

## % Převod seznamu na množinu (odstranění duplikátů)

```
je_clen(X, [X|_]) :- !.
je_clen(X, [_|T]) :- je_clen(X, T).

seznam_do_mnoziny([], []).
seznam_do_mnoziny([H|T], Y) :- je_clen(H, T), seznam_do_mnoziny(T, Y).
seznam_do_mnoziny([H|T], [H|Y]) :- seznam_do_mnoziny(T, Y).
```

## % Zjistit, zda průnik S1 a S2 je podmnožinou S3

```
prunik2([], _, []).
prunik2([H|L1], L2, [H|XTAIL]) :- je_clen(H, L2), prunik2(L1, L2, XTAIL).
prunik2([_|L1], L2, X) :- prunik2(L1, L2, X), !.

je_podmnozina(L1, L2, L3) :-
    prunik2(L1, L2, PRUNIK),
    prunik2(PRUNIK, L3, L3), !.
```

## % Součet prvních dvou prvků a dvou posledních prvků lin. Seznamu (seznam obsahuje vždy alespoň 2 prvky)

```
soucetprvni2([H1,H2|_], X) :- X is H1 + H2.

soucetposledni2([T1,T2], X) :- X is T1 + T2.
soucetposledni2([_|T], X) :- soucetposledni2(T,X).

soucetp2p2(S,X) :- soucetprvni2(S,X1), soucetposledni2(S,X2), X is X1 + X2.
```

## % Zjistit, zda je lineární seznam množinou

```
je_mnozina([]).
je_mnozina([H|T]) :- \+ je_clen(H, T), je_mnozina(T).
```

## % Součet dvou největších prvků v číselném seznamu (bez vnořených)

```
dva_nej([X1,X2], P) :- P is X1 + X2, !.
dva_nej([H1,H2,H3|T], P) :- H1 <= H2, H1 <= H3, dva_nej([H2,H3|T], P).
dva_nej([H1,H2,H3|T], P) :- H2 <= H1, H2 <= H3, dva_nej([H1,H3|T], P).
dva_nej([H1,H2,H3|T], P) :- H3 <= H1, H3 <= H2, dva_nej([H1,H2|T], P).
```

## % Součet prvního kladného a posledního záporného čísla v seznamu

```
prvni_kladne([H|_], H) :- H > 0.  
prvni_kladne([_|T], X) :- prvni_kladne(T, X), !.  
  
vrat_posledni([X], X) :- !.  
vrat_posledni([_|T], X) :- vrat_posledni(T, X).  
  
pouze_zaporne([], []).  
pouze_zaporne([H|T], L) :- H >= 0, pouze_zaporne(T, L), !.  
pouze_zaporne([H|T], [H|L]) :- H < 0, pouze_zaporne(T, L), !.  
  
posledni_zaporne(L, X) :- pouze_zaporne(L, L1), vrat_posledni(L1, X).  
  
soucet_prvni_kladne_posledni_zaporne(L, X) :-  
    prvni_kladne(L, K),  
    posledni_zaporne(L, Z),  
    X is K + Z.
```

## % Zjištění **maximálního** čísla v seznamu (umí pouze lineární seznamy)

```
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).
```

## % Zjištění **minimálního** čísla v seznamu (umí pouze lineární seznamy)

```
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).
```

## % Zjištění indexu největšího čísla v seznamu

```
indexof([H|_], H, 0).  
indexof([_|T], X, I) :- indexof(T, X, J), I is J + 1.  
  
index_max_cisla([H|T], I) :- maximum([H|T], M), indexof([H|T], M, I).
```

## % Sjednocení množin I a J a následný průnik s L

```
vlozZ(P, S, S) :- je_clen(P, S), !.  
vlozZ(P, S, [P|S]).
```

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

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

## % Odstranit poslední prvek v seznamu

```
remove_last([], []) :- !, fail.  
remove_last([_], []) :- !.  
remove_last([X | T], [X | T2]) :- remove_last(T, T2).
```

## % Součet minimálního a maximálního čísla v seznamu

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

## % Počet atomických prvků v seznamu

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

## % Sjednocení 3 množin

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

## % Faktorial levého čísla

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

## % Monotónnost seznamu (jen roste nebo jen klesá)

```
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]).
```

## % Fibonacci

```
fib(0, 1) :- !.  
fib(1, 1) :- !.  
fib(N, Result) :-  
    N1 is N - 1,  
    N2 is N - 2,  
    fib(N1, Result1),  
    fib(N2, Result2),  
    Result is Result1 + Result2.
```