Music Festival

You are organizing a music festival and need to create a schedule for the performances. You have a list of artists that will perform at the festival, and you also have a list of time slots that are available for the performances.

The festival wants you to sort the list of artists, according to some rules. You will be given the list of time slots, in the order they should happen. Then, you'll have access to a list of artists; each artist will have one time slot for their performance.

The festival wants you to put the list of artists in order, sorted by their preferred time slot. There may be some artists who want to perform at the same time slot. If so, order them in the order that they appear on the original list. Some artists may have a 'wildcard' time slot, thereby requesting a time slot that was not on the original time slot list. Put these artists at the end, grouped by time slot, in the order that their time slots appear in the list of artists. There also may be time slots with no artists. They will probably be taken off the list and replaced by breaks later, but that's not your problem.

Input

You are given multiple test cases. Each test case begins with two integers, n ($1 \le n \le 1{,}000$) and m ($1 \le m \le 100{,}000$), where n is the number of available time slots and m is the number of artists. The time slots will be listed on the next n lines, with each time slot name being a single word consisting of 1 to 30 capital letters. All time slot names in any test case will be unique. Following the time slots, the next m lines will list the artists, with each line containing two words consisting of 1 to 30 capital letters separated by a single space. The first word is the name of the artist, and the second word is the name of the time slot they prefer to perform. The input ends with a line containing two zeros.

Output

Print the names of the artists for each test case, with one name per line, following the requested order. Do not leave any blank lines between the outputs for different test cases.

Sample Input

```
3 6
MORNING
AFTERNOON
EVENING
LUCY AFTERNOON
JOHN MORNING
BOB EVENING
ALICE MORNING
PETER MORNING
SARA AFTERNOON
1 4
ANYTIME
PETER ANYTIME
JOHN EVENING
ARTHUR MORNING
SAM EVENING
0 0
```

Sample Output

```
JOHN
ALICE
PETER
LUCY
SARA
BOB
PETER
JOHN
SAM
ARTHUR
```