

Alien ABC (alienabc)

An alien race decided to create a new alphabet (*alienabc*) based on the English alphabet. For each letter in the English alphabet, they can:

- Exclude the letter entirely.
- Include the letter (e.g., $a \rightarrow a$).
- Include the letter in a doubled form (e.g., $a \rightarrow aa$) (aa is one letter in their new alphabet).
- Include both the single and double versions of the letter (e.g., $a \rightarrow a \ aa$).



Figure 1: An ancient alien writing?

They wrote down the alienabc, each alien letter exactly once, in their own, alien alphabetic order. The order of letters might differ from the English alphabetic order, and a letter and its doubled version (e.g., a and aa) do not have to be adjacent in the alienabc. Unfortunately, they did this *without* any separator.

Can you reconstruct the alienabc? You only have to insert spaces between the alien 'letters'. For example, if they wrote $adccb$, then the alienabc is clearly $a \ d \ cc \ b$. The reconstruction might not be unique, for example aaa can be coming from $a \ aa$ or $aa \ a$ as well. In such case you can choose any of them. Finally, it can happen that they made some mistake and the given string does not correspond to a valid alienabc. For example $aabaa$ could come from $aa \ b \ aa$, but in an alienabc each 'letter' must appear at most once. In such cases you should output -1 .

📎 Among the attachments of this task you may find a template file `alienabc.*` with a sample incomplete implementation.

Input

Each test contains multiple test cases. The first line of input contains a single integer T , the number of test cases.

The only line of each test case contains a string S representing a (not necessarily valid) alienabc without spaces.

Output








For each test case, if the input string corresponds to an alienabc: print the original space separated alienabc (in case there are more solutions: output any of them), otherwise print `-1`.

Constraints

- $1 \leq T \leq 1000$.
- The length of S is between 1 and 100 (inclusive), and it contains only lowercase English letters.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- | | |
|---|---|
| – Subtask 1 (0 points) | Examples. |
|  | |
| – Subtask 2 (7 points) | The length of S is at most 10 and contains at most the first 2 letters of the English alphabet (ie. <code>a</code> or <code>b</code>). All input strings correspond to a valid alienabc. |
|  | |
| – Subtask 3 (11 points) | The length of S is at most 10 and contains at most the first 2 letters of the English alphabet. |
|  | |
| – Subtask 4 (13 points) | The length of S is at most 20 and contains at most 5 different letters of the English alphabet. All the input strings correspond to a valid alienabc. |
|  | |
| – Subtask 5 (17 points) | The length of S is at most 20 and contains at most 5 different letters of the English alphabet. |
|  | |
| – Subtask 6 (23 points) | Each input string corresponds to a valid alienabc. |
|  | |
| – Subtask 7 (29 points) | No additional limitations. |
|  | |

Examples

input	output
3 abcd eeezzoppoo aabacccbbbd	a b c d e ee zz o pp oo -1

Explanation

In the **first test-case** there is exactly one valid alienabc: `a b c d`.

In the **second test-case** there are two solutions: you can print either `e ee zz o pp oo` or `ee e zz o pp oo`.

In the **third test-case** there is no valid alienabc for the given string (we have too many b-s).