import java.awt.\*;

import java.util.\*;

import java.util.List;

/\*

https://leetcode.com/problems/reorder-data-in-log-files/

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:

0 <= logs.length <= 100

3 <= logs[i].length <= 100

logs[i] is guaranteed to have an identifier, and a word after the identifier.

\*/

**public class ReorderDatainLogFiles {**

**public String[] reorderLogFiles(String[] logs) {**

**if(logs == null || logs.length == 0)**

**return new String[0];**

**HashMap<String,String> stringMap = new HashMap<String, String>();**

**List<String> digitsList = new ArrayList<String>();**

**List<String> letterList = new ArrayList<String>();**

**for(int i=0;i<logs.length;i++) {**

**int spaceIndex = logs[i].indexOf(" ");**

**String secondPart = logs[i].substring(spaceIndex + 1);**

**char[] chars = secondPart.toCharArray();**

**if (Character.isDigit(chars[0])) {**

**digitsList.add(logs[i]);**

**} else {**

**// stringMap.put(secondPart, logs[i]);**

**letterList.add(logs[i]);**

**}**

**}**

**Collections.sort(letterList, new Comparator<String>() {**

**@Override**

**public int compare(String s1, String s2) {**

**String newS1 = s1.substring(s1.indexOf(" ")+1);**

**String newS2 = s2.substring(s2.indexOf(" ")+1);**

**return newS1.equals(newS2) ? s1.compareTo(s2) : newS1.compareTo(newS2);**

**}**

**});**

**List<String> result = new ArrayList<String>();**

**for(Iterator iter1 = letterList.iterator();iter1.hasNext();)**

**{**

**result.add(iter1.next().toString());**

**}**

**for(Iterator iter2 = digitsList.iterator();iter2.hasNext();)**

**{**

**result.add(iter2.next().toString());**

**}**

**String[] resultArray = new String[result.size()];**

**return result.toArray(resultArray);**

**}**

**}**