```
% Solving nqueens, for SICStus
%                                   (Based on our old example in CHIP)

:- use_module(library(clpfd)).

% nqueens(N,List) - List represents a solution of N queens problem;
%                i-th number in List = the row of the queen in i-th column

nqueens(N,List):-
        length(List,N),
        domain( List, 1, N ),    % X is 1..N for each element of List
% SWI    List ins 1..N,
        all_different( List ),
        constrain_queens(List),
        labeling( [], List ).

% SWI needed   labeling( [ff], List ).
%     ff - The leftmost variable with the smallest domain is selected.

% constrain_queens( L ) - the queens described by L do not atack
%                         each other diagonally

constrain_queens([]).
constrain_queens([X|L]):-
        safe(X,L,1),
        constrain_queens(L).

% safe(X,L,K) - a queen in row X of the current column is not attacked
%        diagonally
%        by the queens described by list L;  K is the distance between
%        the current column and those described by L.

safe(_,[],_).
safe(X,[Y|T],K):-
        noattack(X,Y,K),
        K1 #= K+1,
        safe(X,T,K1).

% noattack(X,L,K) - a queen in row X is not attacked diagonally by
%                   a queen in row Y and K kolumns away

noattack(X,Y,K):-
%       X #\= Y,       % done already by all_different
        Y #\= X+K,
        X #\= Y+K.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/* A query corresponding to slide 8 (with the 6th queen placed)
    length(List,8), domain( List, 1, 8 ), all_different( List ),
    List = [X,Y,Z|T],
    safe(X, [Y,Z|T], 1), safe(Y, [Z|T], 1), safe(Z, T, 1),
    List=[1,3,5|_].
*/

/* Experiments with SWI.   SICStus much more efficient
For query
?- N=8, length(List,N), List ins 1..N,
   all_different( List ), constrain_queens(List), List=[1,3|_].

we get > 50 constraints, including
```

```
    List = [1, 3, _G11670, _G11673, _G11676, _G11679, _G11682, _G11685],
    _G11670 in 5..8,
    _G11673 in 2\/6..8,
    _G11676 in 2\/4\/7..8,
    _G11679 in 2\/4..5\/8,
    _G11682 in 2\/4..6,
    _G11685 in 2\/4..7,
*/
/*
For default labelling this program is slower than a similar Prolog program
(For computing the first answer with N=20, time/1 reports
 119,473,330 inferences, 24s CPU, while for the Prolog program
 114,328,520 inferences, 18s CPU.
 Note however that is/2 is called 1,327,907 times and 19,894,853 times
 respectively.)
 (In safe/3 it was  K1 is K+1,  now it is  K1 #= K+1.)

Labelling  ff  chooses the leftmost variable with smallest domain.
(Under N=20 the first answer after <0.1s CPU, 188,014 inferences,
 is/2 is called 4,486 times.)
*/
```