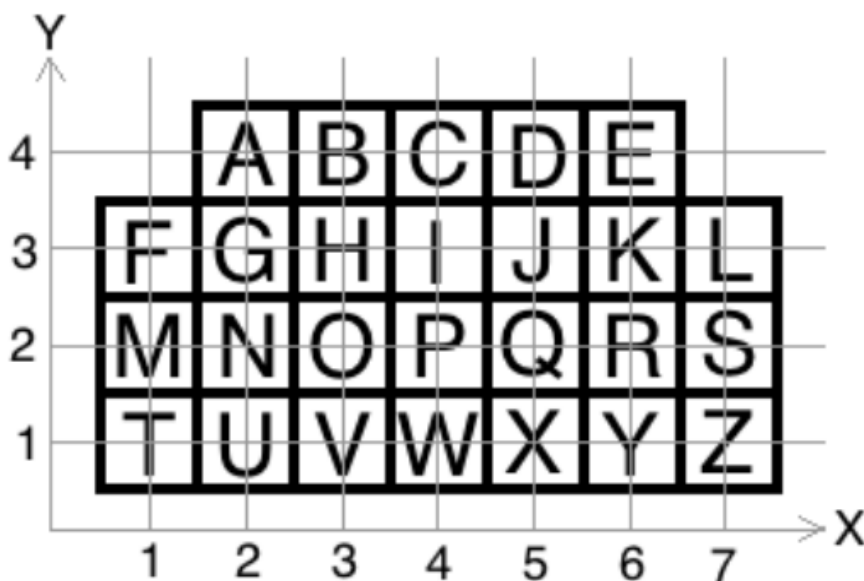


6755 Swyper Keyboard

Swyper keyboard is a new kind of keyboards for touch screen mobile phones, where you just draw a sequence of one or more connected segments (poly-line) instead of tapping every letter of the word you want to type. The keyboard layout which we will deal with in this problem will look like the following one (the width and height of each square is 1 unit, and the center of each square is always a point with integer coordinates):



Please note that the keyboard which is described in this problem does not behave exactly like the real popular keyboard, so do not make any assumption which is not mentioned in the problem statement.

A poly-line in this problem will be represented using a string of 2 or more upper case English letters, where every 2 consecutive letters are always different. For example the string "ACM" represents a poly-line which starts from the center of the letter 'A' then goes to the center of the letter 'C' then goes to the center of the letter 'M'. But this poly-line touches some other letters, between 'A' and 'C' it touches 'B' and between 'C' and 'M' it touches 'B' then 'H' then 'G' then 'N' (in that order). Given the initial string which represents a poly-line, we can get another string of all the letters which that poly-line touches in the correct order, which is referred to as the expansion of the poly-line. In this example this string is "ABCBHGNM". When you draw this poly-line (which was initially represented by the string "ACM"), you might mean any word which is a subsequence of the expansion string of the poly-line (this string is "ABCBHGNM"), you might mean "ACM", "BGN", "ABCBHGNM" or any other subsequence.

A string X is considered a subsequence of another string Y , when X can be obtained by deleting zero or more letters from the string Y without changing the order of the remaining letters.

You are given a dictionary containing one or more words and a string which is the initial representation for a poly-line, your task is to find a word in the dictionary which might be meant by that poly-line as described above.

Note that a poly-line which touches only the boundaries of a square does not count as a touch, for example a segment going from 'A' to 'H' does not touch 'B' or 'G'.

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, the first line of each test case contains an integer and a string separated by a single space $N P$ ($1 \leq N \leq 1,000$) representing the number of words in the dictionary and the initial representation of the poly-line, respectively. Followed by N lines, each line contains a single string which represents a word in the dictionary. All the strings in the input are non-empty strings of at least 2 and at most 100 upper case English letters (from ‘A’ to ‘Z’).

Note that the given string for the poly-line is just to describe its segments, and it might not include all the letters which the poly-line touches. Also, one poly-line might touch the same letter more than once, even though consecutive letters of a poly-line can not be the same.

Output

For each test case, print a single line which should be ‘NO SOLUTION’ if there is no word in the dictionary which is a subsequence of the expansion of the given poly-line. Otherwise you should print the first word (according to the input’s order) which is a subsequence of the expansion of the given poly-line.

Sample Input

```
2
3 ACM
ACPC
AM
ACM
2 ACM
XY
ZW
```

Sample Output

```
AM
NO SOLUTION
```