Alice and Bob are playing a board game. The board is divided into positions labeled a, b, c, d, $\ldots$ and the players use a gamepiece to mark the current position. Each round of the game consists of two steps:

1. Alice makes a choice. Depending on the current position, she has different options, where each option is a set of positions. Alice chooses one set S among the available sets of positions.
2. Bob makes a choice. His choice is one position p from the set S that Alice chose in step 1. Bob moves the gamepiece to position p, which is the position for the start of the next round.

Prior to the first round, each player independently selects one of the positions and reveals it at the start of the game. Bob's position is where the game starts. Alice wins the game if she can force Bob to move the gamepiece to the position she has chosen. To make things interesting, they have decided that Bob will pay Alice a certain amount if he loses, but Alice must pay Bob a certain amount after every round. The game now ends if Alice's position is reached or when Alice runs out of cash.

Both Alice and Bob play optimally: Alice will always choose an option that will lead to her winning the game, if this is possible, and Bob will always try to prevent Alice from winning.

For all possible start and end positions, Alice would like you to determine whether she can win the game and if so, how many rounds it will take.

Input

The input file contains several test cases, each of them as described below.

The first line contains the number of positions n ($1 \leq n \leq 25$). The n positions are labeled using the first n letters of the English alphabet in lowercase. The rest of the test case consists of n lines, one for each position p, in alphabetical order. The line for position p contains the options available to Alice in position p. It starts with the number of options m ($1 \leq m < 2^n$), which is followed by m distinct strings, one for each option. Each string contains the positions available to Bob if Alice chooses that option. The string has at least 1 character, the characters (which correspond to valid board positions) are in alphabetical order, and no characters are duplicated. The total number of options for the test case is at most 10^6.

Output

For each test case, the output must follow the description below.

For each position p in alphabetical order, display one line. In that line, for each position q in alphabetical order display the minimal number of rounds in which Alice can be guaranteed to arrive at position q when starting the game in position p, or '`-1`' if Alice cannot be guaranteed to reach q from p.

Sample Input

```
2
2 ab b
1 b
3
1 b
2 b a
2 ab ac
```

Sample Output

```
0 1
-1 0
0 1 -1
1 0 -1
2 2 0
```