1. *Solution.* The following program solves the exercise:

```
% graph G
edge(a,a).
edge(a,b).
edge(b,b).
edge(b,c).
edge(c,d).
edge(d,e).

% connected(X,Y) is true if X is connected to Y in G
%
connected(X,X).
connected(X,Z) :-
        edge(X,Y),
        connected(Y,Z).
```

   We show that the height of the SLD-tree is (grossly) bounded by the number of vertices n in the graph G. Suppose the height is strictly larger than n; then there exists a path in G whose length is larger than n. Contradiction to the assumption that G is acyclic. This argument works for ground and non-ground queries. □

2. *Solution.* (a) The given goal order leads to non-termination, even if both arguments are complete lists. (b) The difference can be easiest seen from the queries given below:

```
:- \+ sublist([a,d],[a,b,c,d]).
:- subsequence([a,d],[a,b,c,d]).
```

   □

3. *Solution.* Consider the following program:

```
isotree(nil,nil).
isotree(tree(X,Left1,Right1),tree(X,Left2,Right2)) :-
  isotree(Left1,Left2), isotree(Right1,Right2).
isotree(tree(X,Left1,Right1),tree(X,Left2,Right2)) :-
  isotree(Left1,Right2), isotree(Right1,Left2).
```

   □

4. *Solution.* Consider the following programs:

```
% prop/1 --> DCG that generates well-parenthesised propositional formulas
%
prop --> "p".
prop --> "q".
prop --> "r".
prop --> "true".
prop --> "false".
prop --> "~", prop(A).
prop --> "(", prop(A), "^", prop(B), ")".
prop --> "(", prop(A), "v", prop(B), ")".

% prop2/1 --> DCG that generates propositional formulas using
% the standard precedence
%
atom --> "p".
atom --> "q".
atom --> "r".
atom --> "true".
atom --> "false".
atom --> "(", prop2, ")".
unary  --> "~", unary.
unary --> atom.
and --> unary, "^", and.
and --> unary.
prop2 --> and, "v", prop2.
prop2 --> and.
```

□

5. *Solution.* Consider the following program:

```
palindrome(Xs) :- palindrome(q0,Xs,[]).

palindrome(q0,[X|Xs],Ys) :- palindrome(q0,Xs,[X|Ys]).
palindrome(q0,[X|Xs],Ys) :- palindrome(q1,[X|Xs],Ys).
palindrome(q0,[_X|Xs],Ys) :- palindrome(q1,Xs,Ys).
palindrome(q1,[X|Xs],[X|Ys]) :- palindrome(q1,Xs,Ys).
palindrome(q1,[],[]).
```

□

6. *Solution.*

statement	**yes**	**no**
An existential fact is a fact that contains existentially quantified variables.	☐	☑
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.	☑	☐
Data is structured in logic programs to obtain for example (i) better modularity or (ii) better organisation of the data.	☑	☐
Almost all, but not all basic elements of a relation database model can be expressed in Prolog.	☐	☑
Consider the standard implementation of *member*/2. Then any call to *member* terminates iff the second argument is a complete list.	☑	☐
A Prolog clause is called *iterative* if it has one recursive call and zero or more calls to system predicates that appear before the recursive call.	☑	☐
A cut fixes all choices between (and including) the moment of matching the rule's head with parent goal and the cut. If backtracking should reaches the cut, then the cut succeeds and the execution is continued with the clause *after* the clause containing the cut.	☐	☑
Constraint logic programs typically fall into two parts. In one part the pure relation is formuated, possible using constraints. In the second part the constraints are instantiated.	☑	☐
The predicate *bagof*($Template$,$Goal$,Bag) unifies Bag with the alternatives of $Goal$ that meet $Template$.	☐	☑
Time complexity of Prolog programs is measured via the number of inferences performed and real time consumption depends on the time used for unification attempts.	☑	☐

□