

```

:- dynamic
    pos/1.

/* ----- */

mkTrans([],[]).
mkTrans([[]|_],[]).
mkTrans(LS,[HS|HHS]) :-
    trans(LS,HS,TS),
    mkTrans(TS,HHS).

trans([],[],[]).
trans([H|T]|LS],[H|HS],[T|TS]) :-
    trans(LS,HS,TS).

/* ----- */

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

subseq(S,[[]|SS]) :-
    suff(S,SUFS),
    proc(SUFS,SS).

proc([],[]).
proc([S|SS],RES) :-
    pref(S,[_]|PS),
    proc(SS,PSS),
    append(PS,PSS,RES).

pref([H|T],[[]|R]) :-
    pref(T,PS),
    prepAll(H,PS,R).
pref([],[][]).

prepAll(_,[],[]).
prepAll(X,[L|LS],[[X|L]|XS]) :-
    prepAll(X,LS,XS).

/* ----- */

search(S,E,Res) :-
    retractall(pos(_)),
    steptry(S,E,0,Res).

steptry(S,E,N,Res) :-
    s(S,E,N,Res), !.
steptry(S,E,N,Res) :-
    NN is N+1,
    steptry(S,E,NN,Res).

s(E,E,0,[E]) :- !.
s(_,_,N,_) :- N < 0, !, fail.
s(A,E,N,[A|R]) :-
    assertz(pos(A)),
    NN is N-1,
    nextStep(A,AA),
    not(pos(AA)) ,
    s(AA,E,NN,R).
s(A,_,_,_) :-
    pos(A),
    retract(pos(A)),
    !,fail.

nextStep(p(X,Y),p(XX,Y)) :- XX is X+1.
nextStep(p(X,Y),p(XX,Y)) :- XX is X-1.
nextStep(p(X,Y),p(X,YY)) :- YY is Y+1.
nextStep(p(X,Y),p(X,YY)) :- YY is Y-1.

/* ----- */

/* rac(numerartor,denominator) */

```

```

/* op( '+',L,R).
 * op( '*' ,L,R).
 * rac(N,D).
 */

gcd(N,N,N) :- !.
gcd(N,M,M) :-
    N > M,
    NN is mod(N,M),
    NN==0, !.
gcd(N,M,D) :-
    N > M, !,
    NN is mod(N,M),
    gcd(M,NN,D).
gcd(N,M,N) :-
    M > N,
    MM is mod(M,N),
    MM is 0, !.
gcd(N,M,D) :-
    MM is mod(M,N),
    gcd(N,MM,D).

ev(op( '+',L,R),rac(NN,DD)) :-
    ev(L,rac(LN,LD)),
    ev(R,rac(RN,RD)),
    N1 is LN*RD + RN*LD,
    D1 is LD*RD,
    norm(N1,D1,NN,DD).
ev(op( '*' ,L,R),rac(NN,DD)) :-
    ev(L,rac(LN,LD)),
    ev(R,rac(RN,RD)),
    N1 is LN*RN,
    D1 is LD*RD,
    norm(N1,D1,NN,DD).
ev(rac(X,Y),rac(X,Y)).

norm(N,D,NN,DD) :-
    gcd(N,D,G), G>1,!,
    NN is div(N,G),
    DD is div(D,G).
norm(N,D,N,D).

/* EOF */

```