# Modules

## Problem Statement

Tom has just enrolled into NUS to pursue a degree in Computer Science. Tom is interested in many of the CS modules that NUS has to offer, but while he is planning his schedule, Tom soon runs into an obstacle. Tom realises that many of the modules he is interested have prerequisites to fulfill. In turn, the prerequisite modules may also have their own set of prerequisites.

Understandably, Tom is confused by all this, and he wants to determine the appropriate order in which his modules should be taken. Will you be able to help Tom with this task?

## Input

The first line of input consists of the number of test cases *T* (1 <= *T* <= 20). For each test case, the first line consists of a single integer *N* (1 <= *N* <= 1000), the number of modules. This is followed by *N* lines, each consisting of the description for a module. Each module description begins with the name of the module and is followed by a space separated list of the module's unique prerequisites. It is guaranteed that no module will have itself as a prerequisite. Every module name is a string of up to 50 non-whitespace characters using the alphabet (both lowercase and uppercase characters) and digits (0 – 9).

Note: As this lab is **meant to be done after PE**, it will be good for everyone to be acquainted with using BufferedReader to read in line-based input. The readLine() method of BufferedReader can be used to take in a line as a String, and subsequently the split() method of String can be used to split the line into an array.

The relevant documentation for BufferedReader and String can be found here:

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/BufferedReader.html

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html

## Output

For each test case, output the order of modules that allows every module to be taken after their prerequisites. If there are multiple possible orderings, output the lexicographically smallest order. Note that sometimes, there can be bugs, and more modules show up in the prerequisites list than intended. If there is a list of modules that cannot be ordered, output the statement "BUG FOUND". Put a blank line between each pair of test cases.

Sample Input 1

```
1
9
CS1010S
MA2001
MA2002
CS1231S
CS2040 CS1010S
CS3243 CS2040 CS1231S
CS3244 MA2001 MA2002 CS2040 ST2131
ST2131 MA2002
CS4248 CS3243 MA2002 ST2131
```

Sample Output 1

```
CS1010S
CS1231S
CS2040
CS3243
MA2001
MA2002
ST2131
CS3244
CS4248
```

Sample Input 2

```
2
2
CS1010S CS2040
CS2040 CS1010S
2
CS2040C CS1010E
CS1010E
```

Sample Output 2

```
BUG FOUND


CS1010E
CS2040C
```