

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	real	: int -> real
map	: ('a -> 'b) -> 'a list -> 'b list	List.last	: 'a list -> 'a
op@	: 'a list * 'a list -> 'a list	ord	: char -> int
op::	: 'a * 'a list -> 'a list	rev	: 'a list -> 'a list
op^	: string * string -> string	tl	: 'a list -> 'a list

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

- (a) [(1 = real 2), op^("a", "bc") <> #"a", [1\*4] = [4\*1]]
- (b) (ord #"B", 2\*4 = 4-2, ~4.0) = (#"B", true, ~3+1)
- (c) foldl (fn (a,b) => explode a ^ b) [" "] ["one", "zwei", "drie"]

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

- (a) val (\_::s::\_::\_) = explode "ho" @ rev(tl(explode "tel"))
- (b) val (r::s) = List.filter (fn (a,b) => (a>=b)) [(4+0,2\*2), (2,2-1), (2-1,2)]
- (c) val s = map List.last [explode("3c4d"), [#"Q"], rev(explode("let")) @ [#"z"]]

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

```
(* g : int list * int -> int list      f : int list * int list -> int list *)
fun g (n, xs) =
  let fun f (a::b::bs, zs) =
        if b-a=n then f(b::bs, 10*a+b::zs) else f(b::bs, zs)
      | f (_, zs) = rev zs
  in f(xs, []) end;
```

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

- (a) val x = g(1, [1,2,3,4,5,6])
- (b) val x = g(2, [1,3,4,6,8,9])
- (c) val x = g(~3, [1,~2,~5,~8,~5,~2,1])
- (d) val x = g(0, [1,~2,~2,4,4,4,5,6,7])

Complete the incomplete head-comment.

(e) (\* g (0,xs) = the list of those elements of xs multiplied by ..., which are ... \*)

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

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

We call an (a, b, c) 3-tuple a hollowtriple, if  $a * c > 0 \wedge b = 0$ . Write a function called htriples, which when applied to an (int\*int\*int) H argument, returns the hollowtriples found in it, in their original order in a list. Try to make your solution efficient and prefer the use of higher-order functions. You can use auxiliary functions if you write proper head-comment for them. (8 points)

```
(* htriples : (int * int * int) H -> (int * int * int) list
   htriples t = the hollowtriples in t, in their original order *)
```

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