

**Deklaratív programozás**  
**Záróvizsga-feladat, 2004. június 2.**

Tekintse át a deklaratív nyelvek listakezelési lehetőségeit!

1. Ismertesse a Prolog és SML nyelvek lista-jelöléseit, beleértve a szintaktikus „édesítőszereket”.
2. Hasonlítsa össze a két nyelvnek azokat a módszereit, amelyekkel a listákat minimális memóriaigénnyel lehet feloldozni.
3. Térjen ki a Prolog logikaiváltozó-fogalmának listaépítéskor való használatára (listák építése előlről és hátulról).

Mondanivalóját illusztrálja egy-egy SML, ill. Prolog programon az alábbi nem-hatékony szetrak függvény, ill. eljárás szetdob nevű változatának megírásával, amelynek (1) minimális a memóriaigénye, (2) minimális a futásigénye!

SML példa:

```
(* szetrak L = SL, ahol az SL n-elemű lista az L n-elemű lista
   ,,szétrakásával'' áll elő, azaz
   SL[i] = L[2i-1],   i = 1, 2, ..., (n+1) div 2, és
   SL[n+1-i] = L[2i], i = 1, 2, ..., n div 2.
*)
fun szetrak (X::Y::L) = X :: szetrak L @ [Y]
  | szetrak L = L;

szetrak [1,2,3,4,5] = [1,3,5,4,2];
szetrak [1,2,3,4,5,6] = [1,3,5,6,4,2];
```

Prolog példa:

```
:-use_module(library(lists)).

% szetrak(L, SL): Az SL n-elemű lista az L n-elemű lista ,,szétrakásával''
% áll elő, azaz SL[i] = L[2i-1],   i = 1,2,...,(n+1)//2, és
%           SL[n+1-i] = L[2i], i = 1,2,...,n//2.
szetrak([X,Y|L], SL) :- !,
    szetrak(L, SL0),
    append([X|SL0], [Y], SL).
szetrak(L, L).

| ?- szetrak([1,2,3,4,5], L).
L = [1,3,5,4,2] ? ;
no
```