Problem 1

Consider the following DTD:
```
<!ELEMENT forward ((rule | fact)*)>
<!ELEMENT rule (if, then)>
<!ELEMENT fact (#PCDATA)>
<!ELEMENT if (#PCDATA)>
<!ELEMENT then (#PCDATA)>
```
Are the following XML elements valid with respect to this DTD?

1. ```
   <forward> </forward>
   ```
2. ```
   <forward> the weather </forward>
   ```
3. ```
   <forward>
        <rule> if it rains then the road gets wet </rule>
   </forward>
   ```
4. ```
   <forward>
        <rule>
             <if> it rains </if>
             <then> the road gets wet </then>
        </rule>
   </forward>
   ```
5. ```
   <forward>
        <fact> it rains </fact>
        <rule>
             <if> it rains </if>
             <then> the road gets wet </then>
        </rule>
   </forward>
   ```

Problem 2

Here is a DTD:
```
<!DOCTYPE X [
        <!ELEMENT X (Y, Z*)*>
        <!ELEMENT Y (Z+)
        <!ELEMENT Z (#PCDATA) ]>
```
1. Give an example of the smallest data set (i.e., the fewest number of elements) you can think of that is valid with respect to this DTD and includes at least one each of X, Y, and Z elements.
2. Write your answer as well-formed XML

Problem 3

Write a DTD and an XML Schema for storing an ancestor structure for a person. By an ancestor structure is meant the name and birthday of the person, and the same information recursively for his or her father and mother, grandfathers and grandmothers, etc., to an unlimited depth. The structure should allow stops at different depth in different branches, when some ancestor is not known any further. Provide one DTD (XML Schema) which does not use ID and IDREF(S) (**key** and **keyref**) and one which does.

Problem 4

Suppose you have to represent the information about parts. Each part has a name (unique), and a textual description. Parts may be simple or complex. A simple part has a color but no children subparts. A complex part has a number of children subparts (which can be simple or complex), each of which may be repeated. (E.g., a car has 4 wheels.) You can assume that each part can be a child subpart of at most one other part (so each part, together with its subparts, can be viewed as a tree). Do not assume any fixed number of levels of part composition. Define the DTD and the XML Schema of XML documents containing part information.