```
if hall_wet and kitchen_dry
        then leak_in_bathroom.
if hall_wet and bathroom_dry
        then problem_in_kitchen.
if window_closed or no_rain
        then no_water_from_outside.
if problem_in_kitchen and no_water_from_outside
        then leak_in_kitchen.

start:-abolish(fact/2), enter_facts, forward.

enter_facts:-write('Is the hall wet? yes/no:   '),read(X),write('What is the certainty of that?   '),
read(Y), nl,assert(fact(X,hall_wet:Y)),
write('Is the bathroom dry? yes/no:   '),read(X1),write('What is the certainty of that?   '),
read(Y1),assert(fact(X1,bathroom_dry:Y1)),
write('Is the window closed? yes/no:   '), read(X2), write('What is the certainty of that?   '),
read(Y2), assert(fact(X2,window_closed:Y2)).


forward:- new_derived_fact(fact(yes,P:X)),!,write('Derived - '),
write(P),write(':'), write('\t'), write(X), nl,
assert(fact(yes,P:X)),
forward ; write('No more facts').

new_derived_fact(fact(yes,Concl:X)) :- if Cond then Concl,
not(fact(yes,Concl:X)), composed_fact(fact(yes,Cond:X)).

composed_fact(fact(yes,Cond:X)):- fact(yes,Cond:X).
composed_fact(fact(yes,(Cond1 and Cond2):X)) :-
composed_fact(fact(yes,Cond1:X1)),
composed_fact(fact(yes,Cond2:X2)), min(X1,X2,X).
composed_fact(fact(yes,(Cond1 or Cond2):X)) :-
composed_fact(fact(yes,Cond1:X1)),
composed_fact(fact(yes,Cond2:X2)), max(X1,X2,X).

min(X, Y, X) :- X =< Y, !.
min(X, Y, Y) :- Y =< X.
max(X, Y, Y) :- X =< Y, !.
max(X, Y, X) :- Y =< X.

/* sample execution

?- start.
Is the hall wet? yes/no:   yes.
What is the certainty of that?   80.
Is the bathroom dry? yes/no:   yes.
What is the certainty of that?   70.
Is the window closed? yes/no:   no.
What is the certainty of that?   40.
Derived - problem_in_kitchen:  70
No more facts
*/
```