General Computer Science I (320101) Fall 2011
Assignment 4: Practicing SML
(Given Oct. 7., Due Oct. 13.)

20pt

Problem 4.1 (Euclid on the net)

The network administrator of JUB tries to compute some metrics of the local network. Therefore, he captured, for every computer i on the network, a list of messages l_i that it transmits. He first computes the total message length t_i for every computer and then wants to deduce the greatest common divider of these lengths.

Help him by writing an SML function `computeGCD` that takes the list of lists of messages and returns the required common divider.

Example:

```
- computeGCD [["games", "wire", "address", "linuxbox"],
              ["foo","operation", "remote"]];
val it = 6 : int;
```

Note: You are required to use `foldl`, `foldr`, or `map` wherever it is possible.

30pt

Problem 4.2 (Checking relation properties)

Given a set A and a relation R on this set, create SML functions to check whether R is reflexive, symmetric, transitive, equivalence relation, and linear order. Each function should take two arguments: a list of elements (the set A) and a list of pairs (the members of R), and should return a boolean value.

Test examples:

```
- reflexive([1,2,3],[(1,1),(3,3),(2,2)]);
val it = true : bool
- symmetric([1,2,3],[(1,3),(3,1)]);
val it = true : bool
- transitive([1,2,3],[(1,3),(3,1)]);
val it = false : bool
- transitive([1,2,3],[(1,2),(2,3),(1,3)]);
val it = true : bool
- equivalenceRelation([1,2,3],[(1,1),(2,2),(3,3),(1,2),(2,1)]);
val it = true : bool
- linearOrder([1,2,3],[(1,2),(1,3),(2,3)]);
val it = true : bool
```

Hint: For shorter solutions, you may make use of higher-order SML functions rather than recursion everywhere. Consult `http://www.standardml.org/Basis/list.html#LIST:SIG:SPEC` for a description of a variety of functions on lists.