Declarative programming, midterm exam, 22nd April, 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-excercises 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) Z = 1+2, A is Z+1, A = 6-2.
(b) append([_,A],[_|_],[a,2,3]).
(c) Z =:= 3+1, A is Z*2.
(d) D is 6-1, \+ D > 0.
(e) A+B=3*4/2+1.
```

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) [A, 2+4 |D] = .(4, [B+A, e]).
(b) r(A+4*5, [3, B|C], s) = r(B, [_, ], A).
```

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

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([1,2,4], 2, X).
(b) p([1,5], 5, X).
(c) p([3,2,4,2], 0, X).
(d) p([1,5,3,4,4,5], 6, X).
(e) p([1,1,4,5,3,2,1,6], 4, X).

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. We call a structure of the form K-V a pair. We call the sum of the first elements of two pairs the keysum of the pairs. The distance of two pairs is defined as the absolute value of the difference of their second elements. Write a Prolog predicate (closer) which gets a list of pairs K-V as input and as output enumerates the keysum of those adjacent pairs, whose distance is smaller than a given value (the second argument of the predicate). You may define auxiliary predicate(s), provided that you give a head comment for them.

```
% closer(L, B, E): E is a keysum of two adjacent pairs in the list L,
% such that the distance of the pairs is < B
% L and B are input parameters, E is an output parameter
| ?- closer([1-4], 2, E). \Rightarrow no
| ?- closer([1-4,3-5], 2, E). \Rightarrow no
| ?- closer([1-4,2-7,3-5], 2, E). \Rightarrow no
| ?- closer([1-4,2-7,3-5], 2, E). \Rightarrow no
| ?- closer([3-4,1-6,10-6], 1, E). \Rightarrow E = 11; no
| ?- closer([1-4,2-5,5-11,6-7,2-8], 5, E).
\Rightarrow E = 3; E = 11; E = 8; no
```

(o points)

(8 points)