



Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldl	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	chr	: int -> char
map	: ('a -> 'b) -> 'a list -> 'b list	Char.isUpper	: char -> bool
op@	: 'a list * 'a list -> 'a list	size	: string -> int
List.revAppend	: 'a list * 'a list -> 'a list	op::	: 'a * 'a list -> 'a list
op^	: string * string -> string		

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **statikus szemantikai hiba** van. Melyek ezek? (7 pont)

- (a) [op>=(2, "2" ), false = false, "c" <> #"b"]
- (b) (chr 127, 8+8 ^ 4\*4, ~1) = (#"B", "", true)
- (c) foldl (op mod) ~1.0 [(0,1),(4,3),(1,2)]

6. Mi a t **értéke** és **típusa** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val (\_::\_:t::\_) = List.revAppend(explode "MaTeX", explode "" @ [#"+"])
- (b) val (\_::\_:t) = List.filter Char.isUpper (explode "LaTeX-MaTeX")
- (c) val t = map (fn (a,b) => a-b) [(4+0,2\*2), (1,2), (size "ab", size "bc")]

7. A redl függvényt így definiáljuk: (8 pont)

```
(* redl : ('a * 'b * int -> 'b) -> 'a -> 'b -> int -> 'b *)
fun redl f c e 0 = e | redl f c e n = redl f c (f(c, e, n)) (n-1)
```

Mi az x **értéke** az alábbi, egymástól független deklarációk kiértékelése után?

- (a) val x = redl (fn (c, i, k) => i \* c + 2 + k) 2 0 0
- (b) val x = redl (fn (c, \_, k) => c \* 2 + k) 2 1 1
- (c) val x = map (redl (fn (c, i, \_) => c + i \* i) 5 0) [0,1]
- (d) val x = List.filter (fn i => i mod 5 <> 0)
   
 (map (redl (fn (c, \_, k) => c + k) 5 0) [0,1])

8. Leszállónak nevezzük az olyan a,b,c számhármast, amelyre a-b>b-c, leszállási meredekségének pedig az (a-b)-(b-c) különbséget.

Írjon olyan függvényt lemer néven, amelynek egy egészlista az argumentuma, az eredménye pedig az argumentumlista egymást követő a, b és c elemeiből álló leszálló számhármastok leszállási meredekségének az eredeti sorrendet megőrző listája.

Ha tud, alkalmazzon jobbrekurziót. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet. (8 pont)

```
(* lemer: int list -> int list
   lemer ns = az ns lista egymást követő elemeiből álló leszálló számhármastok
               leszállási meredekségének az eredeti sorrendet megőrző listája
*)
```

Példák: lemer [] = [];  
 lemer [6,4,3] = [1];  
 lemer [6,4,2] = [];  
 lemer [9,5,2,1,8,4,3] = [1, 2, 8, 3];



Deklaratív programozás, pótzárthelyi, 2005. november 24. 18.15–19.45

Munkaidő: 90 perc, összpontszám: 60

Standard ML, „B” csoport (30 pont)

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	ord	: char -> int
map	: ('a -> 'b) -> 'a list -> 'b list	Char.isLower	: char -> bool
op@	: 'a list * 'a list -> 'a list	op::	: 'a * 'a list -> 'a list
List.revAppend	: 'a list * 'a list -> 'a list	chr	: int -> char

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **statikus szemantikai hiba** van. Melyek ezek? (7 pont)

- (a) [1.0 / 2, 171717, ord #"a" mod 3.0]
- (b) (chr 64, 4 > 4, ~12) = ("b", 2+2 = 4 div 1, ~(7-7.7))
- (c) foldr (op/) ~10 [[1.4, 4.4], [7.2,3.4], [2.1,1.2]]

6. Mi a t **értéke** és **típusa** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val (\_::t::\_::\_) = List.revAppend(explode " @ [#"-", explode "MaTeX")
- (b) val (\_::\_::\_::t) = List.filter Char.isLower (explode "MaTeX-LaTeX")
- (c) val t = map (fn (a,b) => b-a) [(2,1), (2+2,4\*2), (ord #"B",ord #"A")]

7. A redr függvényt így definiáljuk: (8 pont)

```
(* redr : ('a * 'b * int -> 'b) -> 'a -> 'b -> int -> 'b *)
fun redr f c e 0 = e | redr f c e n = f(c, redr f c e (n-1), n)
```

Mi az x **értéke** az alábbi, egymástól független deklarációk kiértékelése után?

- (a) val x = redr (fn (\_, i, k) => i + 3 + k) 2 0 0
- (b) val x = redr (fn (c, \_, k) => c \* 3 + k) 2 1 1
- (c) val x = map (redr (fn (c, i, \_) => c + i \* i) 5 0) [1,0]
- (d) val x = List.filter (fn i => i mod 3 <> 0) (map (redr (fn (c, \_, k) => c + k) 3 0) [1,0])

8. Elszállónak nevezzük az olyan a,b,c számhármast, amelyre a-b<b-c, elszállási meredekségének pedig a (b-c)-(a-b) különbséget.

Írjon olyan függvényt elmer néven, amelynek egy egészlista az argumentuma, az eredménye pedig az argumentumlista egymást követő a, b és c elemeiből álló elszálló számhármastok elszállási meredekségének az eredeti sorrendet megőrző listája.

Ha tud, alkalmazzon jobbrekurziót. Segédfüggvényt definiálhat, ha ír hozzá fejkomentet. (8 pont)

```
(* elmer: int list -> int list
   elmer ns = az ns lista egymást követő elemeiből álló elszálló számhármastok
   elszállási meredekségének az eredeti sorrendet megőrző listája
*)
```

Példák: elmer [] = [];  
elmer [5,4,2] = [1];  
elmer [5,4,3] = [];  
elmer [9,6,2,9,3,7,8] = [1, 13, 3];