

2366 Algernon's Noxious Emissions

One of the greatest alchemists of the lower Middle Renaissance, Algernon da Vinci (one of Leonardo's lesser-known cousins), had the foresight to construct his chemical works directly over a fast-running stream. Through a series of clever pipes and sluices, he routed portions of the stream past each of the tables where his alchemists prepared their secret brews, allowing them to dispose of their chemical byproducts into the waters flowing by the table.

As Algernon's business grew, he even added additional floors to his factory, with water lifted to the higher floors by treadmill-powered pumps (much to the dismay of the apprentices who found themselves assigned to pump duty). The pipework for the entire disposal system became quite complex. It was even rumored by some that the pipes actually circled back in some places, so that a particularly odorous compound flushed away from one table might return to that very same spot a few minutes later.

All was not well, however. Algernon's factory suffered from a series of mishaps, minor explosions, gas clouds, etc. It became obvious that chemicals dumped at one table might react violently with other chemicals dumped from another table downstream. Algernon realized that he needed to trace the possible chemical flows through his factory.

Write a program to aid Algernon in this task. To preserve the secrecy of the chemical processes that are Algernon's stock in trade, all chemicals will be identified by a single upper-case letter. All tables are identified by positive numbers in the range $1 \dots N$, where N is the number of tables.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

Line 1: # of work tables, integer (henceforth referred to as N). $N < 50$.

Lines $2 \dots N + 1$: For each table:

- a list of chemicals dumped into the stream at that table, followed by
- a list of chemicals that, if they appeared at that table, would be harmlessly neutralized by the reactions at that table, allowing no further trace of that chemical to flow downstream (we will assume that the rate of work at each table can be adjusted as necessary to guarantee total neutralization of whatever amount of these chemicals arrive from upstream).

Each of these lists is given as a series of upper-case alphabetic characters. The only exception is that a special list, consisting of a single '.' character, will be used to denote an empty list. The two lists are separated from one other by one or more blanks. The same chemical will never appear in both lists.



Lines $N + 2 \dots ?$: These lines provide a description of the pipeworks. Each line contains a pair of integers in the range $1 \dots N$, separated by one or more blanks:

$I J$

meaning that the table number I is upstream of table number J -anything dumped into the stream at table I or that arrives in the stream at table I and is not neutralized can then be counted on to arrive at table J .

No (I, J) pair will be listed more than once, but the pairs may occur in any order. I and J will never be the same number.

The end of input is signaled by a pair of zeros:

0 0

Note that if a table only receives water directly from the stream entering the building, that table will never occur in the second position of a pair. Similarly, any table that discharges only into the stream leaving the building will never occur in the first position of a pair.

Output

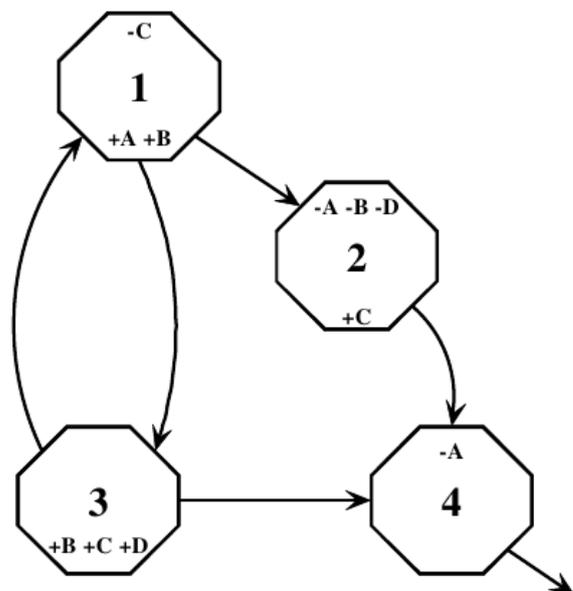
For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

There will be N lines of output, one for each table, in the same order as they appeared in the program input. Each line will contain the list of chemicals that can be expected at that table’s output. This list will be printed as a (possibly empty) list of upper-case alphabetic characters between two colons (:). No empty spaces should be printed on the line. The characters in the list should be sorted in alphabetic order.

Note: For the figure at the right, an input file would be the sample below.

Sample Input

```
1
4
AB C
C BDA
BCD .
. A
1 2
2 4
3 1
1 3
3 4
0 0
```



Sample Output

```
:ABD:
:C:
:ABCD:
:BCD:
```