**8 queen problem**

**Code:**use\_module(library(lists)).

/\*

N-queens problem solved in Prolog (SWI-Prolog or GNU Prolog).

Usage: n\_queen(4,X).

\*/

n\_queen(N, Solution) :-

%create a list of N dummy variabiles

length(Solution, N),

queen(Solution, N). %search for a configuration of N queens

%returns a list of integer from K to N 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 N queens problem. All queens

are in a safe position.

queen([Q|Qlist],N) :-

queen(Qlist, N), %first we solve the subproblem

%we then generate all possible positions for queen Q

up2N(1,N,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

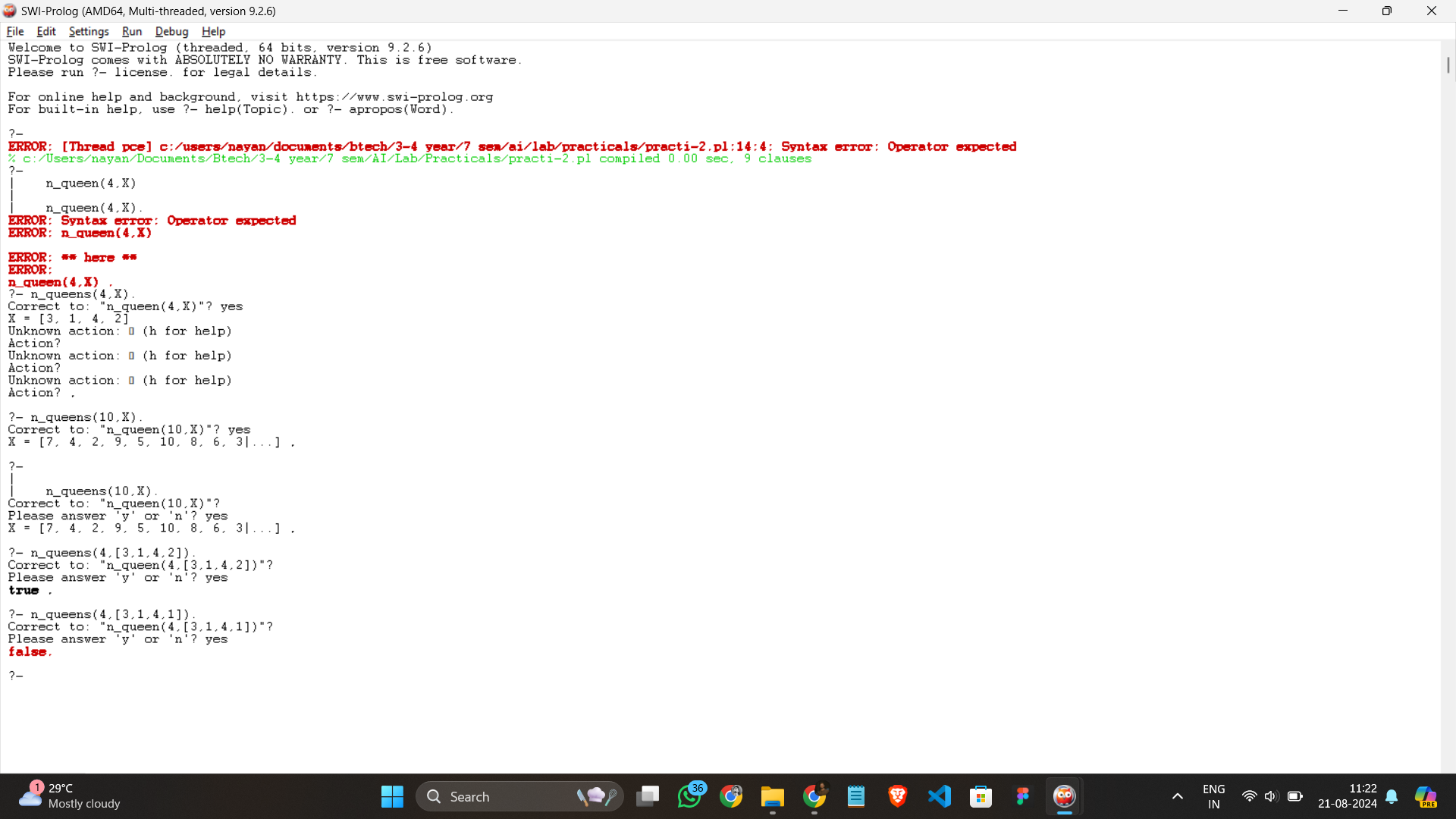
Test is abs(Q1-Q),

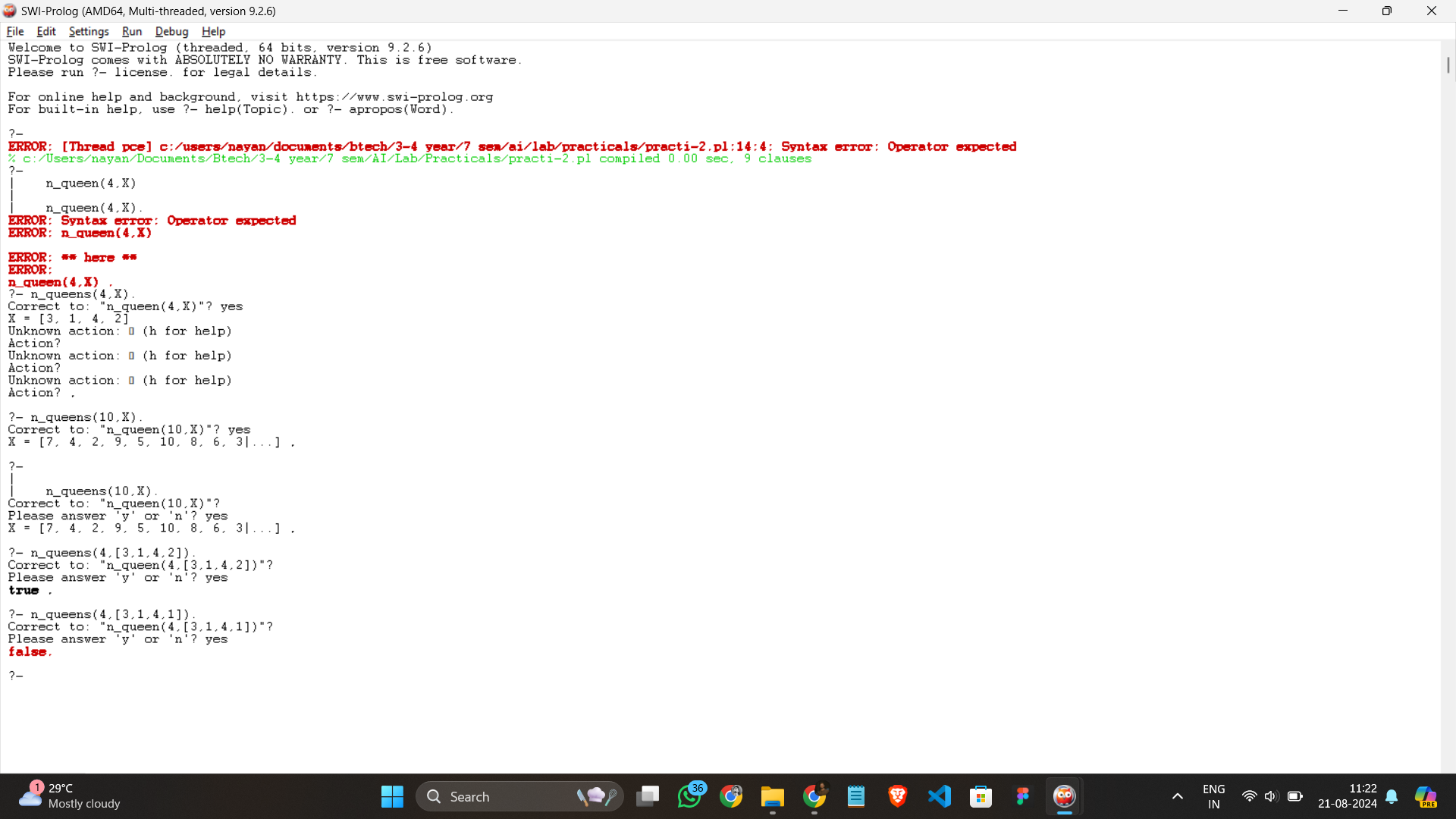
Test =\= Xdist, %diagonal distance

Xdist1 is Xdist + 1,

check\_solution(Q,Qlist,Xdist1).

**Output:**

****

****