

B: Be Strong

Source file name: `be.c`, `be.cpp`, `be.java`, or `be.py`

Author: Camilo Rocha

A *prefix* of a string is a substring that occurs at the beginning. If a string has size $N \geq 0$, then it has exactly $N + 1$ prefixes (the empty string is a prefix of any string). Given a collection of strings, its *strong prefix* is the longest prefix common to all strings in the collection. Any difference between lowercase and uppercase characters is considered immaterial.

For example, `te` is the strong prefix for the collection

`tequila` `tEnnessee` `Telephone` `tetris`

The empty string is the strong prefix for the collection

`hello` `world`

In this problem, you are asked to compute the strong prefix of a collection of words.

Input

The input consists of several test cases. Each test case begins with a line containing a number M ($0 \leq M \leq 5\,000$) denoting the number for words in the collection. Then, M lines follow, each containing a string W ($1 \leq |W| \leq 200$) made from English lowercase and uppercase characters. The end of the input is given with $M = 0$, which should not be processed.

The input must be read from standard input.

Output

For each test case, output a single line with the strong prefix of the collection of M words where the case is immaterial. The output can only contain English lowercase characters. If the strong prefix is empty, output a star `*`.

The output must be written to standard output.

Sample Input	Sample Output
4 tequila tEnnessee Telephone tetris	te * de
2 hello world	
2 De dE	
0	