# 题目

给你一个字符串数组words和一个字符串pref。

返回words中以pref作为前缀的字符串的数目。

字符串s的前缀就是s的任一前导连续字符串。

示例 1：

输入：words = ["pay","attention","practice","attend"], pref = "at"

输出：2

解释：以 "at" 作为前缀的字符串有两个，分别是："attention" 和 "attend" 。

示例 2：

输入：words = ["leetcode","win","loops","success"], pref = "code"

输出：0

解释：不存在以 "code" 作为前缀的字符串。

提示：

1 <= words.length <= 100

1 <= words[i].length, pref.length <= 100

words[i]和pref由小写英文字母组成

# 分析

## 方法一：模拟

代码：

class Solution {

public:

int prefixCount(vector<string>& words, string pref) {

int res = 0;

for (auto &word : words) {

if (word.compare(0, pref.size(), pref) == 0) {

res++;

}

}

return res;

}

};

复杂度分析

时间复杂度：O(n×m)，其中n是输入words的长度，m是输入pref的长度。

空间复杂度：O(1)，仅需要常数空间。