

```

#####
% 1. Prvek P je členen seznamu S.
% Použítí: member(P, S).

member(P, [_T]) :- member(P, T).

#####
% 2. Ověření, že seznam S je seznamem.
% Použítí: jeSeznam(S).

jeSeznam([]).
jeSeznam(_).

#####
% 3. Vypište první prvek P seznamu S.
% Použítí: první(S, P).

první([], false).
první([_H_], H).

#####
% 4. Vypište předposlední prvek P seznamu S.
% Použítí: předposlední(S, P).

předposlední([], false).
předposlední([], false).
předposlední([_H_], H) :- posledníP(T).
předposlední([_T], P) :- předposlední(T, P).

posledníP([_]) :- true, !.
posledníP([_]) :- false.

#####
% 5. Vypište poslední prvek P seznamu S.
% Použítí: poslední(S, P).

poslední([], false).
poslední([_H_], H).
poslední([_T], P) :- poslední(T, P).

#####
% 6. Vložte prvek P na začátek seznamu S.
% Použítí: vlozZ(P, S, new_S).

vlozZ(P, S, S) :- member(P, S), !.
vlozZ(P, S, [_P|S]).

#####
% 7. Vložte prvek P na konec seznamu S.
% Použítí: vlozK(P, S, new_S).

vlozK(P, [], [_P]).
vlozK(P, [_H_], [_H_|T2]) :- vlozK(P, T1, T2).

#####
% 8. Smažte prvek P ze seznamu S.

```

```

% Použítí: smaz(P, S, new_S).

%smaz(P,[], []).
smaz(P,[P|T], T) :- !,
    smaz(P, [H|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 9. Vypíšte počet prvků N lineárního seznamu S.
% Použítí: pocetL(S, N).

pocetL([], 0).
pocetL([_|T], N) :- pocetL(T, N1), N is 1 + N1.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 10. Vypíšte počet atomických prvků N seznamu S.
% Použítí: pocetA(S, N).

pocetA([], 0).
pocetA([H|T], N) :- jeSeznam(H), pocetA(H, N1), pocetA(T,
N2), N is N1 + N2.
pocetA([_|T], N) :- pocetA(T, N1), N is N1 + 1.

%jeSeznam([]).
%jeSeznam([_|_]).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 11. Vypíšte nejmenší prvek P seznamu S.
% Použítí: minimum(S, P).

minimum([], false).
minimum([P], P).
minimum([H1,H2|T], P) :- H1 <= H2, minimum([H1|T], P).
minimum([H1,H2|T], P) :- H1 > H2, minimum([H2|T], P).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 12. Vypíšte největší prvek P seznamu S.
% Použítí: maximum(S, P).

maximum([], false).
maximum([P], P).
maximum([H1,H2|T], P) :- H1 >= H2, maximum([H1|T], P).
maximum([H1,H2|T], P) :- H1 < H2, maximum([H2|T], P).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 13. Vypíšte součet nejmenšího a největšího prvku P seznamu
S.
% Použítí: soucetMinMax(S, MinMax).

soucetMinMax([], 0).
soucetMinMax(S, MinMax) :- minimum(S, Min), maximum(S, Max),
    MinMax is Min + Max.

%minimum([], false).
%minimum([P], P).
%minimum([H1,H2|T], P) :- H1 <= H2, minimum([H1|T], P).
%minimum([H1,H2|T], P) :- H1 > H2, minimum([H2|T], P).

%maximum([], false).

```

```

%maximum([P], P).
%maximum([H1,H2|T], P) :- H1 >= H2, maximum([H1|T], P).
%maximum([H1,H2|T], P) :- H1 < H2, maximum([H2|T], P).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 14. Vypíšte součet nejmenšího a druhého nejmenšího prvku P
% seznamu S.
% Použítí: soucetMinMin(S, MinMin).

soucetMinMin([], 0).
soucetMinMin(S, MinMin) :- minimum(S, Min1), smaz(Min1, S,
S1), minimum(S1, Min2), MinMin is Min1 + Min2.

%minimum([], false).
%minimum([P], P).
%minimum([H1,H2|T], P) :- H1 <= H2, minimum([H1|T], P).
%minimum([H1,H2|T], P) :- H1 > H2, minimum([H2|T], P).

%%smaz(P, [], []).
%smaz(P, [P|T], T) :- !.
%smaz(P, [_|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 15. Vypíšte součet největšího a druhého největšího prvku P
% seznamu S.
% Použítí: soucetMaxMax(S, MaxMax).

soucetMaxMax([], 0).
soucetMaxMax(S, MaxMax) :- maximum(S, Max1), smaz(Max1, S,
S1), maximum(S1, Max2), MaxMax is Max1 + Max2.

%maximum([], false).
%maximum([P], P).
%maximum([H1,H2|T], P) :- H1 >= H2, maximum([H1|T], P).
%maximum([H1,H2|T], P) :- H1 < H2, maximum([H2|T], P).

%%smaz(P, [], []).
%smaz(P, [P|T], T) :- !.
%smaz(P, [_|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 16. Vypíšte součet všech prvků P v seznamu S.
% Použítí: soucetAll(S, A).

soucetAll([], 0).
soucetAll([_], A).
soucetAll([H|T], A) :- soucetAll(T, A1), A is A1 + H.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 17. Vypíšte index nejmenšího prvku P seznamu S.
% Použítí: indexMin (S, I).

indexMin([], 0).
indexMin([H|T], I) :- min(T, H, M), M = H, I is 1, !.
indexMin([_|T], I) :- indexMin(T, I1), I is I1 + 1.

min([H2|T], H1, M) :- H2 <= H1, min(T, H2, M).
min([H2|T], H1, M) :- H2 > H1, min(T, H1, M).

```

```

min([], M, M).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 18. Vypíšte index největšího prvku P seznamu S.
% Použítí: indexMax(S, I).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
indexMax([], 0).
indexMax([H|T], I) :- max(T, H, M), M = H, I is 1, !.
indexMax([_|T], I) :- indexMax(T, I1), I is I1 + 1.

max([H2|T], H1, M) :- H2 >= H1, max(T, H2, M).
max([H2|T], H1, M) :- H2 < H1, max(T, H1, M).
max([], M, M).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 19. Zjistěte, zda je seznam S monotónní, tzn. jestli je
% rostoucí, popř. klesající.
% Použítí: monotonnost(S).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
monotonnost(S) :- rostouci(S) ; klesajici(S).

rostouci([]).
rostouci([_|]).
rostouci([H1,H2|T]) :- H1 <= H2, rostouci([H2|T]).

klesajici([]).
klesajici([_|]).
klesajici([H1,H2|T]) :- H1 >= H2, klesajici([H2|T]).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 20. Seřadte seznam od nejmenšího prvku po největší.
% (Nefunguje korektně.)
% Použítí: seradMinMax(S, new_S).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
seradMinMax([], []).
seradMinMax([_|], S3).
seradMinMax(S1, S3) :- minimum(S1, P), vložz(P, S3, S3),
smaz(P, S1, S2), seradMinMax(S2, S3).

%minimum([], false).
%minimum([P], P).
%minimum([H1,H2|T], P) :- H1 <= H2, minimum([H1|T], P).
%minimum([H1,H2|T], P) :- H1 > H2, minimum([H2|T], P).

%vložz(P, S, S) :- member(P, S), !.
%vložz(P, S, [P|S]).

%smaz(P, [], []).
%smaz(P, [P|T], T) :- !.
%smaz(P, [_|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 21. Seřadte seznam od největšího prvku po nejmenší.
% (Nefunguje korektně.)
% Použítí: seradMaxMin(S, new_S).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
seradMaxMin([], []).
seradMaxMin([_|], S3).

```

```

%seradMaxMin(S1, S3) :- maximum(S1, P), vlozZ(P, S3, S3),
smaz(P, S1, S2), seradMaxMin(S2, S3).

%%maximum([], false).
%%maximum([P], P).
%%maximum([H1,H2|T], P) :- H1 >= H2, maximum([H1|T], P).
%%maximum([H1,H2|T], P) :- H1 < H2, maximum([H2|T], P).

%%vlozZ(P, S, S) :- member(P, S), !.
%%vlozZ(P, S, [P|S]).

%%smaz(P, [], []).
%%smaz(P, [P|T], T) :- !.
%%smaz(P, [H|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 22. Sjednodte 2 množiny K, L a vraťte množinu M.
% Použiti: sjednoceni2(K, L, M).

sjednoceni2([], L, L).
sjednoceni2([H|T], L, M) :- sjednoceni2(T, L, M1), vlozZ(H,
M1, M).

%vlozZ(P, S, S) :- member(P, S), !.
%vlozZ(P, S, [P|S]).

%member(P, [P|_]).
%member(P, [_|T]) :- member(P, T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 23. Sjednodte 3 množiny J, K, L a vraťte množinu M.
% Použiti: sjednoceni3(J, K, L, M).

sjednoceni3(J, K, L, M) :- sjednoceni2(J, K, M1), sjednoceni2
(M1, L, M).

%sjednoceni2([], L, L).
%sjednoceni2([H|T], L, M) :- sjednoceni2(T, L, M1), vlozZ(H,
M1, M).

%vlozZ(P, S, S) :- member(P, S), !.
%vlozZ(P, S, [P|S]).

%member(P, [P|_]).
%member(P, [_|T]) :- member(P, T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 24. Vytvořte průnik 2 množin K, L a vraťte množinu M.
% Použiti: prunik2(K, L, M).

prunik2([], _, []).
prunik2([H|K], L, [H|M]) :- member(H, L), prunik2(K, L, M).
prunik2([_|K], L, M) :- prunik2(K, L, M).

%member(P, [P|_]).
%member(P, [_|T]) :- member(P, T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% 25. Vytvořte průnik 3 množin J, K, L a vraťte množinu M.
% Použítí: prunik3(J, K, L, M).

prunik3([],_,_,[]).
prunik3([H|J], K, L, [H|M]) :- member(H, K), member(H, L),
prunik3(J, K, L, M).
prunik3([_|J], K, L, M) :- prunik3(J, K, L, M).

%member(P, [P|_]).
%member(P, [_|_]) :- member(P, _).

#####
% 26. Vytvořte sjednocení 2 množin K a L, přičemž množina K
% vznikla pruníkem množiny I a J. Vraťte množinu M.
% Použítí: sjednoceni(I, J, L, M).

sjednoceni(I, J, L, M) :- prunik2(I, J, K), sjednoceni2(K, L,
M).

%sjednoceni2([], L, L).
%sjednoceni2([H|T], L, M) :- sjednoceni2(T, L, M1), vlozZ(H,
M1, M).

%vlozZ(P, S, S) :- member(P, S), !.
%vlozZ(P, S, [P|S]).

%member(P, [P|_]).
%member(P, [_|_]) :- member(P, _).

%prunik2([H|K], L, [H|M]) :- member(H, L), prunik2(K, L, M).
%prunik2([_|K], L, M) :- prunik2(K, L, M).

#####
% 27. Vytvořte průnik 2 množin K a L, přičemž množina K
% vznikla
% sjednocením množiny I a J. Vraťte množinu M.
% Použítí: prunik(I, J, L, M).

prunik(I, J, L, M) :- sjednoceni2(I, J, K), prunik2(K, L, M).

%sjednoceni2([], L, L).
%sjednoceni2([H|T], L, M) :- sjednoceni2(T, L, M1), vlozZ(H,
M1, M).

%vlozZ(P, S, S) :- member(P, S), !.
%vlozZ(P, S, [P|S]).

%member(P, [P|_]).
%member(P, [_|_]) :- member(P, _).

%prunik2([H|K], L, [H|M]) :- member(H, L), prunik2(K, L, M).
%prunik2([_|K], L, M) :- prunik2(K, L, M).

#####
% 28. Vypočítejte faktoriál z čísla F a vraťte N.
% Použítí: faktorial(F, N).

faktorial(0, 1).

```



```

faktorial(F, false) :- F < 0.
faktorial(F, N) :- F is F - 1, faktorial(F1, N1), N is F *
N1.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Něco navíc:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% 29. Generování čísel do seznamu S ze zadaného intervalu.
% Použiti: generuj(od, do, S).

generuj(M, N, []) :- M > N.
generuj(M, N, [M|S]) :- M =< N, M1 is M + 1, generuj(M1, N,
S).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 30. Obrátte seznam S.
% Použiti: obrat(S, new_S).

obrat(S1, S2) :- obrat(S1, [], S2).
obrat([], P, P).
obrat([H|T], P, S) :- obrat(T, [H|P], S).

```