Problem A: Says Gone By

It is known that languages evolve in much the same way as biological species – that is, a language may give rise to two or more daughter languages, which may in turn give rise to daughter languages, and so on. As words are inherited from parent languages and change independently in each descendent language, a word in the original ancestral language may appear in each of its descendent languages but in mutated form. To reconstruct such an ancestral word from the mutated forms in the descendent languages, the common practice in linguistics is to (1) align these mutated words such that all letters line up and (2) for each word-position i, reconstruct the character in that position in the ancestral word as the character that occurs in that position in $> 50\%$ of the mutated forms, if such a character exists, and as character ``*'' otherwise. For example, given the set of mutated words {``apple'', ``apfel'', ``apfol''}, the ancestral word reconstruction would be ``apf*l''.

Write a program which, given several groups of mutated words, computes and outputs the reconstructed ancestral word for each group. Your input will be a textfile in which the first line is the number of groups of mutated words to analyze. The first line of each group that follows gives the number of words, n, in the mutation and the length, l, of the mutated words (all the words within each group have the same length). Each of the subsequent n lines contains a single mutated word within the group. All the subsequent groups of words are structured similarly. You may assume that all input files are formatted correctly.

Sample input (available as file "A.in"):

```
3
3 5
apple
apfel
apfol
6 5
apple
apfel
apfil
opfol
ipvul
upfol
8 7
worldly
earthly
candide
coppert
sindert
analoge
zymrgys
kintles
```

Sample output (available as file "A.out"):

```
apf*l
*pf*l
*******
```