

Problem F

Multidimensional Hangman

The Multidimensional Hangman Game has very peculiar rules. In a way, it is like you are playing several games of the traditional Hangman game at the same time, with the difference that the words don't have to exist in the dictionary. If you've never played Hangman, don't worry: all the information you need is below.

In the multidimensional version of the game, there are several words on a board, initially unknown, all of the same length. At each turn in the game, you discover some characters from certain word positions (how these characters were discovered is not important for this problem). At a certain point, when only one unknown character remains in each word on the board, the game goes into the *all or nothing* phase. At this point, you must choose a word that maximizes the number of compatibilities with the words on the board. For a chosen word P , we say it is compatible with a word T on the board if all known letters in T occur in exactly the same positions in P .

Given the known information about the words on the board, you must determine which word to choose for *all or nothing* phase, which maximizes the number of compatibilities. If there is more than one solution, print the lexicographically smallest. We say that a word P is lexicographically smaller than a word Q if $P_i < Q_i$ where P_i is the i -th character of P , Q_i is the i -th character of Q and i is the smallest index such that $P_i \neq Q_i$.

Input

The first line of the input contains two integers N and C satisfying $1 \leq N \leq 10^4$ and $1 \leq C \leq 12$, indicating the number of words on the board and the length of the words it contains. Each of the next N lines contains a word of length C composed only of characters from “a” to “z” except for one of its positions, which will contain a character “*” , indicating that the character at that position is still unknown.

Output

Print a single line, containing, in order, a word T , of length C , and an integer M , such that M is the greatest number of compatibilities a word might have with the input words and T is the lexicographically smallest amongst the words with compatibility M .

Input example 1 5 4 rat* ru*d rot* r*ta r*ta	Output example 1 rata 3
Input example 2 5 4 bon* fon* n*no *eto *ano	Output example 2 nano 2