```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1. Prvek P je členen seznamu S.
% Použití: member(P, S).

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

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

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3. Vypište první prvek P seznamu S.
% Použití: prvni(S, P).

prvni([], false).
prvni([H|_], H).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4. Vypište předposlední prvek P seznamu S.
% Použití: predposledni(S, P).

predposledni([], false).
predposledni([_], false).
predposledni([H|T], H) :- posledniP(T).
predposledni([_|T], P) :- predposledni(T, P).

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 5. Vypište poslední prvek P seznamu S.
% Použití: posledni(S, P).

posledni([], false).
posledni([H|[]], H).
posledni([_|T], P) :- posledni(T, P).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 6. Vložte prvek P na začátek seznamu S.
% Použití: 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.
% Použití: vlozK(P, S, new_S).

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

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

```
% Použití: smaz(P, S, new_S).

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 9. Vypište počet prvků N lineárního seznamu S.
% Použití: pocetL(S, N).

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 10. Vypište počet atomických prvků N seznamu S.
% Použití: 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. Vypište nejmenší prvek P seznamu S.
% Použití: 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. Vypište největší prvek P seznamu S.
% Použití: 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. Vypište součet nejmenšího a největšího prvku P seznamu
S.
% Použití: 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. Vypište součet nejmenšího a druhého nejmenšího prvku P
% seznamu S.
% Použití: 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, [H|T1], [H|T2]) :- smaz(P, T1, T2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 15. Vypište součet největšího a druhého největšího prvku P
% seznamu S.
% Použití: 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, [H|T1], [H|T2]) :- smaz(P, T1, T2).

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

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 17. Vypište index nejmenšího prvku P seznamu S.
% Použití: 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. Vypište index největšího prvku P seznamu S.
% Použití: 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.
% Použití: 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. Seřaďte seznam od nejmenšího prvku po největší.
% (Nefunguje korektně.)
% Použití: seradMinMax(S, new_S).

%seradMinMax([], []).
%seradMinMax([], S3).
%seradMinMax(S1, S3) :- minimum(S1, P), vlozZ(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).

%%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).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 21. Seřaďte seznam od největšího prvku po nejmenší.
% (Nefunguje korektně.)
% Použití: 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. Sjednoďte 2 množiny K, L a vraťte množinu M.
% Použití: 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. Sjednoďte 3 množiny J, K, L a vraťte množinu M.
% Použití: 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.
% Použití: 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.
% Použití: 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, [_|T]) :- member(P, T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 26. Vytvořte sjednoceni 2 množin K a L, přičemž množina K
% vznikla prunikem množiny I a J. Vraťte množinu M.
% Použití: 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, [_|T]) :- member(P, T).

%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
% sjednocenim množiny I a J. Vraťte množinu M.
% Použití: 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, [_|T]) :- member(P, T).

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 28. Vypočtěte faktoriál z čísla F a vraťte N.
% Použití: faktorial(F, N).

faktorial(0, 1).
```

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Něco navíc:                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 29. Generování čísel do seznamu S ze zadaného intervalu.
% Použití: 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. Obraťte seznam S.
% Použití: obrat(S, new_S).

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