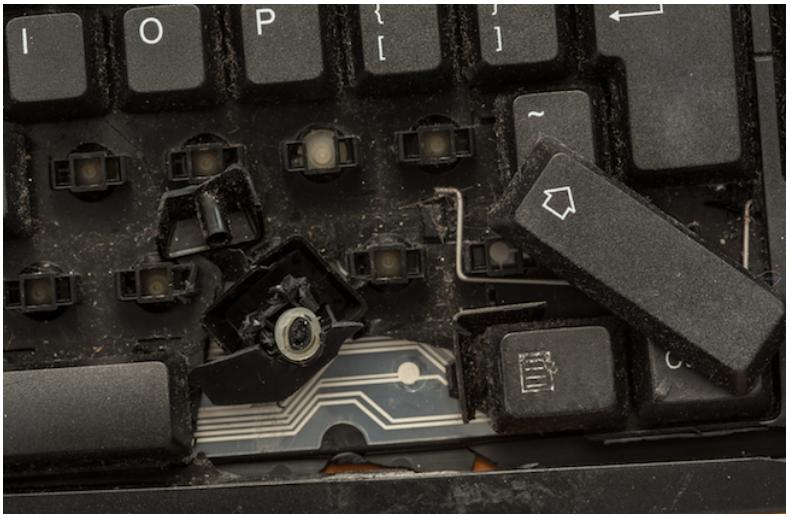

Goal

Recently, your keyboard began to suffer from an unfortunate defect: it happens that key presses are not recorded. As a result, you who have been in the habit of banging fast without reading again, you now see some missing letters in your texts.

Fortunately, most of the time, the context makes it possible to find what word was supposed to be there. So you wonder what ambiguities are likely to appear. Given a list of words, you are trying to find out if it is possible to confuse them all by removing letters in the middle.

The objective of the challenge is to find a series of letters which are found, in the same order, in all words of the input list. We would like this sequence to be as long as possible, in other words, to obtain it from the given words implies the least possible deletions of letters. (For simplicity, in this exercise, the words in the list will all be 10.)

Indication: you can proceed by exhaustive enumeration (brute force).

Data

Input

Row *1*: an integer **N** between *1* and *100*, representing number of words.
Row *2* to ***N*** *+ 1*: on each line, a string of 10 characters made up of only 26 letters of the lowercase alphabet.

Output

A sequence of characters of maximum length that could be obtained from the input words by deleting characters.

Example

With the following input:
```
3
artificiel
algorithme
algebrique
```

All 3 words above can give `arie` by removing 6 characters. By deleting 5 characters or less, we could not get a sequence of letters common to all three words. It is therefore `arie` that must be displayed.

If the input had been:
```
2
artificiel
algorithme
```

Then the `arie` and `arte` answers would both have been accepted.

Finally, on the following input:
```
2
algorithme
algebrique
```

The expected answer is `algrie`.

You can download sample input and output data files to work locally by clicking on the link at the bottom of the French version of the question.

Téléchargez des fichiers d'exemple ainsi qu'un modèle de code pour travailler localement.