1. Consider the following (naive) implementation of reverse

```
reverse([],[]).
reverse([X|Xs],Zs) :-
        reverse(Xs,Ys),
        append(Ys,[X],Zs).
```

   – Show that the SLD proof tree of any ground query reverse(Xs,Ys) is in $\mathsf{O}(|Xs|^2)$. (5 pts)

   – Implement reverse more cleverly such that that size of the proof tree of any ground query becomes linear in $|Xs|$. (5 pts)

2. Implement a predicate *duplicate*/3 that duplicates the elements of a list a given number of times. For example the query duplicate([a,b,c],2, Xs) should deliver the answer Xs = [a, a, b, b, c, c]. Use difference-lists in your implementation, where you can assume that \ seperates difference lists. (10 pts)

3. Consider the following Prolog program.

```
foo(X,Y) :-
        foo([X|Xs]\Xs,Y,[X]).

foo([]\[],Y,Visited) :-
        !, fail.
foo([A|Xs]\Ys,A,Visited).
foo([A|Xs]\Ys,B,Visited) :-
        setof1(N,edge(A,N),Ns),
        foo2(Ns,Visited,Visited1,Xs\Ys,Xs1),
        foo(Xs1,B,Visited1).

foo2([N|Ns],Visited,Visited1,Xs,Xs1) :-
        member(N,Visited),
        foo2(Ns,Visited,Visited1,Xs,Xs1).
foo2([N|Ns],Visited,Visited1,Xs\[N|Ys],Xs1) :-
        \+ member(N,Visited),
        foo2(Ns,[N|Visited],Visited1,Xs\Ys,Xs1).
foo2([],V,V,Xs,Xs).

setof1(Template,Goal,Set) :-
        setof(Template,Goal,Set).
setof1(Template,Goal,Set) :-
        \+ setof(Template,Goal,Set), !, Set = [].
```

- Give a declarative reading of the program. (6 pts)

  Hint: The predicate *edge* represents the edge relation of a graph.

- The meaning of the program changes if *setof1* /3 is replaced by the system predicate *setof* /3. Give an example of a goal that succeeds in the original program, but fails in the altered program. (4 pts)

4. Implement (part of) the *Knight's tour problem*: how can a knight jump on an $N \times N$ chessboard in such a way that it visits every square exactly once?

   Hint: Represent the squares by pairs of their coordinates of the form X/Y, where X and Y are integers between 1 and N. It suffices to implement the relation jump(N,X/Y,U/V) to express the fact that a knight can jump from X/Y to U/V on a $N \times N$ chessboard. (10 pts)

5. Determine whether the following statements are true or false. Every correct answer is worth 1 points. (10 pts)

statement	**yes**	**no**
A rule is a universally quantified logical formula of the form $A \leftarrow B_1, B_2, \ldots, B_n$, where A is a goal and for all $i = 1, \ldots, n$: B_i is a goal.	☐	☐
An SLD-refutation is a finite SLD-derivation ending in the goal to be proven.	☐	☐
Logic programming is a declarative programming paradigm, that is, the computation of a function is made a first-class citizen.	☐	☐
The declarative semantics of a program P is the minimal model of P.	☐	☐
The order of goals is irrelevant in the computation model of logic programming, but not the order of rules.	☐	☐
The order of goals and the order of rules is irrelevant in the computation model of Prolog.	☐	☐
Prolog is a language without types and the main technique to manipulate data is unification.	☐	☐
Difference lists are ineffective if the generation of different sections of a list depend on each other.	☐	☐
A meta-interpreter in Prolog interprets Prolog terms on the Warren abstract machine.	☐	☐
The predicate *bagof* (*Template*,*Goal*,*Bag*) unifies *Bag* with the alternatives of *Goal* that meet *Template*.	☐	☐