

🔗 [编程题]抄送列表

时间限制：C/C++ 1秒，其他语言2秒 空间限制：C/C++ 32M，其他语言64M 热度指数：1268

📺 算法知识视频讲解

NowCoder每天要处理许多邮件，但他并不是在收件人列表中，有时候只是被抄送。他认为这些抄送的邮件重要性比自己在收件人列表里的邮件低，因此他要过滤掉这些次要的邮件，优先处理重要的邮件。现在给你一串抄送列表，请你判断目标用户是否在抄送列表中。

输入描述:

输入有多组数据，每组数据有两行。

第一行抄送列表，姓名之间用一个逗号隔开。如果姓名中包含空格或逗号，则姓名包含在双引号里。总长度不超过512个字符。

第二行只包含一个姓名，是待查找的用户的名字（姓名要完全匹配）。长度不超过16个字符。

输出描述:

如果第二行的名字出现在收件人列表中，则输出“Ignore”，表示这封邮件不重要；否则，输出“Important!”，表示这封邮件需要被优先处理。

示例1

输入

```
Joe,Kewell,Leon
Joe
"Letendre, Bruce",Joe,"Quan, William"
William
```

输出

```
Ignore
Important!
```

C++(clang++11) ▾

● ACM模式

🔄 重置

🧪 自测



```

3 /*
4 #include<iostream>
5 #include<string>
6 #include<sstream>
7 #include<vector>
8 #include<algorithm>
9 using namespace std;
10
11 int main()
12 {
13     string str;
14     string name;
15     while(getline(cin, str))
16     {
17         getline(cin, name);
18         vector<string> v;
19         stringstream ss(str);
20         string token;
21         while(getline(ss, token, ','))//逗号分割存放 单个单词
22         {
23             v.push_back(token);
24         }
25         for(int i=0; i<v.size(); i++)
26         {
27             while(v[i].find('"') != -1)
28             {
29                 int pos=v[i].find('"');
30                 v[i].erase(pos, 1);
31             }
32         }
33         if(find(v.begin(), v.end(), name) != v.end())
34             cout << "Ignore" << endl;
35         else
36             cout << "Important!" << endl;
37     }
38 }
39 return 0;
40 */
41 // write your code here.cpp
42 #include <iostream>
43 #include<string>
44 #include<vector>
45 #include<algorithm>
46 using namespace std;
47
48 int main() {
49     string str;
50     while(getline(cin, str)) {
51         vector<string> names; // 存放所有数据
52         int pos = 0;
53         while (pos < str.length()) { // 位置如果大于等于了长度，那么证明找完了，双引号的情况从该位置跳出
54             if(str[pos] == '\\') { // 找到双引号要处理双引号内部
55                 int end = str.find('"', pos + 1);
56                 names.push_back(str.substr(pos + 1, end - pos - 1)); // 截取出来放到names中
57                 pos = end + 2; // 跳过后面的双引号和逗号
58             }
59             else {
60                 int end = str.find(",", pos + 1);
61                 if (end == -1) { // 找不到逗号证明这是最后一个字符串了，直接存该字符串后跳出
62                     names.push_back(str.substr(pos, str.size() - pos));
63                     break;
64                 }
65                 names.push_back(str.substr(pos, end - pos));
66                 pos = end + 1; // 跳过后面的逗号
67             }
68         }
69         getline(cin, str);
70         if(names.end() == find(names.begin(), names.end(), str))
71             // find函数在两个迭代器中间查找，最后一个参数为要找的值
72             // 找到了返回对应迭代器，找不到返回区间末尾位置，本例子为end()
73             cout << "Important!" << endl;
74         else
75             cout << "Ignore" << endl;
76     }
77     return 0;
78 }
79

```