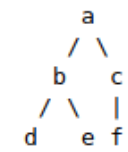
PROLOG EXERCISES

1. Assume given a set of facts of the form `father(name1,name2)` (`name1` is the father of `name2`).
    1. Define a predicate `brother(X,Y)` which holds iff `X` and `Y` are brothers.
    2. Define a predicate `cousin(X,Y)` which holds iff `X` and `Y` are cousins.
    3. Define a predicate `grandson(X,Y)` which holds iff `X` is a grandson of `Y`.
    4. Define a predicate `descendent(X,Y)` which holds iff `X` is a descendent of `Y`.
    5. Consider the following genealogical tree:

```
father(a,b).  % 1
father(a,c).  % 2
father(b,d).  % 3
father(b,e).  % 4
father(c,f).  % 5
```

whose graphical representation is:



Say which answers, and in which order, are generated by your definitions for the queries

```
?- brother(X,Y).
?- cousin(X,Y).
?- grandson(X,Y).
?- descendent(X,Y).
```

Justify your answers by drawing the SLD-tree (akas Prolog search tree) for each query, at least schematically.

2. Consider the following predicates.

```
likes(john, susie).                   /* John likes Susie */
likes(X, susie).                      /* Everyone likes Susie */
likes(john, Y).                       /* John likes everybody */
likes(john, Y), likes(Y, john).       /* John likes everybody and everybody likes John */
likes(john, susie); likes(john,mary). /* John likes Susie or John likes Mary */
not(likes(john,pizza)).               /* John does not like pizza */
likes(john,susie) :- likes(john,mary)./* John likes Susie if John likes Mary.
```

Write the following predicates **friend(X,Y), does_not_like(X,Y), enemies (X,Y).**

3. Consider the following database:

```
owns(k0, book('David Copperfied', 'Charles Dickens')).
owns(k1, book('Tale of Two Cities', 'Charles Dickens')).
owns(k2, book('Tale of Two Cities', 'Charles Dickens')).
owns(k3, dvd('Tale of Two Cities')).
owns(k4, book('Moby Dick', 'Herman Melville')).

borrowed(k2, 'Homer', 44). * he last value represent the date when the book is due.*
borrowed(k4, 'Homer', 46).
borrowed(k3, 'Lisa', 92).
borrowed(k0, 'Lisa', 92).
```

Write the following queries:
A) What books has Homer borrowed?
B) Libraries want to track overdue books. Write a predicate for checking when a borrowed item is overdue.