```
                  Deklaratív programozás nagyzárthelyi
                      Budapest, 2006. november 16.
                      ============================
              SML-megoldások, V1.1, dp06a-zh1-bme-mlmegol-l2.txt

--------------------
   A csoport
--------------------

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

   (a) [7.0 > 3, #"a" > #"b", 1-2]
       - a lista elemei különböző típusúak (bool,bool,int)
       - real és int nem hasonlítható össze

   (b) (ord "a", 5+7, 10+13) = (97, 12, 23, 42)
       - hármas és négyes nem hasonlítható össze
       - ord argumentuma nem char

   (c) map (fn x => (x-1.0) ) (3.1, x, 5.3)
       - map második argumentuma nem lista
       - map második argumentumában az x "unbound variable or constructor"

Pontozás (összesen max. 7 pont):
   5.(a) - 5.(c): helyes válasz 2-2-3 pont.
   Minden hiba megtalálása 1 pontot ér, kivéve az 5.(c)-t,
   ahol az első hiba felismerésére 2 pontot adunk.

6. Mi a k értéke az alábbi, egymástól független deklarációk kiértékelése után?

   (a) val (_::k) = tl(tl(explode "SML"))
          k = []

   (b) val k = List.filter (fn x => x > 3) [1,7,3,5]
          k = [7,5]

   (c) val (_::k::_) = map Char.isUpper (explode "ProLog")
          k = false

Pontozás (összesen max. 7 pont):
   6.a - 6.c: helyes válasz 2-2-3 pont.  Hibákért 1-1 pont levonás.

7. Tekintsük a következő függvénydefiníciókat!

   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))

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

     (a1) val x = f [1,2,3,4,5]
             x = [1,~1,1,~1]

     (a2) val x = f [~1]
             x = []

     (a3) val x = f [~1,1]
             x = [2]

     (a4) val x = f []
             -> uncaught exception Empty
```

```
     (b) kutya([1,2,3,4], [2,3,4]) egyszerűsítési lépései:

             kutya([1,2,3,4],[2,3,4]) ->
             (2,3)::kutya([3,4],[2,3,4]) ->
             (2,1)::(2,3)::kutya([3,4],[4]) ->
             (2,1)::(2,3)::(4,3)::kutya([],[4]) ->
             (2,1)::(2,3)::(4,3)::[] ->
             (2,1)::(2,3)::[(4,3)] ->
             (2,1)::[(2,3),(4,3)] ->
             [(2,1),(2,3),(4,3)]

Pontozás (összesen max. 7 pont):
   7.(a1)-7.(a3)-ra helyes válaszért 1-1,
   7.(a4)-re 2 pont, ha Empty a válasz, vagy ha jelöli valahogy, hogy nincs
     visszatérési érték, hanem kivételt jelez a függvény
   7.=b) 2 pont, nem vonunk le pontot, ha a :: operátorokat egy lépésben írja át
     a [...] alakra, vagy ha meghagyja


8. (* 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
   *)

   Nem jobbrekurzív:

     fun eszorzat ( (k1,v1)::(k2,v2)::ls ) =
           if k1 = k2
           then (v1*v2)::eszorzat( (k2,v2)::ls )
           else eszorzat( (k2,v2)::ls )
       | eszorzat _ = []

   Jobbrekurzív:

     fun eszorzat ls =
        let
           fun loop ( (k1,v1)::(k2,v2)::ls, zs) =
              if k1 = k2
                 then loop( (k2,v2)::ls,(v1*v2)::zs )
                 else loop( (k2,v2)::ls,zs )
             | loop(_,zs) = rev zs
        in
           loop( l,[])
        end

Pontozás (összesen max. 9 pont):
   Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont
   levonás.  Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy a
   végtelen rekurzió (-3 pont).  Segédfüggvényben a fejkomment hiánya: -2
   pont, de nem követelmény a fejkommentben a függvény típusának
   specifikálása.  Ha nagyon rossz a program hatékonysága, -2 pont a levonás.
```

Deklaratív programozás nagyzárthelyi
Budapest, 2006. november 16.
==========

SML-megoldások, V1.0, dp06s-zh1-bme-mlmegol-l2.txt

B csoport

---

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

(a) [3*7<>7*3, round 7, 1*2]
- a lista elemei nem azonos típusúak (bool,int,bool)
- round: real -> int, az argumentuma: int

(b) (chr #"A", 5, 6, 7) = (65, 3+2, 6)
- négyes nem hasonlítható össze hármassal
- chr: int -> char, az argumentuma: char

(c) List.filter [3, x, 5] (fn x => x < 4)
- a filter két argumentuma fel van cserélve
- a filter lista-argumentumában az x unbound variable or constructor

Pontozás (összesen max. 7 pont):
5.a – 5.c: helyes válasz 2-2-3 pont.
Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol az első hiba felismerésére 2 pontot adunk.

6. Mi a k értéke az alábbi, egymástól független deklarációk kiértékelése után?

(a) val (_::_::_::_::k) = tl(explode "Prolog")
k = [#"g"]

(b) val k = map (fn x => real x / 2.0) [8,2,6]
k = [4.0,1.0,3.0]

(c) val (_::k::_) = List.filter Char.isLower (explode "Standard ML")
k = #"a"

Pontozás (összesen max. 7 pont):
6.a – 6.c: helyes válasz 2-2-3 pont. Hibákért 1-1 pont levonás.

7. Tekintsük a következő függvénydefiníciókat!

```
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))
```

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

(a1) val x = f [1,2,3,4,5]
x = [1,~1,1,~1]

(a2) val x = f [~1]
x = []

(a3) val x = f [~1,1]
x = [2]

(a4) val x = f []
-> uncaught exception Empty

(b) macska([1,2,3,4], [2,3,4]) egyszerűsítési lépései:

```
macska([1,2,3,4], [2,3,4]) ->
(1,2)::macska([3,4],[2,3,4]) ->
(1,2)::(3,2)::macska([3,4],[4]) ->
(1,2)::(3,2)::(3,4)::macska([],[4]) ->
(1,2)::(3,2)::(3,4)::[] ->
(1,2)::(3,2)::[(3,4)] ->
(1,2)::[(3,2),(3,4)] ->
[(1,2),(3,2),(3,4)]
```

Pontozás (összesen max. 7 pont):
7.(a1)-7.(a3)-ra helyes válaszért 1-1,
7.(a4)-re 2 pont, ha Empty a válasz, vagy ha jelöli valahogy, hogy nincs visszatérési érték, hanem kivételt jelez a függvény
7.(b) 2 pont, nem vonunk le pontot, ha a :: operátorokat egy lépésben írja át a [...] alakra, vagy ha meghagyja

8.
```
(* 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
 *)
```

Nem-jobbrekurzív:

```
fun eosszeg ( (k1,v1)::(k2,v2)::ls ) =
      if k1 <> k2
      then (v1+v2)::eosszeg( (k2,v2)::ls )
      else eosszeg( (k2,v2)::ls )
  | eosszeg _ = []
```

Jobbrekurzív:

```
fun eosszeg l =
   let
      fun loop ( (k1,v1)::(k2,v2)::ls, zs) =
         if k1 <> k2
            then loop( (k2,v2)::ls,(v1+v2)::zs )
            else loop( (k2,v2)::ls,zs )
        | loop(_,zs) = rev zs
   in
      loop( l,[])
   end
```

Pontozás (összesen max. 9 pont):
Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2 pont, de nem követelmény a fejkommentben a függvény típusának specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a levonás.