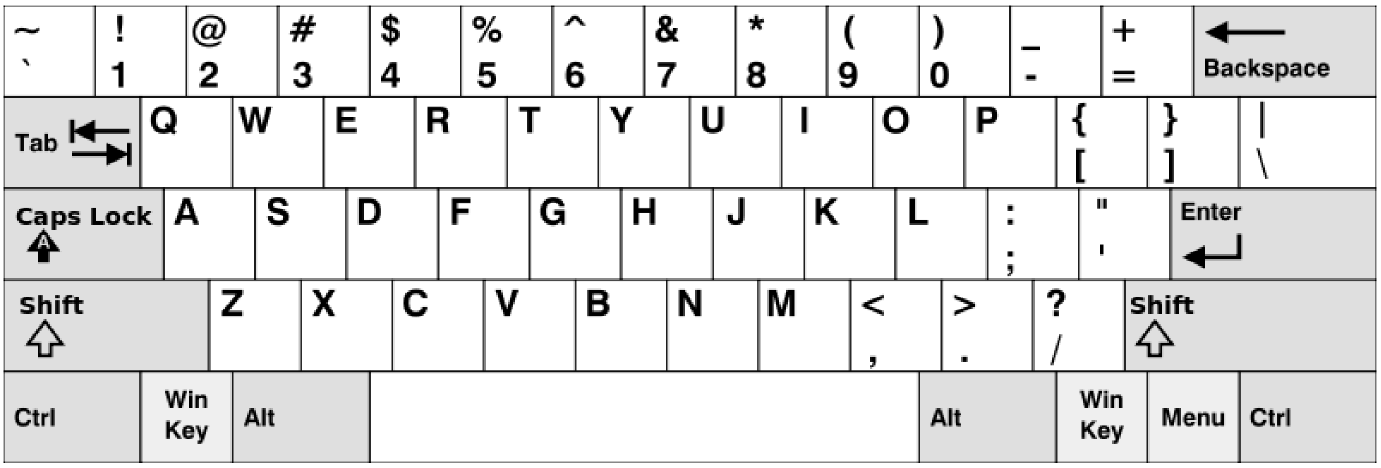
Keyboard Row

Given a List of words, return the words that can be typed using letters of**alphabet** on only one row's of American keyboard like the image below.



Example 1:

```
Input: ["Hello", "Alaska", "Dad", "Peace"]
Output: ["Alaska", "Dad"]
```

Note:

1. You may use one character in the keyboard more than once.
2. You may assume the input string will only contain letters of alphabet.

Solution 1

```
public String[] findWords(String[] words) {
    return Stream.of(words).filter(s -> s.toLowerCase().matches("[qwertyuiop]*|[a
sdfghjkl]*|[zxcvbnm]*")).toArray(String[]::new);
}
```

written by lixx2100 original link here

Solution 2

```java
public class Solution {
    public String[] findWords(String[] words) {
        String[] strs = {"QWERTYUIOP","ASDFGHJKL","ZXCVBNM"};
        Map<Character, Integer> map = new HashMap<>();
        for(int i = 0; i<strs.length; i++){
            for(char c: strs[i].toCharArray()){
                map.put(c, i);//put <char, rowIndex> pair into the map
            }
        }
        List<String> res = new LinkedList<>();
        for(String w: words){
            if(w.equals("")) continue;
            int index = map.get(w.toUpperCase().charAt(0));
            for(char c: w.toUpperCase().toCharArray()){
                if(map.get(c)!=index){
                    index = -1; //don't need a boolean flag.
                    break;
                }
            }
            if(index!=-1) res.add(w);//if index != -1, this is a valid string
        }
        return res.toArray(new String[0]);
    }
}
```

written by Chidong original link here

Solution 3

```java
public String[] findWords(String[] words) {
 String[] base = {"qwertyuiop","asdfghjkl","zxcvbnm"};
 List<String> list = new ArrayList<String>();
 for (String string : words) {
  for (String basStr : base) {
   boolean find = true;
   for (char c : string.toCharArray()) {
    String low = String.valueOf(c).toLowerCase();
    if (!basStr.contains(low)){
     find = false;
     break;
    }
   }
   if (find) list.add(string);
  }
 }
 String[] res = new String[list.size()];
 for (int i = 0; i < res.length; i++) res[i] = list.get(i);
 return res;
}
```

written by shawloatrchen original link here