1. Suppose we are given five attributes: A, B, C, D, E and the following functional dependencies:
   \[ A \rightarrow C, \quad A \rightarrow B, \quad B \rightarrow C, \quad A \rightarrow E, \quad CE \rightarrow D, \quad DE \rightarrow C, \quad CE \rightarrow A. \]
   - Find a minimal equivalent set of functional dependencies. (You need not use any special algorithm, though you should be able to follow along based on the notes.)
   - Find a key of A, B, C, D, E given these functional dependencies.
   - How many keys are there.
   - Find a lossless, dependency-preserving third normal form decomposition of these attributes into table schema. That is, give the column headers of the tables in the form (X,Y,Z), (W,Z) etc.

2. A slight variant of the algorithm to produce canonical covers proceeds as follows: Given a set of FD’s with only one attribute on the right hand side of any FD, repeat the following two steps until there is no change:
   - eliminate all redundant FD’s,
   - remove all extraneous left hand side attributes.

   Give an example showing that the second step can introduce new redundant FDs (and is therefore useful, because those discovered FDs can then be removed).

3. You are given the task of designing a relational database for hotel reservations. Each hotel has several room types (e.g. singles, doubles, doubles with views). For that hotel and room type, there are a certain number of rooms available and the room type has a description for that hotel. (Thus the same room type may have a different description for different hotels.) Hotel reservations reserve a certain number of rooms of each type for certain dates. Numberavailable is the number of rooms of that type in the hotel and is constant over time. Numbertaken is the number taken of that type for the hotel for a particular date. (So Numbertaken is always less than or equal to Numberavailable.)

   Here is the initial schema (done by someone else):
   ```
   hotelroomres( hotelid, roomtypeid, description, numberavailable, date, numbertaken)
   ```

   Find the functional dependencies and decompose this table if necessary into a third normal form design. Identify the keys. Explain the advantages of your decomposition.