C++-数组字符串哈希表-有一个字符串数组 words 和一个字符串 chars
题目描述：
有一个字符串数组 words 和一个字符串 chars。
假如可以用 chars 中的字母拼写出 words 中的某个“单词”（字符串），那么我们就认为你掌握了这个单词。
words 的字符仅由 a-z 英文小写字母组成。 例如: abc
chars 由 a-z 英文小写字母和 “?”组成。其中英文问号“?”表示万能字符，能够在拼写时当做任意一个英文字母。 例如： "?" 可以当做 "a"等字母。
注意：每次拼写时，chars 中的每个字母和万能字符都只能使用一次。
输出词汇表 words 中你掌握的所有单词的个数。 没有掌握任何单词，则输出 0。
输入描述：
第 1 行输入数组 words 的个数，记为 N。
从第 2 行开始到第 N+1 行依次输入数组 words 的每个字符串元素。
第 N+2 行输入字符串 chars。
输出描述：
输出一个整数，表示词汇表 words 中你掌握的单词个数。
补充说明：
注意：
1 <= words.length <= 100
1 <= words[i].length, chars.length <= 100
所有字符串中都仅包含小写英文字母、英文问号
收起
示例 1
输入：
4
cat
bt
hat
tree
atach??
输出：
3
说明：
可以拼写字符串"cat"、"bt"和"hat"
示例 2
输入：
3
hello
world
cloud
welldonehoneyr
输出：
2
说明：

可以拼写字符串"hello"和"world"
示例 3
输入：
3
apple
car
window
welldoneapplec?
输出：
2
说明：
可以拼写字符串"apple"和"car"

```
#include <iostream>
#include <unordered_map>
#include <string>
#include <vector>
using namespace std;

int main() {
    int n;
    cin >> n;
    cin.ignore();
    vector<string> words(n);
    for (int i=0; i<n; i++){
        getline(cin, words[i]);
    }
    string chars;
    cin >> chars;
    unordered_map<char, int> letter_count;
    for (char ch : chars) {
        letter_count[ch]++;
    }
    int wenhao_count = (letter_count.find('?') != letter_count.end()) ? letter_count['?'] : 0;
    int memorized_word_count = 0;
    for (const string& word : words) {
        unordered_map<char, int> word_letter_count;
        int wenhao_required = 0;
        for (auto ch : word) {
            word_letter_count[ch]++;
        }
        for (const auto& p : word_letter_count) {
            char ch = p.first;
            int num = p.second;
            if (letter_count.find(ch) == letter_count.end()) {
```

```
                wenhao_required += num;
            }
            else {
                wenhao_required += std::max(0, num - letter_count[ch]);
            }
        }
        if (wenhao_required <= wenhao_count) memorized_word_count++;
    }
    cout << memorized_word_count;
    return 0;
}
```