```
/* 1 */


gxd(X,0,R) :- (X<0,R is -X; X>=0,R is X).
gxd(0,X,R) :- (X<0,R is -X; X>=0,R is X).
gxd(X,Y,R) :- X<0, !, XX is -X, gxd(XX,Y,R).
gxd(X,Y,R) :- Y<0, !, YY is -Y, gxd(X,YY,R).
gxd(X,X,X) :- !.
gxd(X,Y,R) :- (X<Y,!,YY is Y-X,gxd(X,YY,R); XX is X-Y,gxd(XX,Y,R)).

/* 2 */

app([],X,X).
app([H|T],L,[H|Z]) :- app(T,L,Z).

rev([],[]).
rev([H|T],R) :-
  rev(T,RT),
  app(RT,[H],R).

sum(A,B,C) :-
  rev(A,RA), rev(B,RB),
  add(RA,RB,0,RC),
  rev(RC,C).

add([],[],1,[1]).
add([],[],0,[]).
add([],[H|T],1,R) :- add([1],[H|T],0,R).
add([],[H|T],0,[H|T]).
add([H|T],[],X,R) :- add([],[H|T],X,R).
add([A|S],[Q|W],X,[V|B]) :-
  VAL is A+Q+X,
  (VAL < 2,V=VAL,add(S,W,0,B);
    (VAL == 2, V=0, add(S,W,1,B);
     VAL == 3, V=1, add(S,W,1,B))).

/* 3 */

/*

lvar(NAME).     promenna
lapp(E1,E2).    aplikace
labs(NAME,E).   abstrakce

*/

freeVars(lvar(V),[V]) :- !.
freeVars(labs(N,E),R) :-
  !,
  freeVars(E,EF),
  delet(N,EF,R).
freeVars(lapp(E1,E2),R) :-
  !,
  freeVars(E1,F1),
  freeVars(E2,F2),
  union(F1,F2,R).

delet(_,[],[]) :- !.
delet(X,[X|T],R) :-
  !,delet(X,T,R).
delet(X,[H|T],[H|R]) :-
  !,delet(X,T,R).

/* member byl povolen, zde nahrazeno elem */
elem(_,[]):- !,fail.
elem(X,[X|_]) :- !.
elem(X,[_|T]) :- elem(X,T).

union([],L,L).
union([X|T],L,U) :-
  elem(X,L),!,
  union(T,L,U).
union([X|T],L,[X|U]) :-
  union(T,L,U).

/* 4 */


search(P,LL) :-
  retractall(pos(_)),
  bagof(L,track(P,P,0,L),LL).

track(P,P,N,[P]) :- N >= 20, N =< 22, !.
track(P,P,N,_) :- (N>0,!, fail;N==0,fail).
track(A,P,N,[A|T]) :-
  N < 22,
  assertz(pos(A)),
  nextStep(A,B),
  (not(pos(B));B==P),
  NN is N+1,
  track(B,P,NN,T).
track(A,_,_,_) :-
  pos(A),
  retract(pos(A)),
  !, fail.


/* toto nebylo treba */
```

```
nextStep(p(X,Y),p(XX,YY)) :-
  move(I,J),
  XX is X+I,
  YY is Y+J,
  XX > 0, YY > 0,
  XX < 9, YY < 9.

move(1,0).
move(0,1).
move(-1,0).
move(0,-1).

/* EOF */
```