Problem 175: You Can Depend on Me

Difficulty: Medium

Author: Matt Marzin, King of Prussia, Pennsylvania, United States

Originally Published: Code Quest 2022

Problem Background

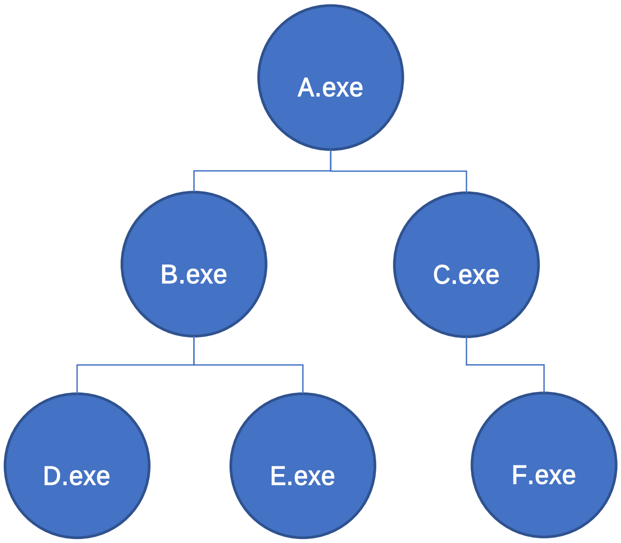
DevSecOps - an abbreviation of development, security, and operations - focuses on making everyone accountable for the security of an application, from the start of its development to its release and beyond. As a DevSecOps engineer at Lockheed Martin, you're responsible for maintaining the software that your team relies on to develop your application. This infrastructure includes tools that constantly evaluate the application for bugs, vulnerabilities, and other issues that should be resolved before delivering an application to the customer.

Unfortunately, the programs that make up this infrastructure are tightly coupled to each other, and when one process crashes, all the processes supporting that process must also be restarted. Also unfortunately, this happens on a somewhat regular basis as your coworkers continue to develop the application and add new features. Your task is to implement a program that monitors the running processes for a failure, then once one occurs, restart that program and all of its dependencies.

Problem Description

Your development infrastructure consists of a number of related processes. When a program becomes unresponsive and needs to be rebooted, all of its dependent processes need to be restarted first. If those processes have dependencies of their own, they need to be restarted as well, and so on.

For example, in the diagram below, a total of five programs are supporting the system. B.exe depends on both D.exe and E.exe to function; if B.exe crashes, D.exe and E.exe will also need to be restarted, before B.exe itself is restarted.



If A.exe were to stop responding, the entire infrastructure would need to be restarted, since all other programs are below it in the dependency tree. Since D, E, and F are the lowest level dependencies, they will again need to be restarted first; B and C can then be restarted before finally restarting A. When multiple programs exist at the same dependency level (as with D, E, and F), programs should be restarted in alphabetical order by name.

Your program will need to read in a list of program dependencies and a list of failure events that must be handled. For each event, it will need to report the programs that must be restarted, and in what order to restart them. Programs may have more than one dependency, as shown above, and more than one program may depend upon another. There will be no circular dependencies; that is, no program will ever depend upon itself, no matter how indirectly.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include the following:

- A line containing two positive integers, separated by a space, representing:
  - D, the number of dependencies
  - E, the number of failure events
- D lines containing program dependency information. Each line consists of two strings, separated by a space. The program named by the first string depends on the program named by the second string. All program names will contain lowercase letters and will end with the '.exe' extension.
- E lines containing failure event information. Each line will contain the name of a program, previously listed in the dependency list, that has failed and must be restarted.

```
1
5 2
a.exe b.exe
a.exe c.exe
b.exe d.exe
b.exe e.exe
c.exe f.exe
b.exe
a.exe
```

Sample Output

For each test case, your program must print the commands to be executed for each failure event, one per line, in the order in which they are to be executed. First, restart commands consist of the phrase "restart", followed by a space and the name of the program to restart. These should be followed by a single 'exit' command to complete the recovery process.

```
restart d.exe
restart e.exe
restart b.exe
exit
restart d.exe
restart e.exe
restart f.exe
restart b.exe
restart c.exe
restart a.exe
exit
```