You have an array of logs.  Each log is a space delimited string of words.

For each log, the first word in each log is an alphanumeric *identifier*.  Then, either:

* Each word after the identifier will consist only of lowercase letters, or;
* Each word after the identifier will consist only of digits.

We will call these two varieties of logs *letter-logs* and *digit-logs*.  It is guaranteed that each log has at least one word after its identifier.

Reorder the logs so that all of the letter-logs come before any digit-log.  The letter-logs are ordered lexicographically ignoring identifier, with the identifier used in case of ties.  The digit-logs should be put in their original order.

Return the final order of the logs.

**Example 1:**

**Input:** logs = ["dig1 8 1 5 1","let1 art can","dig2 3 6","let2 own kit dig","let3 art zero"]

**Output:** ["let1 art can","let3 art zero","let2 own kit dig","dig1 8 1 5 1","dig2 3 6"]

**Constraints:**

1. 0 <= logs.length <= 100
2. 3 <= logs[i].length <= 100
3. logs[i] is guaranteed to have an identifier, and a word after the identifier.