Declarative programming, midterm retake, 3rd May, 2005, Budapest
Working time: 90 minutes, total score: 60
Prolog, group „A” (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 = 4+5, Z is X+1.`
   (b) `append([m,R|_],[],[A,p,q]).`
   (c) `Z = 1+3, Z = X, \+ X = 4.`
   (d) `X = a+b, Z is X - c.`
   (e) `A = 2, 1+5 = 4+A.`

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) `w(2+M, [K|[]], c) = w(K, [N], M).`
   (b) `[f(A,B),B*A+4,c-C|W] = .(f(3,6),[C,_-_]).`

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

```
p([X|_], B, Y) :-
   \+ X = B,
   Y = X.
p([X|Xs], B, Y) :-
   abs(X) < B,
   C is -abs(X),
   p(Xs, C, Y).
p([X|Xs], B, Y) :-
   abs(X) >= B,
   p(Xs, B, Y).
```

   What will the Prolog system answer if we ask the following questions (what will be the substitutions for variable `A`)? 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], 2, A).`
   (b) `p([1,2], -3, A).`
   (c) `p([2,2,2], 2, A).`
   (d) `p([3,4,2,-6,-2], 3, A).`
   (e) `p([5,3,-2,3,2,5,-2], 3, A).`

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

```
% p(L, Z): Z is an element of list L, such that...
p(L, Z) :- p(L, 0, Z).
```

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

4. In a list of integers we call three neighbouring elements a *bigtriple*, if the sum of these three elements is bigger than a given value. Write a Prolog predicate (`bigtriple`) which gets a list of integers and an integer (the given value) as input, and which enumerates the bigtriples as output, in the form of a structure `bt/3`. You may define auxiliary predicate(s), provided that you give a head comment for them. (8 points)

```
% bigtriple(L, B, H): H is a triple in L in the form of an bt/3 structure
% such that the sum of triple is bigger than B
% L and B are input parameters, H is an output parameter

| ?- bigtriple([1,4], 2, H).       ⇒ no
| ?- bigtriple([1,2,1], 2, H).     ⇒ H = bt(1,2,1) ; no
| ?- bigtriple([1,2,2,3], 6, H).   ⇒ H = bt(2,2,3) ; no
| ?- bigtriple([3,4,5,2], 1, H).   ⇒ H = bt(3,4,5) ; H = bt(4,5,2) ; no
| ?- bigtriple([-2,8,5,1,1,2], 6, H).
                                   ⇒ H = bt(-2,8,5) ; H = bt(8,5,1) ; H = bt(5,1,1) ; no
```