```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CS_125 Logic Programming
% Spring 2002
% Solutions to Exercises 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% (a), (b)

move(conf(N,P),conf(M,Q)) :-
  other(P,Q),
  (M is N+1 ; M is N+2).

other(a,b).
other(b,a).


% (c)

won(C) :-
  not(terminal_lost(C)),
  move(C,D),
  not(won(D)).

terminal_lost(conf(N,_)) :- N >= 10.


% (d)

select_move(C, D) :-
    C = conf(N,a),
    write('player a: '),
    read(X),
    M is N+X,
    D = conf(M,b).

select_move(C, D) :-
    C = conf(M,b),   %Betti (i.e., the computer)
    move(C, D),      %tries to find a move leading
    not(won(D)), !,  %to a winning conf, i.e. a
    D = conf(N,a),   %loosing conf. for Alex
    K is N-M,
    write('player b plays: '),
    write(K), nl.

select_move(C, D) :-  % otherwise,
    C = conf(N,b),    % Betti asks for help
    write('please help player b: '),
    read(X),
    M is N+X,
    D = conf(M,a).


% (e)

go(P) :-
    C = conf(0,P),
    display_conf(C), nl,
    run(conf(0,P)).

run(C) :-
    terminal_lost(C), !,
    C = conf(_,P),
    other(P, Q),
    write('player '),
    write(Q),
    write(' has won').

run(C) :-
    select_move(C, D),
    display_conf(D), nl,
    run(D).

display_conf(C) :- write(C).



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 2.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% (a)

all_permutations(L,PP) :-
   findall(P,permutation(L,P),PP).

permutation([],[]).
permutation(L,[X|P]) :-
  select(X,L,L1),
  permutation(L1,P).

select(X,[X|L],L).
select(X,[Y|L],[Y|L1]) :- select(X,L,L1).


% (b)

intersect(L1,L2,L) :-
  setof(X,(member(X,L1),member(X,L2)),L), !.

intersect(_,_,[]).


% (c)

setminus(L1,L2,L) :-
  setof(X,(member(X,L1),not(member(X,L2))),L), !.

setminus(_,_,[]).
```