ACM-ICPC Live Archive

5768 Isabella's Message

Isabella and Steve are very good friends, and they often write letters to each other. they exchange funny experiences, talk about people around, share their feelings and write about almost everything through the letters. When the letters are delivered, they are quite afraid that some other people (maybe their parents) would peek. So they encrypted the letter, and only they know how to decrypt it. this guarantees their privacy.

The encrypted message is an $N \times N$ matrix, and each grid contains a character. Steve uses a special mask to work as a key. the mask is $N \times N$ (where N is an even number) matrix with $\frac{N \times N}{4}$ holes of size 1×1 on it.

The decrypt process consist of the following steps:

1. Put the mask on the encrypted message matrix
2. Write down the characters you can see through the holes, from top to down, then from left to right.
3. Rotate the mask by 90 degrees clockwise.
4. Go to step 2, unless you have wrote down all the $N \times N$ characters in the message matrix.
5. Erase all the redundant white spaces in the message.

For example, you got a message shown in figure 1, and you have a mask looks like figure 2. The decryption process is shown in figure 3, and finally you may get a message "`good morning`".

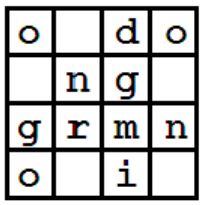


figure.1

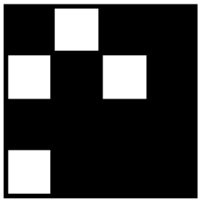
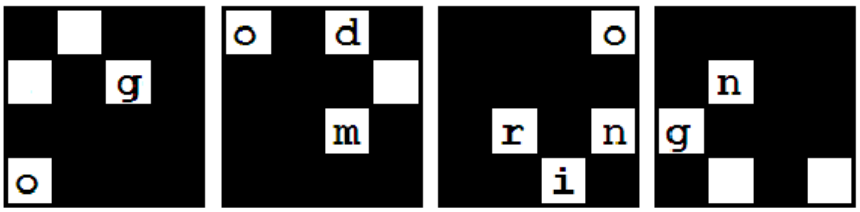


figure.2



" go" "od m" "orni" "ng "

figure.3

You can assume that the mask is always carefully chosen that each character in the encrypted message will appear exactly once during decryption.

However, in the first step of decryption, there are several ways to put the mask on the message matrix, because the mask can be rotated (but not flipped). So you may get different results such as "`od morning go`" (as showed in figure 4), and you may also get other messages like "`orning good m`", "`ng good morni`".

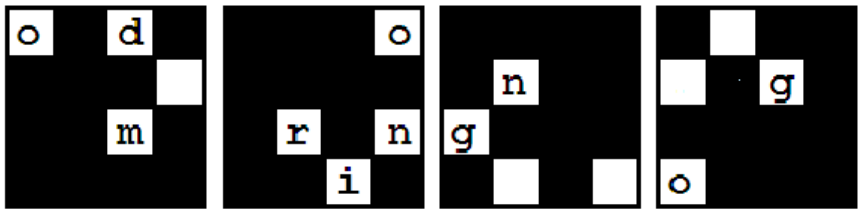


figure.4

Steve didn't know which direction of the mask should be chosen at the beginning, but after he tried all possibilities, he found that the message "good morning" is the only one he wanted because he couldn't recognize some words in the other messages. So he will always consider the message he can understand the correct one. Whether he can understand a message depends whether he knows all the words in the message. If there are more than one ways to decrypt the message into an understandable one, he will choose the lexicographically smallest one. the way to compare two messages is to compare the words of two messages one by one, and the first pair of different words in the two messages will determine the lexicographic order of them.

Isabella sends letters to Steve almost every day. As decrypting Isabella's message takes a lot of time, and Steve can wait no longer to know the content of the message, he asked you for help. Now you are given the message he received, the mask, and the list of words he already knew, can you write a program to help him decrypt it?

Input

The first line contains an integer T ($1 \leq T \leq 100$), indicating the number of test cases.

Each test case contains several lines.

The first line contains an even integer N ($2 \leq N \leq 50$), indicating the size of the matrix.

The following N lines each contains exactly N characters, reresenting the message matrix. The message only contains lowercase letters and periods('`.`'), where periods represent the white spaces. You can assume the matrix contains at least one letter.

The following N lines each contains N characters, representing the mask matrix. the asterisk('`*`') represents a hole, and period('`.`') otherwise. The next line contains an integer M ($1 \leq M \leq 100$), the number of words he knew.

Then the following M lines each contains a string represents a word. The words only contain lowercase letters, and its length will not exceed 20.

Output

For each test case in the input, print one line: '`Case #`X`:` Y', where X is the test case number (starting with 1) and Y is Isabella's message. If Steve cannot understand the message, just print the Y as '`FAIL TO DECRYPT`'.

Sample Input

```
3
4
o.do
.ng.
grmn
o.i.
.*..
*.*.
```

```
....
*...
2
good
morning
4
..lf
eoyv
oeou
vrer
..*.
.*..
....
*.*.
5
i
you
the
love
forever
4
.sle
s.c.
e.fs
..uu
*...
.*..
...*
..*.
1
successful
```

Sample Output

```
Case #1: good morning
Case #2: love you forever
Case #3: FAIL TO DECRYPT
```