Declarative Programming, Midterm exam, 23rd April 2008, 12.15–13.45
Total time available: 90 minutes, total score: 60
Prolog (30 points)

When writing a Prolog program, you may use all of the Prolog predicates which were either defined during the lectures or are built-ins. Please refer to the sub-exercises using their corresponding identifier (for example 2.b).

1. What will be the results of the following Prolog goals (error, failure, success)? In case of success, specify the values of the named variables. The goals are given to the Prolog interpreter separately and on their own. (5 points)

   (a) `X = 1+6+4, \+ X = 11.`

   (b) `A is 6-1, 6 is A+1.`

   (c) `append([_X,2|A],[h],[e,f,g]).`

   (d) `Q is 1+R, R = 4.`

   (e) `X is 4-3, 2+X =:= 4-X.`

2. Specify the canonical forms (or draw the corresponding tree structures) of the left and right hand sides of the following equations. In the case of named variables, give the resulting variable substitutions. (9 points)

   (a) `.(R,[a,1]) = [C,C,B|[]].`

   (b) `f(A+B, [x,2|_]) = f(2+C, [B,A]).`

3. Let us assume that we have loaded the following program in the Prolog system:

```
p([A|_], C, B) :-
     A >= B,
     C is A*B.
p([_|As], C, B) :-
     p(As, C, B).
```

   What will the Prolog system answer if we ask the following questions (what will be the substitutions for variable `X`)? Enumerate all solutions in the same order as the system would, separated by a semicolon. If there is no solution, write `{no}`.

   (a) `p([2,3], X, 3).`

   (b) `p([2,3,5], X, 9).`

   (c) `p([6,2,3], X, 1).`

   (d) `p([20,12,4,3], X, 5).`

   (e) `p([4,11,22,3,40,6], X, 10).`

   Consider the following predicate which is based on the one above:

```
% p(L, Z): Z is such an element of the list L that ...
p(L, Z) :- p(L, Z, 1).
```

   (f) Describe the declarative meaning of predicate `p/2`, i.e. complete the sentence given above. In what order does the predicate generate the solutions? (8 points)

4. We have a list of $A - B$ pairs where A and B are integers. By the definition of lexicographic ordering the pair $A - B$ is greater than the pair $C - D$ if and only if $A > C$ or $A = C$ and $B > D$. Implement a **Prolog procedure** called `pgreater` that takes a list of $A - B$ pairs as input and enumerates the elements of this list that are greater than the previous pair with respect to the ordering defined above. You may define auxiliary procedures, but you have to provide their specification in a head comment. (8 points)

```
| ?- pgreater([1-1], X).          ⇒ no
| ?- pgreater([1-3,1-4], X).      ⇒ X = 1-4 ; no
| ?- pgreater([1-3,1-3,1-2], X). ⇒ no
| ?- pgreater([0-3,0-4,1-2], X). ⇒ X = 0-4 ; X = 1-2 ; no
| ?- pgreater([0-3,1-4,1-5,0-6,2-1,1-2], X).
                                  ⇒ X = 1-4 ; X = 1-5 ; X = 2-1 ; no
```

Declarative Programming, Midterm exam, 23rd April 2008, 12.15–13.45
Total time available: 90 minutes, total score: 60
Standard ML (30 points)

When the task is to write a function, all standard functions of SML and the functions defined during the lectures can be used. Please refer to the sub-exercises using their corresponding identifier (for example 6.b).

The types of the standard functions which appear in the tasks are the following:

```
explode      : string -> char list                | round         : real -> int
List.filter  : ('a -> bool) -> 'a list -> 'a list | Char.isLower  : char -> bool
map          : ('a -> 'b) -> 'a list -> 'b list   | ord           : char -> int
op::         : 'a * 'a list -> 'a list            | real          : int -> real
tl           : 'a list -> 'a list
```

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

   (a) `[7 > 3.3, #"a" < #"b", 2-1]`

   (b) `(ord "C", 7+5, 13-10, 42) = (97, 12, 23)`

   (c) `map (fn z => (1-z)) (3.1, 5.3, 1.0)`

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

   (a) `val (_::k) = tl(tl(explode "Alice"))`

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

   (c) `val (_::_::k::_) = map Char.isLower (explode "AlIcE")`

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

```
fun hipp (x::xs, y::ys) = (y, x) :: hipp(ys, xs) | hipp _ = []
fun h zs = map (fn (b,a) => b-a) (hipp(tl zs, zs))
```

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

   (a1) `val x = h [1,2,3,4,5]`

   (a2) `val x = h [~1]`

   (a3) `val x = h [~1,1]`

   (a4) `val x = h []`

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

8. We have an list of `int * int` pairs. By the definition of lexicographic ordering the pair (a, b) is greater than the pair (c, d) if and only if $a > c$ or $a = c$ and $b > d$. Implement an **SML function** called `pgreater` that takes an `(int * int) list` as input and returns a list containing the elements of the input that are greater than the previous pair with respect to the ordering defined above. You may define auxiliary functions, but you have to provide their specification in a head comment. (9 points)

```
(* pgreater : (int * int) list -> (int * int) list
   pgreater ps = the list of pairs found in ps that are greater
                 than the preceding pair with respect to lexicographic
                 ordering
*)
```

   Examples:

```
pgreater [(1,1)] = []
pgreater [(1,3),(1,4)] = [(1,4)]
pgreater [(1,3),(1,3),(1,2)] = []
pgreater [(0,3),(0,4),(1,2)] = [(0,4),(1,2)]
pgreater [(0,3),(1,4),(1,5),(0,6),(2,1),(1,2)] = [(1,4),(1,5),(2,1)]
```