1. (4 p)

   Lista alla svar som Prolog kommer att ge på följande fråga, samt i vilken ordning de kommer genereras. (Endast svaren, motivering behövs ej.)

   ```
   ?- lookup(D, branch(branch(leaf(-2), leaf(5)), leaf(3))).
   ```

Lösning:

Först 3, sedan 5, sedan -2.

Rättning:

- 0 poäng om man har fel mängd svar (dvs inte inkluderat något av de tre svaren, eller inkluderat något mer svar).
- Om man angett svaren som t.ex. "`leaf(5)`" istället för "5" får man ett poängs avdrag från poängen nedan.
- 2 poäng om felaktig ordning

- 4 poäng om rätt ordning.

2. (10 p)

Beskriv kontrollflödet samt unifieringarna steg för steg för att beräkna den första lösningen till följande fråga:

```
?- lookup(kia, T), lookup(bo, T).
```

Lösning:

Fas 1.

1.1. Målet '`lookup(kia, T)`' utvärderas. Detta lyckas vid första klausulen med unifieringen '`T=leaf(kia)`'.

1.2. Prolog fortsätter till andra målet '`lookup(bo, T)`' med denna unifiering, dvs Prolog utvärderar nu '`lookup(bo, leaf(kia))`'.

1.2.1. Första klausulen undersöks igen. Detta misslyckas eftersom '`bo`' inte kan unifieras med '`kia`'. De resterande två klausuler misslyckas också, därför att en '`leaf`'-term inte kan unifieras med någon '`branch`'-term.

Fas 2.

2.1. Prolog backtrackar till första målet '`lookup(kia, T)`', och försöker nu matcha den mot andra klausulen. Detta lyckas för unifieringen '`T=branch(T1, leaf(kia))`', där '`T1`' är en nygenererad variabel.

2.2. Prolog forsätter med andra målet '`lookup(bo, T)`' med denna nya unifiering av `T`, dvs med '`lookup(bo, branch(T1, leaf(kia)))`'.

2.2.1. Första klausulen kommer inte matcha, men andra klausulen kommer matcha, och Prolog kommer nu utvärdera målet '`lookup(bo, leaf(kia))`'.

2.2.2. Detta failar, Prolog backtrackar och går nu över till tredje klausulen, vilket resulterar i att Prolog nu uvärderar '`lookup(bo, T1)`'.

2.2.3. Detta lyckas vid första klausulen med unifieringen '`T1=leaf(bo)`', och så lyckas hela frågan, med slutunifieringen '`T = branch(leaf(bo), leaf(kia))`'.

Rättning:

(a) 2 poäng för huvuddragen i fas 1:
`lookup(kia, T)` lyckas, sedan misslyckas `lookup(bo, T)` och Prolog backtrackar

(b) 2 poäng om rätt unifiering görs i fas 1.

(c) 2 poäng för huvuddragen i fas 2:
`lookup(kia, T)` lyckas, sedan lyckas `lookup(bo, T)` och (första) svaret är hittat.

(d) 2 poäng om kontrollflödet av `lookup(bo, T)` i fas 2 är korrekt beskrivet

(e) 2 poäng om rätt unifieringar görs i fas 2.

Exempel:

- Om man inte gjort fas 1 utan gått direkt på fas 2 men gjort denna korrekt får man 6 poäng.

3. (6 p)

Modifiera lookup för att maximalt optimera sökningen och undvika onödiga predikatanrop om bara ett svar önskas. Förklara för varenda ändring som du har gjort vad den åstadkommer.

Lösning:

Modifierad kod:

```
lookup(D, leaf(D1)) :- !, D=D1, !.
lookup(D, branch(_, T)) :- lookup(D, T), !.
lookup(D, branch(T, _)) :- lookup(D, T), !.
```

Förklaring av de olika ändringarna:

1. Vi flyttar unifieringen för '`leaf`' till höger om '`:-`' så att vi kan avgöra `lookup` för löv utan att behöva betrakta klausulerna som handlar om sammansatta träd.

   Första snittet kommer då avsluta sökningen om trädet är ett löv men datan inte är samma som den vi letar efter.

2. De tre resterande snitten skär av backtrackningen om sökningen lyckas. Snittet i sista klausulen gör egentligen ingen skillnad: eftersom detta är det sista kommer Prolog efter att detta utvärderats ändå inte söka något mer. Det är god stil att ha med detta i lösningen så att man inte behöver tänka på att lägga till det om man skulle kasta om raderna eller lägga till en fjärde klausul för lookup.

Inga separata snitt behövs vid fail, därför att om andra klausulen inte lyckas (dvs datan inte finns i högra delträdet), så måste man undersöka vänstra delträdet med sista klausulen, och om den failar, så finns det inga fler klausuler och Prolog kommer avsluta sökningen ändå. (Denna motivering till varför fler snitt inte läggs till behöver man ej ha givit i sitt svar.)

Rättning:

2 poäng för ändring (1), varav 1 poäng för ändringen av koden och 1 poäng för förklaringen.
4 poäng för ändring (2), varav 2 poäng för ändringarna och 2 poäng för förklaringen. Man behöver inte ha lagt till något snitt på sista raden.