A package management system takes care of dependencies of packages on installations and removals. Most packages depend on additional packages to function. Such dependency between packages is acyclic, i.e. no two packages depend on each other. Also, a package may be dependency of multiple packages.

Users will command the system to install or remove packages.

When the user installs a package, the system automatically installs all its dependencies as well. However, the system will only keep one copy of each package. If a package is already present on the system (either because the user installed it or the system automatically installed it), the system will not install the package again.

When the user removes a package, the system first check if it is dependency of other packages on the system. If so, the system ignores the removal. If not, the system removes the package. Then the system automatically finds the dependencies of the removed package, and removes each of the dependency as long as it is neither installed directly by user command nor depend by other packages on the system.

You are going to write a program to simulate a package management system that supports the following operations:

- `INSTALL x` : install package x if it is not installed
- `REMOVE x` : removes package x and its dependencies
- `LIST` : list all installed packages

For simplicity, packages will be numbered as $x = 0, 1, 2, \cdots$. See hints section of this problem. The expected behavior of the package management system in sample IO is listed in details.

Input

The first line contains two integers n and q, the number of packages and number of operations.

The following n line describes the package dependencies. The i-th line contains the dependency of package i. The first integer m_i in the line is the number of dependency of package i. After that are m_i integers, being the dependencies of package i.

The following q lines are the operations. The format are mentioned in the problem description.

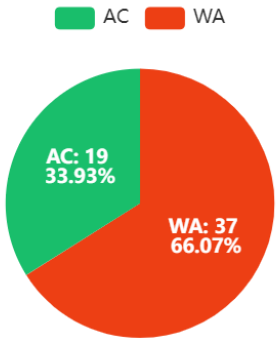
See hint section for detailed explanation on IO format.

Constraints

- For `INSTALL` operation, it will not ask you to install a package that is already installed.

ID	packdep
Time Limit	1000MS
Memory Limit	256MB
IO Mode	Standard IO
Created By	ta_redleaf
Level	Hidden
Score	100
Tags	Show

Statistic Details



- $0 \leq x < n$ for `INSTALL` and `REMOVE` operations

Output

Please output the result for each operation:

- For `INSTALL` operation, please print the number of newly installed packages (including the dependencies).
- For `REMOVE` operation, please print the number of removed packages (including the dependencies).
- For `LIST` operation, please print the number of all installed packages and followed by the packages in sorted order.

Sample Input 1

```
9 15
2 1 2
3 3 4 5
3 3 4 5
0
1 8
0
1 7
0
0
INSTALL 0
LIST
REMOVE 0
LIST
INSTALL 1
INSTALL 2
REMOVE 5
INSTALL 0
REMOVE 0
LIST
REMOVE 2
INSTALL 0
LIST
REMOVE 0
LIST
```

Sample Output 1

```
7
7 0 1 2 3 4 5 8
7
0
5
1
0
1
1
6 1 2 3 4 5 8
1
2
7 0 1 2 3 4 5 8
2
5 1 3 4 5 8
```

Hint

There are 9 packages:

- Package 0 depends on two packages, which are 1 and 2.
- Package 1 depends on three packages, which are 3, 4, and 5.
- Package 2 depends on three packages, which are 3, 4, and 5.
- Package 3 does not depend on any package.
- so and so on for package 4 to 8.

There are 15 operations, the explanation of the first ten operations are shown below:

v. ... and finally seven packages are installed. Therefore print 7.

2. List:
   i. There are currently seven packages installed. Print 7 and then the installed packages $0, 1, 2, 3, 4, 5, 8$.
3. Remove 0:
   i. 0 is not dependency of any package on the system.
   ii. 0 is removed, check its dependency 1 and 2.
   iii. No other installed packages depend on 1, so it can be removed.
   iv. 1 is removed, check its dependency 3, 4, and 5.
   v. ... and finally seven packages are removed. Therefore print 7.
4. List:
   i. There are currently zero packages installed. Print 0.
5. Install 1:
   i. Similarly, 1 along with $3, 4, 5, 8$ are installed. Therefore print 5.
6. Install 2:
   i. 2 should be installed after $3, 4, 5, 8$. However $3, 4, 5, 8$ are already installed.
   ii. Finally there is only one newly installed package. Therefore print 1.
7. Remove 5:
   i. Since 5 is dependency of 1 and 2, so it is still needed and cannot be removed.
   ii. Therefore zero packages are removed. Print 0.
8. Install 0:
   i. 0 should be installed after $1, 2$. Currently both of them are already installed.
   ii. Finally there is only one newly installed package. Therefore print 1.
9. Remove 0:
   i. 0 is not dependency of any package on the system.
   ii. 0 is removed, check its dependency $1, 2$.
   iii. $1, 2$ are user installed packages (fifth and sixth command). They cannot be removed automatically by the system.
   iv. Finally only one package is removed. Therefore print 1.
10. List:
    i. There are currently six packages installed. Print 6 and then the installed packages $1, 2, 3, 4, 5, 8$.

Language: C Theme: Solarized Light

```
1
```

```
9 15
2 1 2
3 3 4 5
3 3 4 5
0
1 8
0
1 7
0
0
INSTALL 0
LIST
REMOVE 0
LIST
INSTALL 1
INSTALL 2
REMOVE 5
INSTALL 0
REMOVE 0
LIST
REMOVE 2
INSTALL 0
LIST
REMOVE 0
LIST
```