

**Deklaratív programozás**  
**Záróvizsga-feladat, 2001. június 11.**

Tömören foglalja össze az SML- és a Prolog-nyelv kifejezésértékelési, ill. végrehajtási mechanizmusának (vezérlési szerkezetének) alapelveit!

Példaként ismertesse az alábbi `symmElems` nevű SML-függvény és Prolog-eljárás működését. Hívja fel a figyelmet azokra a nyelvi elemekre, amelyek segítségével SML-ben, ill. Prologban rekurzió-mentes programokat lehet készíteni! (Dolgozatában a példaprogramok egyes elemeire a megadott sorszámokkal hivatkozhat.)

```
1 (* ListPair.zip : 'a list * 'b list -> ('a * 'b) list
2   ListPair.zip (xs, ys) = az xs és ys azonos indexű elemeiből képzett
                           pároknak az eredeti sorrendet megőrző listája
3   List.map : ('a -> 'b) -> 'a list -> 'b list
4   List.map f xs = az xs elemeiből az f transzformációval előállított
                   elemeknek az eredeti sorrendet megőrző listája
5   List.filter : ('a -> bool) -> 'a list -> 'a list
6   List.filter p xs = az xs elemei közül a p predikátumot kielégítő
                       elemeknek az eredeti sorrendet megőrző listája *)

7 load "ListPair";
8 fun symmElems xs =
9   List.map #1 (List.filter op= (ListPair.zip(xs, rev xs)));

10 :- use_module(library(lists), [append/3, same_length/2]).
11 % same_length(L1, L2): Az L1 és L2 listák elemszáma azonos.
12 % append(L1, L2, L3): Az L3 lista az L1 és az L2 egymás után fűzésével áll elő.

13 symmElems(Xs, Zs) :-
14   findall(Z, symmElem(Xs, Z), Zs).

15 symmElem(Xs, Z) :-
16   append(Pre, [Z|_], Xs), append(_, [Z|Post], Xs),
17   same_length(Pre, Post).
```

Vesse össze a fenti rekurzió-mentes megvalósítások végrehajtását az alábbi rekurzív programok végrehajtásával. Mutasson rá a két nyelv közötti különbségekre a rekurzív megoldásokban!

```
18 fun symmElems xs =
19   let fun symmEs (x::xs) (y::ys) zs =
20         symmEs xs ys (if x = y then x::zs else zs)
21       | symmEs _ _ zs = zs
22   in
23     symmEs xs (rev xs) []
24   end;

25 :- use_module(library(lists), [append/3, reverse/2]).
26 % reverse(L, R): Az L lista megfordítása R.

27 symmElems(Xs, Zs) :-
28   reverse(Xs, Rs), symmEs(Xs, Rs, Zs).

29 symmEs([], [], []).
30 symmEs([X|Xs], [Y|Ys], Zs) :-
31   ( X = Y -> Zs = [X|Zs1]
32   ;   Zs = Zs1
33   ),
34   symmEs(Xs, Ys, Zs1).
```