题目描述：

给定一个元素类型为小写字符串的数组，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符串的数组，2 <= 数组长度<=100，0 < 字符串长度 <= 50。

输出描述：

两个没有相同字符的元素 长度乘积的最大值。

补充说明：

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。

```
#include <iostream>
#include <sstream>
#include <vector>
#include <bits/stdc++.h>
using namespace std;

vector<string> split(string& str,char ch){
    str.push_back(ch);
    vector<string> ans;
    int j = 0;
    for(int i = 0;i<str.size();++i){
        if(str[i]==ch){
            ans.push_back(str.substr(j,i-j));
            j=i+1;
        }
```

```
        }
        return ans;
}

bool sameChar(string str1,string str2){
        std::sort(str1.begin(),str1.end());
        std::sort(str2.begin(),str2.end());

        int index1 = 0,index2=0;
        while (index1<str1.size()&&index2<str2.size()) {
                if(str1[index1]==str2[index2]){
                        return true;
                }
                else if(str1[index1]<str2[index2]){
                        index1++;
                }
                else {
                index2++;
                }
        }
        return false;
}

int process(string str){
        int    maxVal = 0;
        auto vec = split(str, ',');
        for(int i =0;i<vec.size();i++){
                for(int j = i+1;j<vec.size();j++){
                        if(!sameChar(vec[i], vec[j])){
                                maxVal = max(maxVal,(int)(vec[i].size()*vec[j].size()));
                        }
                }
        }
        return maxVal;
}

int main() {
        vector<string> vstr;
        string s;
        while(getline(cin,s)){
                stringstream ss(s);
                string str;
                while (getline(ss,str)) {
                        vstr.push_back(str);
```

```
            }
        }
        cout<<process(vstr[0])<<endl;
        //cout<<vstr[0]<<endl;
        return 0;
}
// 64 位输出请用 printf("%lld")
```