2024/2025 SOUTHERN CALIFORNIA REGIONAL
INTERNATIONAL COLLEGIATE PROGRAMMING CONTEST

Problem 7
N-Segment Display

Liquid Crystal Displays (LCDs) are popular on budget and single-purpose devices such as wristwatches, medical devices, and calculators. These devices display a limited number of characters, and compose their characters from a set of individual segments rather than pixels. Unfortunately, segments burn out over time, making the display ambiguous. Your team is to write a program that, given a set of characters and their segment definitions, checks readings for segments that might be burned out and determines the possible characters that are being displayed.

Input to your program has two sections: the characters and their segment definitions, followed by a number of readings. The first line of input contains N, the number of segments in a display, $1 \leq N \leq 26$. The second line contains C, the number of possible ASCII characters that can be displayed, $1 \leq C \leq 64$. The next C lines are the character definitions. Each character definition line starts with the character in column 1, then a sorted list of segment IDs that are turned on to display the character. The first display segment ID is labeled as *a*, the second as *b*, etc. For example, an 8-segment display uses segments *a* through *h*; a 25-segment display uses segments *a* through *y*. A character that has no segments turned on has no segment IDs, and will appear as the tilde '~' in column 1. Each character's segment definition is distinct from all other characters. A character will only be defined once.

The line following the character definitions contains K, the number of N-segment displays on the device, $1 \leq K \leq 7$. Displays are numbered from left to right. The next line contains R, the number of readings, $1 \leq R \leq 100$. The remaining R lines contain K comma-separated display readings, showing the segments that are on for that reading. It is possible that all segments in a given display are burned out. The device will never attempt to display undefined characters. Segments will either work (can be turned on or off), or be burned out (always off) across all readings for each independent display.

Figure 1 is a device with three 8-segment displays. The device is designed to show positive or negative whole numbers, the letters 'E' and 'R', and a blank character. The first sample input matches the device in Figure 1.

Given all the readings, determine all possible characters that each reading could represent. For each reading, print a line with all the possible characters for the first display, and as necessary print a space and all the possible characters for the second display, all the way up to the Kth display. For each display, sort the possible characters in ascending ASCII order.

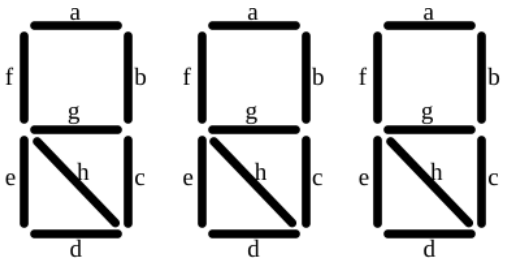

Figure 1. Three 8-segment displays, each with segments *a* through *h*.

Problem 7
N-Segment Display (continued)

Sample Input 1

```
8
14
0abcdef
1bc
2abdeg
3abcdg
4bcfg
5acdfg
6acdefg
7abc
8abcdefg
9abcdfg
-g
Eadefg
Rabefgh
~
3
4
adefg,abefg,abefgh
adefg,abcdef,bc
g,bcfg,abcdefg
,,bc
```

Output for Sample Input 1

```
68E R R
68E 0 1
- 4 8
1~ ~ 1
```

Sample Input 2

```
1
2
-a
~
1
2

a
```

Output for Sample Input 2

```
~
-
```