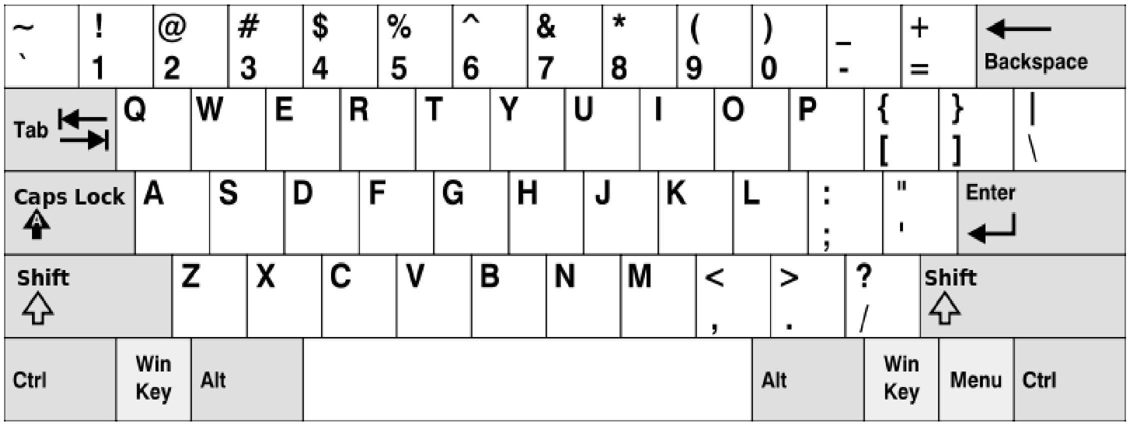
Keyboard Row (LeetCode500 – Easy)

Problem Description

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.


~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Backspace
Tab Q W E R T Y U I O P { [}] | \
Caps Lock A S D F G H J K L : ; " ' Enter
Shift Z X C V B N M < , > . ? / Shift
Ctrl Win Key Alt Alt Win Key Menu Ctrl

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

Output: ["Alaska", "Dad"]

Problem Analysis

- Define a final String array to store keyboard rows.
- Use HashMap to store <character, row number> pairs.
- For each word, use the first character to get the corresponding row number.
- Search through the word character by character.
    - If the corresponding row number doesn't change, keep searching next character.
    - If got a different row number, break to search next word.
    - After iterating through all the characters in current word, if the corresponding row number doesn't change, save current word.
- Notice: convert to Upper Case before searching.

Solution

```java
/**
 * Given a List of words,
 * return the words that can be typed,
 * using letters of alphabet
 * on only one row's of American keyboard
 * @param words
 */
public String[] findWords(String[] words) {

        //Initialize
        final String[] keyboard = {"QWERTYUIOP", "ASDFGHJKL", "ZXCVBNM"};
        HashMap<Character, Integer> keyboardMap = new HashMap<Character, Integer>();
        for (int i = 0; i < keyboard.length; i++) {
                char[] keyboardRow = keyboard[i].toCharArray();
                for (char c : keyboardRow) {
                        keyboardMap.put(c, i);
                }
        }
        ArrayList<String> list = new ArrayList<String>();
        //Search
        for (String word : words) {
                char[] wordInChar = word.toCharArray();
                int rowNumber = keyboardMap.get(Character.toUpperCase(wordInChar[0]));
                for (char c : wordInChar) {
                        if (keyboardMap.get(Character.toUpperCase(c)) != rowNumber) {
                                rowNumber = -1;
                                break;
                        }
                }
                if (rowNumber != -1) {
                        list.add(word);
                }
        }
        //Return
        return list.toArray(new String[0]);
}
```

Tips: how to dynamically add String objects to a String array

Since array must be declared with fixed size, we cannot actually add objects to an array dynamically. To achieve this goal, we should use another data structure that support dynamical add() operation. Take ArrayList as an example.

```java
ArrayList<String> list = new ArrayList<String>();
list.add(new String("string1"));
list.add(new String("string2"));
//...keep adding.
return list.toArray(new String[0]);
```