```
/* De facut si la grupa 144 corectura predicatului inversa (trebuie
scris fara !), operatorul zeroar cputime, a doua diferenta dintre
bagof si findall, cuantificarea existentiala in setof si bagof si
produsul cartezian definit recursiv, fara metapredicate:
?- Init is cputime, ciur(10000,L), Fin is cputime, write(L), Dif is
Fin-Init, nl, write(Dif), tab(1), write(secunde).
?- setof(X, member((X,Y),[(a,1),(b,1),(a,2),(c,3),(c,3),(d,3)]), L).
?- bagof(X, member((X,Y),[(a,1),(b,1),(a,2),(c,3),(c,3),(d,3)]), L).
?- findall(X, member((X,Y),[(a,1),(b,1),(a,2),(c,3),(c,3),(d,3)]),
L).
?- bagof(X, Y^member((X,Y),[(a,1),(b,1),(a,2),(c,3),(c,3),(d,3)]),
L).
?- setof(X, Y^member((X,Y),[(a,1),(b,1),(a,2),(c,3),(c,3),(d,3)]),
L).
*/

inversa([],[]).
inversa([H|T],L) :- inversa(T,M), append(M,[H],L).

%%%%%%

prodcart([],_,[]).
prodcart([H|T],L,P) :- prodsgl(H,L,Q), prodcart(T,L,R),
```

```
append(Q,R,P).

prodsgl(_,[],[]).
prodsgl(H,[K|T],[(H,K)|U]) :- prodsgl(H,T,U).

prodcartmult(L,M,LxM) :- prodcart(L,M,P), elimdupl(P,LxM).

% De facut si la grupele de JOI afisarea unei liste cu fiecare
element pe alt rand:

afislista([]).
afislista([H|T]) :- write(H), nl, afislista(T).
```