Given a list of phrases, generate a list of Before and After puzzles.

A *phrase* is a string that consists of lowercase English letters and spaces only. No space appears in the start or the end of a phrase. There are no consecutive spaces in a phrase.

*Before and After puzzles* are phrases that are formed by merging two phrases where the **last word of the first phrase** is the same as the **first word of the second phrase**.

Return the Before and After puzzles that can be formed by every two phrases phrases[i] and phrases[j] where i != j. Note that the order of matching two phrases matters, we want to consider both orders.

You should return a list of **distinct** strings **sorted lexicographically**.

**Example 1:**

Input: phrases = ["writing code","code rocks"]  
Output: ["writing code rocks"]

**Example 2:**

Input: phrases = ["mission statement",  
 "a quick bite to eat",  
  "a chip off the old block",  
  "chocolate bar",  
  "mission impossible",  
  "a man on a mission",  
  "block party",  
  "eat my words",  
  "bar of soap"]  
Output: ["a chip off the old block party",  
  "a man on a mission impossible",  
  "a man on a mission statement",  
  "a quick bite to eat my words",  
 "chocolate bar of soap"]

**Example 3:**

Input: phrases = ["a","b","a"]  
Output: ["a"]

**Constraints:**

* 1 <= phrases.length <= 100
* 1 <= phrases[i].length <= 100