Declarative Programming, Midterm exam, 26th April 2006
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.isAlpha  : char -> bool
op@           : 'a list * 'a list -> 'a list                | rev           : 'a list -> 'a list
op::          : 'a * 'a list -> 'a list                     | 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>(#"a", "b"), (1, 2) <> (1, 2, 3), true = false]`

   (b) `(2*3 = 3+3, chr 95, ~9) = (6*1, "b", 0-5-4)`

   (c) `foldl op@ [] [4, 2, 6, 4, 1, 2.0]`

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

   (a) `val (_::_::_::x) = explode "ap" @ rev [#"e", #"l", #"p"]`

   (b) `val (_::x::_) = map Char.isAlpha (explode "4a3r2a1d")`

   (c)
   ```
   val x =
   List.filter (fn (b, a) => a > b) [(7, 3*3), (1, 2), (ord #"Z", ord #"A")]
   ```

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

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

   (a) What is the value of `x` after evaluating the following (independent) value-definitions?

   (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) Show the evaluation steps of `comb([1,2,3,4], [2,3,4])`, using the substitution model and eager evaluation!

8. Three neighbouring elements of a list are called sum-triple if the sum of the first two is equal to the third, and are called dif-triple if the difference of the first and second is equal to the third. Write an SML function called `sumdif` which tells if a list contains sum-triples or dif-triples. You may define auxiliary functions only with appropriate head-comment! (9 points)

   ```
   (* sumdif : int list -> bool
      sumdif zs = true, if zs constains sum-triples or dif-triples
                  otherwise false
   *)
   ```

   Examples:
   ```
   sumdif [1,2,4] = false;
   sumdif [1,2,~1] = true;
   sumdif [1,2,3,~1] = true;
   sumdif [1,2,3,~1,2] = true;
   sumdif [1,2,4,~3,4] = false;
   sumdif [1,1] = false;
   sumdif [1] = false;
   sumdif [] = false;
   ```