# Question

Given a string, we can "shift" each of its letter to its successive letter, for example: "abc" -> "bcd". We can keep "shifting" which forms the sequence:

"abc" -> "bcd" -> ... -> "xyz"

Given a list of **non-empty** strings which contains only lowercase alphabets, group all strings that belong to the same shifting sequence.

**Example:**

**Input:** ["abc", "bcd", "acef", "xyz", "az", "ba", "a", "z"],

**Output:**

[

["abc","bcd","xyz"],

["az","ba"],

["acef"],

["a","z"]

]

# Solution

|  |
| --- |
| public class Solution {  public List<List<String>> groupStrings(String[] strings) {  List<List<String>> result = new ArrayList<List<String>>();  Map<String, List<String>> map = new HashMap<String, List<String>>();  for (String str : strings) {  int offset = str.charAt(0) - 'a';  String key = "";  for (int i = 0; i < str.length(); i++) {  char c = (char) (str.charAt(i) - offset);  if (c < 'a') {  c += 26;  }  key += c;  }  if (!map.containsKey(key)) {  List<String> list = new ArrayList<String>();  map.put(key, list);  }  map.get(key).add(str);  }  for (String key : map.keySet()) {  List<String> list = map.get(key);  Collections.sort(list);  result.add(list);  }  return result;  }  } |