6322 The Swapping Game

Last year you invented a game which can be played using a board and a die (singular of dice), this year you invented another new game which can be played using a single string of some letters.

The game starts with a string of N lower case English letters ('a' to 'z'), and you can only swap two different characters in this string, and you can make this step zero or more times. Your goal is to reach the lexicographically smallest string after doing zero or more moves.

But there are some constraints on the final string. For each position, it must be a letter of some given letters (the given letters are not necessary the same for each position). For example, the first letter must be 'a' or 'b', the second letter must be 'b' or 'c', and so on.

Note that these constraints are on the final string only, which means you can make moves which cause invalid strings to reach a valid string after some more moves.

Given the initial string and the constraints on each position, your task is to write a program to find the lexicographically smallest valid string after making zero or more moves.

Note: When comparing two different strings of the same length, the lexicographically smaller one is the one with a smaller letter on the first place where they differ.

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cases ($1 \leq T \leq 100$). Followed by the test cases, each test case starts with a line containing the initial string S which consists of N lower case English letters ($1 \leq N \leq 100$). Followed by N lines each line contains a string C_i which consists of L_i distinct lower case English letters ($1 \leq L_i \leq 5$) which are the valid letters for the i-th position in the final string. Each letter in each C_i will appear at least once in S.

Output

For each test case, print on a single line the lexicographically smallest valid string you can get after zero or more moves. If there is no such valid string print 'NO SOLUTION' (without the quotes).

Sample Input

```
2
abcde
abcde
a
abcde
abcde
abcde
abcde
ab
ab
ab
abcde
abcde
```

Sample Output

```
bacde
NO SOLUTION
```