

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	implode	: char list -> string
map	: ('a -> 'b) -> 'a list -> 'b list	length	: 'a list -> int
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) [(1.3 = 2), op^("a", #"b") = "ab", [] = [4*1]]
- (b) (ord "B", 2-4 = 4-2, ~3.4) = (65, true, ~3-4)
- (c) foldl (fn (a,b) => explode a @ b) #" " ["egy", "ketto", #"3"]

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

- (a) val (_::_:t::_) = explode "ab" @ tl(rev(explode "cde"))
- (b) val (s::t) = List.filter (fn (a,b) => (a<=b)) [(4+0,2*2), (2,2-1), (2-1,2)]
- (c) val t = map length [explode "1a2b3c4d", ["#Q"], [], explode ""]

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::c::cs, zs) =
                if a+b>n then f(b::c::cs, 10*n+c::zs) else f(b::c::cs, zs)
                | f (_, zs) = rev zs
            in f(xs, []) end;
```

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

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

Fejezze be a fejkommentet!

- (e) (* g 0 xs = az xs összes olyan eleméből álló lista, amelyek ... *)

8. Tekintsük az alábbi adattípus-deklarációt:

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

Farneheznek nevezzük az olyan (a, b, c, d) négyest, amelyre $a + b + c \leq d$. Írjon olyan függvényt farnehezek néven, amely egy $(int * int * int * int)$ H típusú adatstruktúrában található farnehezek eredeti sorrendet megőrző listáját adja eredményül. Törekedjék hatékony megoldásra és magasabbrendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet. (8 pont)

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

```
Példák: farnehezek(A(6,4,~3,3)) = [];
         farnehezek(A(4,3,0,8)) = [(4,3,0,8)];
         farnehezek(A(4,3,~7,0)) = [(4,3,~7,0)];
         farnehezek(B[]) = [];
         farnehezek(B[B[],B[],A(6,4,~2,9)]) = [(6,4,~2,9)];
         farnehezek(B[B[A(1,2,4,8),A(6,3,0,9)],B[A(0,1,3,2),B[A(8,~7,0,0)]]],
                   B[],A(4,3,1,9)]) = [(1,2,4,8), (6,3,0,9), (4,3,1,9)];
```

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	implode	: char list -> string
map	: ('a -> 'b) -> 'a list -> 'b list	length	: 'a list -> int
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) [1.0 > 2, op=(#"a", "b") = false, [7-3] = []]
- (b) (93, 0, ~12) = (ord "b", 2+1 = 4-1, ~7-7)
- (c) foldr (fn (a,b) => implode a ^ b) #"i" [[#"a", #"z"], "abc", [#"3"]]

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

- (a) val (_::_:t::_) = rev(tl(explode "abc")) @ explode "de"
- (b) val (s::t) = List.filter (fn (a,b) => (a>=b)) [(4+0,2*2), (2,2-1), (2-1,2)]
- (c) val t = map length [explode "", explode "d4c3b2a1", [], [#"R", #"S"]]

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::c::cs, zs) =
                if a+b<n then f(b::c::cs, 10*n+c::zs) else f(b::c::cs, zs)
                | f (_, zs) = rev zs
            in f(xs, []) end;
```

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

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

Fejezze be a fejkommentet!

- (e) (* g 0 xs = az xs összes olyan eleméből álló lista, amelyek ... *)

8. Tekintsük az alábbi adattípus-deklarációt:

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

Fejenagynak nevezzük az olyan (x, y, z) hármast, amelyre $x > y + z$. Írjon olyan függvényt fejenagyok néven, amely egy (int*int*int) G típusú adatstruktúrában található fejenagyok eredeti sorrendet megőrző listáját adja eredményül. Törekedjék hatékony megoldásra és magasabbrendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.

(8 pont)

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

```
Példák: fejenagyok(A(6,4,3)) = [];
         fejenagyok(A(4,3,0)) = [(4,3,0)];
         fejenagyok(A(4,3,1)) = [];
         fejenagyok(B[]) = [];
         fejenagyok(B[B[],B[],A(6,4,1)]) = [(6,4,1)];
         fejenagyok(B[B[A(8,2,4),A(6,9,4)],B[A(7,5,1),B[A(8,7,1)],A(~1,7,~9)],
                     B[],A(4,3,0)]) = [(8,2,4), (7,5,1), (~1,7,~9), (4,3,0)];
```