

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：238889

■ 算法知识视频讲解

开发一个简单错误记录功能小模块，能够记录出错的代码所在的文件名称和行号。

处理：

- 1、记录最多8条错误记录，循环记录，最后只用输出最后出现的八条错误记录。对相同的错误记录只记录一条，但是错误计数增加。最后一个斜杠后面的带后缀名的部分（保留最后16位）和行号完全匹配的记录才做算是“相同”的错误记录。
- 2、超过16个字符的文件名称，只记录文件的最后有效16个字符；
- 3、输入的文件可能带路径，记录文件名称不能带路径。
- 4、循环记录时，只以第一次出现的顺序为准，后面重复的不会更新它的出现时间，仍以第一次为准

输入描述：

每组只包含一个测试用例。一个测试用例包含一行或多行字符串。每行包括带路径文件名称，行号，以空格隔开。

输出描述：

将所有的记录统计并将结果输出，格式：文件名 代码行数 数目，一个空格隔开，如：

示例1

输入

```
D:\zwtymj\xccb\ljj\cqzlyaszjvlsjmkwoqijggmybr 645
E:\je\rzuwnjvnuz 633
C:\km\tgjwbp\gy\atl 637
F:\weioj\hadd\connsh\rwyfvzsopsuiqjnr 647
E:\ns\mfwj\wqkoki\eez 648
D:\cfmwafhhgeyawnool 649
E:\czt\opwip\osnll\c 637
G:\nt\f 633
F:\fop\yzwzqaop 631
F:\yay\jc\yzwzqaop 631
```

输出

```
rzwnjvnuz 633 1
atl 637 1
rwyfvzsopsuiqjnr 647 1
eez 648 1
fmwafhhgeyawnool 649 1
c 637 1
f 633 1
yzwzqaop 631 2
```

```

1 #include <iostream>
2 #include <string>
3 #include <vector>
4 #include <sstream>
5 #include <algorithm>
6
7 using namespace std;
8
9 string getFileName(string path){
10     int pos = path.rfind('\\'); //转义 找到最后一个\ 截取文件名
11     return path.substr(pos + 1);
12 }
13
14 string modifyName(string name){
15     if (name.size() > 16) {
16         name = name.substr(name.size() - 16); //长度超过16 截取部分
17     }
18
19     return name;
20 }
21
22 struct ErrRecord{ //定义一个结构体存放内容 并可计数
23     string file;
24     int lineNo;
25     int count;
26
27     ErrRecord(string file, int lineNo){
28         this->file = file;
29         this->lineNo = lineNo;
30         count = 1;
31     }
32
33     bool operator==(const ErrRecord &a){
34         return (file == a.file) && (lineNo == a.lineNo);
35     }
36 };
37
38 int main(){
39
40     string file;
41     int lineNo;
42     vector<ErrRecord> myvec;
43     while (cin >> file >> lineNo){
44         ErrRecord record(getFileName(file), lineNo);
45         auto res = find(myvec.begin(), myvec.end(), record);
46         if (res == myvec.end()){
47             myvec.push_back(record); //如果为新记录则计入myvec中 否则如果旧记录已存在 计数+1
48         }
49         else{
50             res->count++;
51         }
52     }
53
54     int n = 0;
55     for (auto item : myvec){
56         if (n+8 >= myvec.size()) //输出倒数8条记录
57             {
58                 cout << modifyName(item.file) << " " << item.lineNo << " " << item.count << endl;
59             }
60         n++;
61     }
62     return 0;
63 }
64
65
66

```