Schema Normalization BCNF

Given a relational schema and the set of functional dependencies (FDs), normalize the schema to BCNF.

Assume that all the FD are of the form where consequent (right side) has only one attribute.

Example input 1:

1. 6
2. 4
3. 1 2
4. 4
5. 4
6. 3
7. 1 2
8. 5
9. 2
10. 6

This input file is exactly same as the practice assignment.

Example output 1:

1. 3
2. 1 2 4 5
3. 2 6
4. 4 3

Line 1 specifies the number of relations in the resultant schema.

Each of the next lines describes one relation. It is sufficient to describe only the list of attributes.

Within each relation, the attributes are sorted in the increasing order of value.

Relations are sorted based on the lexicographic order.

There could be multiple correct ways to normalize the given schema. However, we will stick to one particular order of processing the FDs. This will help us to have a unique answer for the purpose of evaluation. You should process the FDs in the same order as they are specified in the input file. Check if the current FD violates the BCNF. If the answer is yes, split the relation in two parts. If the answer is no, move on to the next FD.

Note on code submission:

Make sure that all your code fits into a single file <roll number>.cpp

How your code should compile?

```
g++  <roll number>.cpp
```

How your code should run?

```
a.out < input_file_name
```

Your code should write the output on the stdout

A TA will collect your code on a USB drive. Make sure that you compute md5 hash of your code file before you submit. Use command md5sum <filename>.

Submit code to TA only when you are sure that it is your final code. We will not entertain requests to accept newer version of the code.

Preparing your system:

Any Linux flavor.

GNU C++ compiler version 11.2