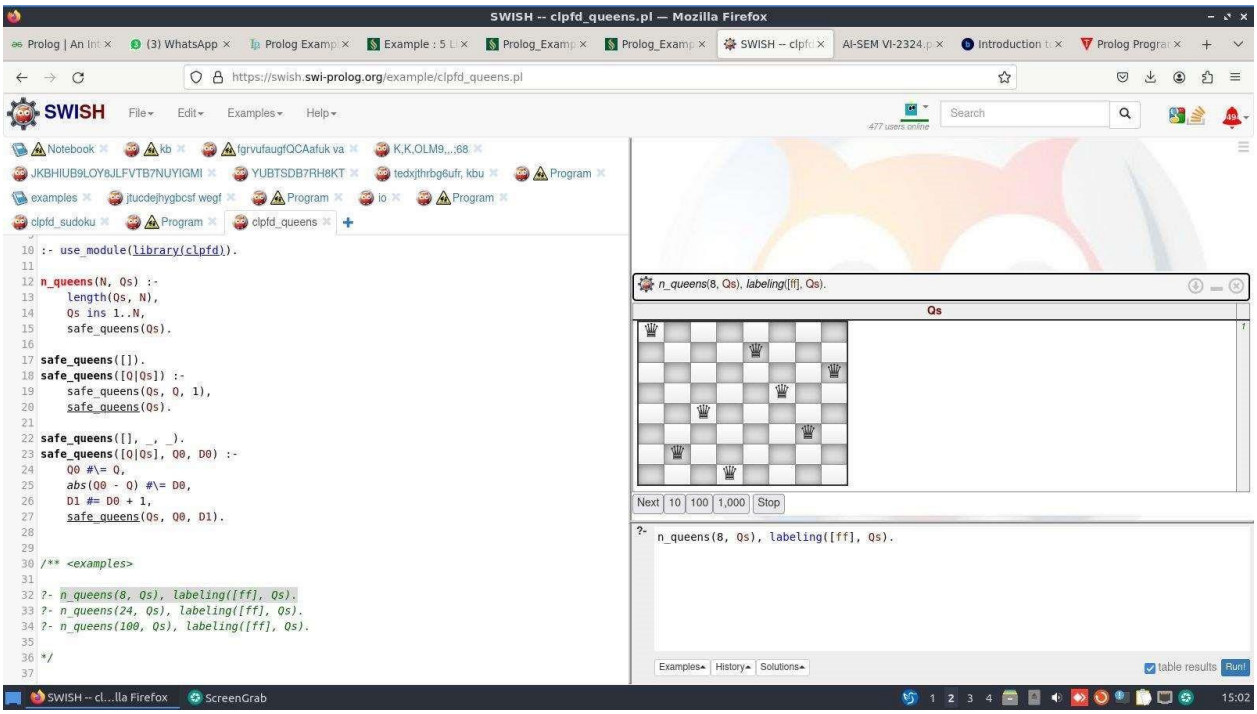
```
%%      n_queens(?N, ?Cols) is nondet.
%
%       @param The k-th element of Cols is the column number of the
%       queen in row k.
%       @author Markus Triska

:- use_module(library(clpfd)).

n_queens(N, Qs) :-
        length(Qs, N),
        Qs ins 1..N,
        safe_queens(Qs).

safe_queens([]).
safe_queens([Q|Qs]) :-
        safe_queens(Qs, Q, 1),
        safe_queens(Qs).

safe_queens([], _, _).
safe_queens([Q|Qs], Q0, D0) :-
        Q0 #\= Q,
        abs(Q0 - Q) #\= D0,
        D1 #= D0 + 1,
        safe_queens(Qs, Q0, D1).


/** <examples>
```

?- n_queens(8, Qs), labeling([ff], Qs).

?- n_queens(24, Qs), labeling([ff], Qs).

?- n_queens(100, Qs), labeling([ff], Qs).

*/



Some more eg: