

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	real	: int -> real
map	: ('a -> 'b) -> 'a list -> 'b list	List.last	: 'a list -> 'a
op@	: 'a list * 'a list -> 'a list	ord	: char -> int
op::	: 'a * 'a list -> 'a list	rev	: 'a list -> 'a list
op^	: string * string -> string	tl	: 'a list -> 'a list

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

- (a) [(1 = real 2), op^("a", "bc") <> #"a", [1*4] = [4*1]]
- (b) (ord #"B", 2*4 = 4-2, ~4.0) = (#"B", true, ~3+1)
- (c) foldl (fn (a,b) => explode a ^ b) [" "] ["one", "zwei", "drie"]

6. Mi az s **értéke** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val (_::s::_::_) = explode "ho" @ rev(tl(explode "tel"))
- (b) val (r::s) = List.filter (fn (a,b) => (a>=b)) [(4+0,2*2), (2,2-1), (2-1,2)]
- (c) val s = map List.last [explode("3c4d"), ["Q"], rev(explode("let")) @ ["z"]]

7. Tekintsük a következő függvénydefiníciót! (8 pont)

```
(* g : int list * int -> int list      f : int list * int list -> int list *)
fun g (n, xs) =
  let fun f (a::b::bs, zs) =
        if b-a=n then f(b::bs, 10*a+b::zs) else f(b::bs, zs)
      | f (_, zs) = rev zs
  in f(xs, []) end;
```

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

- (a) val x = g(1, [1,2,3,4,5,6])
- (b) val x = g(2, [1,3,4,6,8,9])
- (c) val x = g(~3, [1,~2,~5,~8,~5,~2,1])
- (d) val x = g(0, [1,~2,~2,4,4,4,5,6,7])

Fejezze be a fejkommentet!

- (e) (* g (0, xs) = az xs azon elemeinek ...szereiseiből álló lista, amelyek ... *)

8. Tekintsük az alábbi adattípus-deklarációt:

```
datatype 'a H = A of 'a | B of 'a H list
```

Lyukashármasnak nevezzük az olyan (a, b, c) hármasot, amelyre $a * c > 0$ és $b = 0$. Írjon olyan függvényt lyukasharmasok néven, amely egy (int*int*int) H típusú adatstruktúrában található lyukasharmasok eredeti sorrendet megőrző listáját adja eredményül. Törekedjék hatékony megoldásra és magasabbrendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet. (8 pont)

```
(* lyukasharmasok : (int * int * int) H -> (int * int * int) list
   lyukasharmasok t = a t-beli lyukasharmasok eredeti sorrendet megőrző listája *)
```

```
Példák: lyukasharmasok(B[B[],A(6,4,3),A(~6,0,3)]) = [];
         lyukasharmasok(A(6,0,3)) = [(6,0,3)];
         lyukasharmasok(A(~6,0,~3)) = [(~6,0,~3)];
         lyukasharmasok(B[B[],B[],A(6,0,3)]) = [(6,0,3)];
         lyukasharmasok(B[B[A(1,0,4),A(6,0,9),B[A(1,3,2),B[A(7,0,0)]]],
                           B[],A(~3,0,~9)]) = [(1,0,4), (6,0,9), (~3,0,~9)];
```