```
:- style_check(-singleton).
citeste(X,Y):-seeing(A),see('C:/Users/Sandu/Desktop/Project KRR/indp.txt'),
   read(X),read(Y),read(end_of_file),seen,see(A).

get_terms(X,L):-append(X,R),sort(R,L).

negate(n(A), A) :- !.
negate(A, n(A)).

dot_procedure1([],_,[]).
dot_procedure1([H|T],P,R):-member(P,H),dot_procedure1(T,P,R),!.
dot_procedure1([H|T],P,R):-member(n(P),H),dot_procedure1(T,P,R),!.
dot_procedure1([H|T],P,[H|R]):- dot_procedure1(T,P,R).

dot_procedure2([],_,[]).
dot_procedure2([H|T],P,R):-member(P,H),dot_procedure2(T,P,R),!.
dot_procedure2([H|T],P,[R1|R]):-negate(P,Q),
          member(Q,H),
          delete(H,Q,R1),
          dot_procedure2(T,P,R),!.
dot_procedure2([H|T],P,[H|R]):- dot_procedure2(T,P,R).

dot_procedure(X,P,Res):-dot_procedure1(X,P,R1),
        dot_procedure2(X,P,R2),
        union(R1,R2,Res).

inp1([[n(a),b],[c,d],[n(d),b],[n(c),b],[n(b)]]). %no
inp2([[n(b),a],[n(a),b,e],[e],[a,n(e)],[n(a)]]). %no
inp3([[n(a),b],[c,f],[n(f),b],[n(c),b],[n(c)]]). %yes
inp4([[n(a),n(e),b],[n(d),e,n(b)],[n(e),f,n(b)],[f,n(a),e],[e,f,n(b)]]). %yes
inp5([[a,b],[n(a),n(b)],[n(a),b],[a,n(b)]]). %no

choose_p(X,P):-
   findall([Len, Q], (
      member(C,X),
      member(Q,C),
      length(C,Len)
   ), Lista_lungimi),
   sort(Lista_lungimi, Lista_sortata),
   Lista_sortata=[[_,P]|_].

%choose_p2(X,P):-member([P],X),!;[H|_]=X,[P|_]=H.
choose_p2(X,P):-
   get_terms(X,L),
   member(P,L),
   negate(P,Q),
   \+member(Q,L),!;
   [H|_]=X,
   [P|_]=H.

:-dynamic adev/1.
%afisadev(_,_):-forall(adev(A), (write(A),write(' = true, '))).
```

```
afisadev(A,_):-forall(adev(A), (write(A),write(' = true, '))),nl,!.

dp([]):-write('YES\n'),afisadev(_,_),!.
dp(X):-member([],X),write('No\n'),!.
dp(X):-
    choose_p(X,P),
    dot_procedure(X,P,Res),
    assertz(adev(P)),
    dp(Res),!;
    negate(P,Q),
    dot_procedure(X,Q,Res2),
    retract(adev(P)),
    assertz(adev(Q)),
    dp(Res2),!.

dp2([]):-write('YES\n'),afisadev(_,_),!.
dp2(X):-member([],X),write('No\n'),!.
dp2(X):-
    choose_p2(X,P),
    dot_procedure(X,P,Res),
    assertz(adev(P)),
    dp2(Res),!;
    negate(P,Q),
    dot_procedure(X,Q,Res2),
    retract(adev(P)),
    assertz(adev(Q)),
    dp(Res2),!.

solve():-citeste(X,Y),telling(A),tell('C:/Users/Sandu/Desktop/Project KRR/outdp.txt'),
      retractall(adev(_)),dp(X),retractall(adev(_)),dp(Y),told,tell(A).
```

```
:- style_check(-singleton).
stream('C:/Users/Sandu/Desktop/Project KRR/a.txt').
write_to_file(Stream, Message):-
                        telling(OldStream),tell(Stream),
                        write(Message),
                        told,tell(OldStream).

read_from_file(Stream, X):-
                        seeing(OldStream),see(Stream),
                        read(X),
                        read(end_of_file),
                        seen,see(OldStream).

wtf(Message):-
            telling(OldStream),tell('C:/Users/Sandu/Desktop/Project KRR/b.txt'),
            write(Message),
            told,tell(OldStream).

rff(X):-
            seeing(OldStream),see('C:/Users/Sandu/Desktop/Project KRR/a.txt'),
            read(X),
            read(end_of_file),
            seen,see(OldStream).

resolve(C1,C2,P,Res) :-
      member(P, C1),
      negate(P,Q),
      member(Q, C2),
      delete(C1,P,D1),
      delete(C2,Q,D2),
      subtract(D1,D2,D),
      merge(D,D2,Res),
      !.

resolve(C1,C2,P,Res) :- member(P, C2),
                                    negate(P,Q),
                                    member(Q, C1),
                                    delete(C1,Q,D1),
                                    delete(C2,P,D2),
                                    subtract(D1,D2,D),
                                    merge(D,D2,Res),
                              %dif(Res,[]),
                                    !.


add_to_formula(Res,X1,X):-
                                    \+member(Res,X1),
                                    merge([Res],X1,X).

negate_all([],R,R).
negate_all([H|T],R,R1):-negate(H,P),negate_all(T,[P|R],R1).

get_terms(X,L):-append(X,R),negate_all(R,[],R1),merge(R1,R,R2),sort(R2,L).

search_formula(_,[],_,_,_).
search_formula(P,[C1|T],C1,X,C2):- member(P,C1),search_formula2(n(P),X,C2).
search_formula(P,[H|T],C1,X,C2):- search_formula(P,T,C1,X,C2).

search_formula2(_,[],_):-false.
search_formula2(P,[C1|T],C1):- member(P,C1),!.
search_formula2(P,[H|T],C1):- search_formula2(P,T,C1).
```

```
negate(n(A), A) :- !.
negate(A, n(A)).

rec([[]|_]):-!,wtf('UNSATISFIABLE'),nl,!,false.
rec(X):-
    (search_formula(H,X,C1,X,C2),
    %get_terms(X,L),
    %member(H,L),
    member(C1,X),
    member(C2,X),
    resolve(C1,C2,H,Res),
      add_to_formula(Res,X,F),!;wtf('SATISFIABLE'),nl,!,false),
      rec(F).

solve(X):- rff(X),rec(X).
```

```
:- style_check(-singleton).
:-dynamic answer/1.
question1():-
    write('What is patient temperature? (answer is a number)'),read(Inp),
 (Inp = 'stop',abort,!;Inp>38,assertz(answer([temperature])),!;!).

question2():-
    write('For how many days has the patient been sick? (answer is a number)'),read(Inp),
 (Inp = 'stop',abort,!;Inp>=2,assertz(answer([sick])),!;!).

question3():-
    write('Has patient cough? (answer is yes/no)'),read(Inp),
 (Inp = 'stop',abort,!;Inp = 'yes',assertz(answer([cough])),!;!).

questions(Out):- question1(),question2(),question3(),findall(X,answer(X),Out).

read_file(X):-
  seeing(OldStream),see('C:/Users/Sandu/Desktop/input5.txt'),
  read(X),read(end_of_file),
  seen,see(OldStream).

negate(n(A), A) :- !.
negate(A, n(A)).

negate_all([],R,R).
negate_all([H|T],R,R1):-negate(H,P),negate_all(T,[P|R],R1).

member_all([],R).
member_all([H|T],R):-member(H,R),member_all(T,R).

backchaining([],_):-write('backchaining: Pacient has pneumonia'),nl,!.
backchaining([H|T],Kb):-
    member(R,Kb),member(H,R),
    delete(R,H,R1),
    negate_all(R1,[],R2),
    append(R2,T,R3),
    backchaining(R3,Kb).

forwardchaining(G,S,Kb):-member(pneumonia,S),write('forwardchaining: Pacient has pneumonia'),nl,!.
forwardchaining(G,S,Kb):-
    member(R,Kb),member(H,R),
    delete(R,H,R1),
    negate_all(R1,[],R2),
    member_all(R2,S),
    \+member(H,S),
    append([H],S,S1),
    forwardchaining(G,S1,Kb).


solve():- read_file(F), questions(A), append(F,A,Kb),flatten(A,S),
 (backchaining([pneumonia],Kb),!;write('backchaining: Pacient DOESNT have pneumonia'),nl),
 (forwardchaining(pneumonia,S,Kb),!;write('forwardchaining: Pacient DOESNT have pneumonia'),nl),nl,
```

```
retractall(answer(_)),
solve.
```

```
:- style_check(-singleton).
:-dynamic wm/1.

question1():-
   write('What is patient temperature? (answer is a number)'),
   read(Inp),(Inp='stop',abort,!;Inp>38,assertz(wm(temperature)),!;!).

question2():-
   write('For how many days has the patient been sick? (answer is a number)'),
   read(Inp),(Inp='stop',abort,!;Inp>=2,assertz(wm(sick)),!;!).

question3():-
   write('Has patient muscle pain? (answer is yes/no)'),
   read(Inp),(Inp='stop',abort,!;Inp='yes',assertz(wm(muscle_pain)),!;!).

question4():-
   write('Has patient cough? (answer is yes/no)'),
   read(Inp),(Inp='stop',abort,!;Inp='yes',assertz(wm(cough)),!;!).


read_file(X):-
  seeing(OldStream),see('C:/Users/Sandu/Desktop/input6.txt'),
  read(X),
  read(end_of_file),
  seen,see(OldStream).

write_file(Out1,Out2,Out3):-
  telling(OldStream),tell('C:/Users/Sandu/Desktop/output6.txt'),
  write(Out1),nl,
  write(Out2),nl,
  write(Out3),nl,
  told,tell(OldStream).


questions:- question1,question2,question3,question4.

asses([]).
asses([H|T]):-wm(H),asses(T).

my_if(R):- /*if*/ R=[H|_],asses(H).

my_then(R):- /*then*/ R=[_|T],T=[[H|_]|_],\+wm(H),assertz(wm(H)).

run(File):-member(R,File), my_if(R),my_then(R),run(File),\+wm(pneumonia);!.

show(Out):-findall(X, wm(X), Out).

diagnostic():-wm(pneumonia),write('YES'),nl,!;write('NO'),nl,!.

reset_wm():- retractall(wm(_)).

execute:-
```

```
read_file(File),
show(Initial),questions,
show(Questions),run(File),
show(Done),diagnostic,
write_file(Initial,Questions,Done),
reset_wm,
execute.
```