

## [1044 Find Common Characters \(link\)](#)

### Description

Given a string array `words`, return *an array of all characters that show up in all strings within the words (including duplicates)*. You may return the answer in **any order**.

#### Example 1:

```
Input: words = ["bella","label","roller"]  
Output: ["e","l","l"]
```

#### Example 2:

```
Input: words = ["cool","lock","cook"]  
Output: ["c","o"]
```

#### Constraints:

- $1 \leq \text{words.length} \leq 100$
- $1 \leq \text{words}[i].\text{length} \leq 100$
- `words[i]` consists of lowercase English letters.

(scroll down for solution)

# Solution

Language: *cpp*

Status: Accepted

```
#include <vector>
#include <string>
#include <algorithm>

class Solution {
public:
    std::vector<std::string> commonChars(std::vector<std::string>& words) {
        // Инициализируем массив minFreq для подсчета минимальной частоты каждого символа
        int minFreq[26] = {0}; // Поскольку только строчные английские буквы

        // Инициализируем minFreq большим числом (в данном случае 100, что больше макс
        for (int i = 0; i < 26; ++i) {
            minFreq[i] = 100;
        }

        // Обновляем minFreq с реальными частотами из каждого слова
        for (const std::string& word : words) {
            int freq[26] = {0}; // Частота для текущего слова

            // Считаем частоту каждого символа в текущем слове
            for (char c : word) {
                ++freq[c - 'a'];
            }

            // Обновляем minFreq минимальными частотами для каждого символа
            for (int i = 0; i < 26; ++i) {
                minFreq[i] = std::min(minFreq[i], freq[i]);
            }
        }

        // Создаем результат на основе minFreq
        std::vector<std::string> result;
        for (int i = 0; i < 26; ++i) {
            // Добавляем символ i + 'a' в результат minFreq[i] раз
            for (int j = 0; j < minFreq[i]; ++j) {
                result.push_back(std::string(1, i + 'a'));
            }
        }

        return result;
    }
};
```