```
n_queen(4, Solution) :-
      %create a list of 4 dummy  variabiles
      length(Solution, 4),
     queen(Solution, 4). %search for a configuration of 4 queens
    %returns a list of integer from K to 4 included es up2N(1,3,X) X = [1,2,3]
   up2N(N,N,[N]) :-!.
   up2N(K,N,[K|Tail]) :- K < N, K1 is K+1, up2N(K1, N, Tail).
   queen([],_). %No queens is a solution for any 4 queens problem. All queens are in a safe position.
  queen([Q|Qlist],4) :-
  queen(Qlist, 4), %first we solve the subproblem
  %we then generate all possible positions for queen Q
 up2N(1,4,Candidate_positions_for_queenQ),
 %we pick one of such position
 member(Q, Candidate_positions_for_queenQ),
%we check whether the queen Q is safe
 check_solution(Q,Qlist, 1).
check_solution(_,[], _).
check_solution(Q,[Q1|Qlist],Xdist) :-
          Q =\= Q1, %not on the same row
          Test is abs(Q1-Q),
          Test =\= Xdist, %diagonal distance
          Xdist1 is Xdist + 1,
          check_solution(Q,Qlist,Xdist1).
```