



Deklaratív programozás, nagyzárthelyi, 2008. április 10.

Munkaidő: 90 perc, összpontszám: 60

Standard ML, „A” 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 alapműveletek és a relációk kivételével):

<code>explode</code>	: <code>string -&gt; char list</code>	<code>round</code>	: <code>real -&gt; int</code>
<code>List.filter</code>	: <code>('a -&gt; bool) -&gt; 'a list -&gt; 'a list</code>	<code>Char.isLower</code>	: <code>char -&gt; bool</code>
<code>map</code>	: <code>('a -&gt; 'b) -&gt; 'a list -&gt; 'b list</code>	<code>ord</code>	: <code>char -&gt; int</code>
<code>op::</code>	: <code>'a * 'a list -&gt; 'a list</code>	<code>tl</code>	: <code>'a list -&gt; 'a list</code>

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **hiba** van (két helyen kell javítani). Mik ezek? (7 pont)

- (a) `[7 > 3.3, #"a" < #"b", round 2.1 - 1]`
- (b) `(ord "C", 7+5, 13-10, 42) = (97, 12, 23)`
- (c) `map (fn z => (1.0-z)) (3.1, 5.3, z)`

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

- (a) `val (_::k) = tl(tl(explode "Alice"))`
- (b) `val k = List.filter (fn x => x <= 3) [1,7,3,5,3]`
- (c) `val (_::_::k::_) = map Char.isLower (explode "AlIcE")`

7. Nézzük a következő függvények definícióját! (7 pont)

```
fun hipp (x::xs, y::ys) = (y, x) :: hipp(ys, xs) | hipp _ = []
fun h zs = map (fn (b,a) => b-a) (hipp(tl zs, zs))
```

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

- (a1) `val x = h [1,2,3,4,5]`
- (a2) `val x = h [~1]`
- (a3) `val x = h [~1,1]`
- (a4) `val x = h []`

(b) Mutassa be a `hipp([2,3,4], [1,2,3,4])` függvényalkalmazás **egyszerűsítési lépéseit**, mohó kiértékelést feltételezve!

8. Adott egy `int * int` típusú párokból álló lista. Az ilyen párok halmazán értelmezett lexikografikus rendezés szerint egy  $(a, b)$  párt akkor és csak akkor mondunk nagyobbak egy  $(c, d)$  párnál, ha vagy  $a > c$ , vagy  $a = c$  és  $b > d$ . Írjon olyan **SML függvényt** `pnagyobb` néven, amely egy `(int * int) list` típusú listát kap bemenetként, és az eredménye olyan párok listája, amelyek a fenti lexikografikus rendezés szerint nagyobbak az őket megelőzőnél! Segédfüggvényt definiálhat, ha ad hozzá fejkommentet. (9 pont)

```
(* pnagyobb : (int * int) list -> (int * int) list
   pnagyobb ps = azoknak a ps-beli pároknak a listája, amelyek a fenti
   lexikografikus rendezés szerint nagyobbak az őket megelőzőnél
*)
```

Példák: `pnagyobb [(1,1)] = []`  
`pnagyobb [(1,3), (1,4)] = [(1,4)]`  
`pnagyobb [(1,3), (1,3), (1,2)] = []`  
`pnagyobb [(0,3), (0,4), (1,2)] = [(0,4), (1,2)]`  
`pnagyobb [(0,3), (1,4), (1,5), (0,6), (2,1), (1,2)] = [(1,4), (1,5), (2,1)]`



Deklaratív programozás, nagyzárthelyi, 2008. április 10.

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 alapműveletek és a relációk kivételével):

<code>explode</code>	: <code>string -&gt; char list</code>	<code>round</code>	: <code>real -&gt; int</code>
<code>List.filter</code>	: <code>('a -&gt; bool) -&gt; 'a list -&gt; 'a list</code>	<code>Char.isUpper</code>	: <code>char -&gt; bool</code>
<code>map</code>	: <code>('a -&gt; 'b) -&gt; 'a list -&gt; 'b list</code>	<code>chr</code>	: <code>int -&gt; char</code>
<code>op::</code>	: <code>'a * 'a list -&gt; 'a list</code>	<code>tl</code>	: <code>'a list -&gt; 'a list</code>

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **hiba** van (két helyen kell javítani). Mik ezek? (7 pont)

- (a) `[3*7>7*3, round 3, 2*1]`
- (b) `(chr #"x", 5, 7) = (65, 3+2, 1, 9*9)`
- (c) `List.filter [3, 5, z] (fn z => 4 > z)`

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

- (a) `val (_::_:::_:::_:_) = tl(explode "Haskell")`
- (b) `val k = List.filter (fn x => x >= 4) [1,8,2,4,8]`
- (c) `val (_::k::_) = map Char.isUpper (explode "HaSkElL")`

7. Nézzük a következő függvények definícióját! (7 pont)

```
fun hopp (r::rs, s::ss) = (r, s) :: hopp(ss, rs) | hopp _ = []  
fun h zs = map (fn (a,b) => b-a) (hopp(tl zs, zs))
```

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

- (a1) `val x = h [1,2,3,4,5]`
- (a2) `val x = h [~1]`
- (a3) `val x = h [~1,1]`
- (a4) `val x = h []`

(b) Mutassa be a `hopp([2,3,4], [1,2,3,4])` függvényalkalmazás **egyszerűsítési lépéseit**, mohó kiértékelést feltételezve!

8. Adott egy `int * int` típusú párokból álló lista. Az ilyen párok halmazán értelmezett lexikografikus rendezés szerint egy  $(a, b)$  párt akkor és csak akkor mondunk kisebbnek egy  $(c, d)$  párnál, ha vagy  $a < c$ , vagy  $a = c$  és  $b < d$ . Írjon olyan **SML függvényt** `pkisebb` néven, amely egy `(int * int) list` típusú listát kap bemenetként, és az eredménye olyan párok listája, amelyek a fenti lexikografikus rendezés szerint kisebbek az őket követőnél! Segédfüggvényt definiálhat, ha ad hozzá fejkommentet. (9 pont)

```
(* pkisebb : (int * int) list -> (int * int) list  
   pkisebb ps = azoknak a ps-beli pároknak a listája, amelyek a fenti  
               lexikografikus rendezés szerint kisebbek az őket követőnél  
)
```

Példák: `pkisebb [(1,1)] = []`  
`pkisebb [(1,3), (1,4)] = [(1,3)]`  
`pkisebb [(1,3), (1,3), (1,2)] = []`  
`pkisebb [(0,3), (0,4), (1,2)] = [(0,3), (0,4)]`  
`pkisebb [(0,3), (1,4), (1,5), (0,6), (2,1), (1,2)] = [(0,3), (1,4), (0,6)]`