



Deklaratív programozás, nagyvárhelyi, 2006. november 16.

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.isUpper</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>real</code>	: <code>int -&gt; real</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) `[7.0 > 3, #"a" > #"b", 1-2]`
- (b) `(ord "a", 5+7, 10+13) = (97, 12, 23, 42)`
- (c) `map (fn x => (x-1.0) ) (3.1, x, 5.3)`

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

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

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

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

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

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

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

8. Adott egy `int * int` párokból álló lista. Két szomszédos pár második tagjainak szorzatát értékszorzatnak hívjuk. Írjon olyan **SML függvényt** eszorzat néven, amely egy `(int * int) list` listát kap bemenetként és eredményül olyan listát ad, melyben azon értékszorzatok szerepelnek, amelyek olyan egymást követő párokhoz tartoznak, ahol a párok első tagjai megegyeznek! Segédfüggvényt definiálhat, ha ad hozzá fejkommentet. (9 pont)

```
(* eszorzat : (int * int) list -> int list
eszorzat pl = azon pl-beli értékszorzatok listaja,
amelyek olyan egymást követő párokhoz tartoznak, ahol a párok
első tagjai megegyeznek *)
```

Példák: `eszorzat [(1,1)] = []`  
`eszorzat [(1,3), (1,2)] = [6]`  
`eszorzat [(1,3), (2,4), (1,2)] = []`  
`eszorzat [(0,3), (0,4), (0,2)] = [12,8]`  
`eszorzat [(0,3), (1,4), (1,5), (1,6), (2,4), (2,2)] = [20,30,8]`



Deklaratív programozás, nagyzárthelyi, 2006. november 16.

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.isLower</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>real</code>	: <code>int -&gt; real</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 7, 1*2]`
- (b) `(chr #"A", 5, 6, 7) = (65, 3+2, 6)`
- (c) `List.filter [3, x, 5] (fn x => x < 4)`

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

- (a) `val (_::_::_::_::k) = tl(explode "Prolog")`
- (b) `val k = map (fn x => real x / 2.0) [8,2,6]`
- (c) `val (_::k::_) = List.filter Char.isLower (explode "Standard ML")`

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

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

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

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

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

8. Adott egy `int * int` párokból álló lista. Két szomszédos pár második tagjainak összegét értékösszegnek hívjuk. Írjon olyan **SML függvényt** `eosszeg` néven, amely egy `(int * int) list` listát kap bemenetként és eredményül olyan listát ad, melyben azon értékösszegek szerepelnek, amelyek olyan egymást követő párokhoz tartoznak, ahol a párok első tagjai különbözőek! Segédfüggvényt definiálhat, ha ad hozzá fejkomentet. (9 pont)

```
(* eosszeg : (int * int) list -> int list
eosszeg pl = azon pl-beli értékösszegek listája,
amelyek olyan egymást követő párokhoz tartoznak, ahol a párok
első tagjai különbözőek *)
```

Példák: `eosszeg [(1,4)] = []`  
`eosszeg [(1,4), (2,5)] = [9]`  
`eosszeg [(1,4), (1,5), (2,5)] = [10]`  
`eosszeg [(1,4), (2,2), (3,7)] = [6,9]`  
`eosszeg [(0,2), (1,4), (2,2), (2,4), (3,7)] = [6,6,11]`