

# Problem M

## Managing Packaging


**Problem ID:** managingpackaging

**CPU Time limit:** 1 second

**Memory limit:** 1024 MB

**Author:** Greg Hamerly

**Source:** Baylor Competitive Learning course

**License:** 

Modern operating system distributions have tools to manage installed software, making it easy to keep the software up-to-date. Putting different pieces of software into ‘packages’ that can be installed separately keeps things simple and avoids duplicate effort and code. But this means that a package may depend on other packages, requiring their installation before that package can be installed. For example, many programs need to use ‘libc,’ which contains standard C-library functions. To keep track of these dependencies, many distributions use some sort of package manager.

When a user wants to install a new package (or set of packages), a package manager takes care of the headache of tracking down which packages are required to install the desired package. Of course, those may also depend on other packages.

For this problem, determine an order (if one exists) that allows a given list of packages to be installed. No package should be installed before all the packages it depends on are installed. You may assume that, at the beginning, no packages have been installed.

### Input

Input consists of up to 10 test cases. Each test case start with a number  $1 \leq n \leq 1000$ , which is the number of packages the user wants to install. This is followed by  $n$  lines which describe  $n$  packages. Each package description starts with the name of the package and is followed by a space-separated list of the package’s unique dependencies. Each package has at most 20 dependencies, and each is one of the other  $n - 1$  packages. Each package name is a string of up to 40 non-whitespace characters using the English alphabet (a-z, A-Z), digits (0-9), as well as the characters `_`, `-`, `.`, and `+` (i.e. underscore, minus, period, and plus). Input ends when  $n$  is zero.

### Output

For each test case, output the order of package installation that allow them all to be installed after their respective dependencies. If there are multiple possible orderings, then give the ordering that is lexicographically first (using ASCII values for string ordering). If there is some group of packages that are not able to be ordered within the list, output ‘cannot be ordered’ instead of ordering the packages. Put a blank line between each pair of test cases.

#### Sample Input 1

```
14
libattr
vim-X11 vim-common gtk2 libattr
vim-common
gtk2 libtiff atk pango glib2
libtiff zlib libjpeg
atk
pango xorg-x11-libs freetype glib2
glib2
zlib
libjpeg
xorg-x11-libs grep freetype
grep pcre
pcre
freetype
3
emacs xorg-x11 lisp
xorg-x11
lisp emacs
0
```

#### Sample Output 1

```
atk
freetype
glib2
libattr
libjpeg
pcre
grep
vim-common
xorg-x11-libs
pango
zlib
libtiff
gtk2
vim-X11

cannot be ordered
```

#### Sample Input 2

```
4
atk pango
glib2 pango
grep atk
pango
0
```

#### Sample Output 2

```
pango
atk
glib2
grep
```

