CULINARY CLASH

Welcome to Culinary Clash, the most prestigious cooking tournament of the year! As the event coordinator, you face a critical logistical challenge. The venue has multiple specialized cooking stations, and each competing chef has expressed a preference for a particular station. Your task? Create an optimal assignment order following the tournament director's specific requirements.

You must arrange the chef roster based on their station preferences according to these guidelines:

- Chefs should be grouped by their preferred cooking stations
- When organizing chefs into groups based on their station preferences, you must follow the exact order of stations as they appear in the venue's official list - not alphabetically or by any other ordering system
- When multiple chefs request the same station, maintain their original registration order
- Some ambitious chefs may request specialty stations not on the venue's official list - these contestants should be placed at the end of your arrangement, grouped by their requested specialty stations
- The order of these specialty station groups should follow the order in which they first appear in the chef preference list
- Note that some stations might remain unassigned - this is acceptable as they'll be repurposed later.

Input

Your program will process multiple tournament scenarios. Each scenario begins with two integers:

- s ($1 \le s \le 1{,}000$): The number of standard cooking stations
- p ($1 \le p \le 100{,}000$): The number of participating chefs

The next s lines contain unique station identifiers - unique single word codes consisting of 1-30 uppercase letters.

Following this, p lines describe the chefs and their preferences. Each line contains two space-separated codes (1-30 uppercase letters each):

- First code: Chef identifier
- Second code: Preferred station identifier

A line with "0 0" signals the end of all test case scenarios.

Output

For each tournament scenario, print the chef identifiers in the required order, one per line. Do not separate outputs from different scenarios with blank lines.

Sample Input

```
3 7
PASTRY
GRILL
SAUCIER
```

```
DRAHOMIRA GRILL
SVETOZAR PASTRY
ZELMIRA SAUCIER
BOZENA PASTRY
JAROMIR PASTRY
VELESLAV GRILL
LUTOSLAV GRILL
1 4
STANDARD
ANTON STANDARD
TOBIAS MULTIFUNCTIONAL
ZUZANA VEGAN
MARTIN MULTIFUNCTIONAL
0 0
```

Sample Output

```
SVETOZAR
BOZENA
JAROMIR
DRAHOMIRA
VELESLAV
LUTOSLAV
ZELMIRA
ANTON
TOBIAS
MARTIN
ZUZANA
```