Declarative Programming, Midterm exam, 22nd April 2005, 15.15–17.45
Total time available: 90 minutes, total score: 60
Standard ML, Group „A" (30 point)

When the task is to write a function, all standard functions of SML and the functions defined in the lectures can be used. The types of the standard functions which appear in the tasks are the following:

`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`	`Char.isDigit`	`: char -> bool`
`op@`	`: 'a list * 'a list -> 'a list`	`size`	`: string -> int`
`op::`	`: 'a * 'a list -> 'a list`	`tl`	`: 'a list -> 'a list`
`op^`	`: string * string -> string`	`chr`	`: int -> char`

5. There are exactly two semantic errors in each of the following (independent) syntactically correct SML expressions. Which are these errors? (7 points)

   (a) `[op>(1.3, 2), "a" = #"b", true]`

   (b) `(chr 65, 2*4 = 4+4, ~12) = ("B", 8, ~5-7)`

   (c) `foldl op^ [(1,0),(3,4),(2,1)] ~10`

6. What is the value of `t` after evaluating the following (independent) value-definitions? (7 points)

   (a) `val (_::t::_::_) = explode "X" @ tl(explode "mas")`

   (b) `val (_::_::t) = List.filter Char.isDigit (explode "1a2b3c4d")`

   (c) `val t = map (fn (a,b) => a<b) [(4+0,2*2), (1,2), (size "ab", size "bc")]`

7. Assume the following function definitions. (8 points)

```
(* g : ('a * 'a -> bool) -> 'a list -> 'a list list *)
fun g cmp [] = []
  | g cmp (x::xs) =
        let (* f : 'a * 'a list * 'a list * 'a list list -> 'a list list *)
            fun f (x, [], zs, zss) = rev(x::zs)::zss
              | f (x, y::ys, zs, zss) =
                  if cmp(x, y) then f(y, ys, x::zs, zss)
                               else f(y, ys, [], rev(x::zs)::zss)
        in f(x, xs, [], []) end;
```

   What is the value of `x` after evaluating the following (independent) value-definitions?
   Complete the incomplete head-comment.

   (a) `val x = g op> [4,3]`

   (b) `val x = g op< [4,3]`

   (c) `val x = g op> [3,4,3,4]`

   (d) `val x = g op< [1,2,2,5,6,5,6,0,5,2,2]`

   (e) `val x = map implode (g op< (explode "abcbgefgfh"))`

   (f)
```
(* g cmp xs = the list of those lists,
which are ...... xs list .... according to cmp ... *)
```

8. Assume the following datatype-declaration. (8 points)

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

   An `(x,y)` pair is „heavy-headed" if `x > y`. Write a function `heavyheaded` which, when applied to an `(int*int) G` argument, returns the number of heavy-headed pairs found in that data structure. Try to make your solution efficient and prefer the use of higher-order functions.

```
(* heavyheaded : (int * int) G -> int
   heavyheaded t = the number of heavy-headed pairs found in t *)
```

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