

计算最大乘积

题目描述：

给定一个元素类型为小写字母的数组，请计算两个没有相同字符的元素 长度乘积的最大值，如果没有符合条件的两个元素，返回 0。

输入描述：

输入为一个半角逗号分隔的小写字母的数组， $2 \leq \text{数组长度} \leq 100$ ， $0 < \text{字符串长度} \leq 50$ 。

输出描述：

两个没有相同字符的元素 长度乘积的最大值。

示例 1

输入：

iwdvpbn,hk,iuop,iikd,kadgpf

输出：

14

说明：

数组中有 5 个元素。

iwdvpbn 与 hk 无相同的字符，满足条件，iwdvpbn 的长度为 7，hk 的长度为 2，乘积为 14（7*2）。

iwdvpbn 与 iuop、iikd、kadgpf 均有相同的字符，不满足条件。

iuop 与 iikd、kadgpf 均有相同的字符，不满足条件。

iikd 与 kadgpf 有相同的字符，不满足条件。

因此，输出为 14。

//两个没有相同字符的字符串的最大长度乘积

//分割字符串 vector<string>

//统计每个元素的字符出现频率

```
#include <iostream>
```

```
#include <string>
```

```
#include <vector>
```

```
using namespace std;
```

```
void parse(string &s, vector<string> &words) {
```

```
    int sz = (int)s.size();
```

```
    for (int i = 0; i < sz; ++i) {
```

```
        int j = i + 1;
```

```
        while (j < sz && s[j] != ',') ++j;
```

```
        words.push_back(s.substr(i, j - i));
```

```
        i = j;
```

```
    }
```

```
}
```

```
void staticFreq(vector<string> &words, vector<vector<int>> &freq) {
```

```
    int sz = (int)words.size();
```

```
    for (int i = 0; i < sz; ++i) {
```

```
        for (int j = 0; j < words[i].size(); ++j) {
```

```
            freq[i][words[i][j] - 'a']++;
```

```
        }
```

```
}
```

```
}
```

```
bool check(vector<int> &f, vector<int> &f2) {
```

```
    for (int i = 0; i < 26; ++i) {
```

```
        if (f[i] > 0 && f2[i] > 0)
```

```
            return false;
```

```
    }
```

```
    return true;
```

```
}
```

```
int main () {
```

```
    string s; getline(std::cin, s);
```

```
    vector<string> words;
```

```
    parse(s, words);
```

```
    int sz = (int)words.size();
```

```
    vector<vector<int>> freq(sz, vector<int>(26, 0));
```

```
    staticFreq(words, freq);
```

```
    int ans = 0;
```

```
    for (int i = 0; i < sz; ++i) {
```

```
        for (int j = i + 1; j < sz; ++j) {
```

```
        if (check(freq[i], freq[j]) == true) {  
            ans = max(ans, (int)words[i].size() * (int)words[j].size());  
        }  
    }  
}  
  
std::cout << ans << endl;  
}
```