

限定语言: C(clang11), C++(clang++11), Pascal(fpc 3.0.2), Java(javac 1.8), Python2(2.7.3), PHP(7.4.7), C#(mcs 5.4), ObjC(gcc 5.4), Python3(3.9), JavaScript Node(12.18.2), JavaScript V8(6.0.0), Sqlite(3.7.9), R(4.0.3), Go(1.14.4), Ruby(2.7.1), Swift(5.3), matlab(Octave 5.2), bash(GNU bash4.3.46), pypy2(pypy2.7.13), pypy3(pypy3.6.1), Rust(1.44), Scala(2.11.12), Kotlin(1.4.10), Groovy(3.0.6), TypeScript(4.1.2), Mysql(8.0)

题目描述:

XX 市机场停放了多架飞机,每架飞机都有自己的航班号 CA3385, CZ6678, SC6508 等,航班号的前 2 个大写字母(或数字)代表航空公司的缩写,后面 4 个数字代表航班信息。但是 XX 市机场只有一条起飞用跑道,调度人员需要安排目前停留在机场的航班有序起飞。为保障航班的有序起飞,调度员首先按照航空公司的缩写(航班号前 2 个字母)对所有航班进行排序,同一航空公司的航班再按照航班号的后 4 个数字进行排序最终获得安排好的航班的起飞顺序。请编写一段代码根据输入的航班号信息帮助调度员输出航班的起飞顺序。

说明:

1、航空公司缩写排序按照从特殊符号\$ & *, 0~9, A~Z 排序;

输入描述:

第一行输入航班信息,多个航班号之间用逗号(“,”)分隔,输入的航班号不超过 100 个例如:

CA3385, CZ6678, SC6508, DU7523, HK4456, MK0987

备注:航班号为 6 位长度,后 4 位为纯数字,不考虑存在后 4 位重复的场景

输出描述:

CA3385, CZ6678, DU7523, HK4456, MK0987, SC6508

示例 1

输入:

CA3385, CZ6678, SC6508, DU7523, HK4456, MK0987

输出:

CA3385, CZ6678, DU7523, HK4456, MK0987, SC6508

说明:

输入目前停留在该机场的航班号,输出为按照调度排序后输出的有序的航班号

示例 2

输入:

MU1087, CA9908, 3U0045, FM1703

输出:

3U0045, CA9908, FM1703, MU1087

说明:

```
#include <string>
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
struct aa{
```

```
    string s1, s2;
```

```
}e[110];
```

```
bool cmp(aa x, aa y){  
    if(x.s1!=y.s1)  
        return x.s1<y.s1;  
    return x.s2<y.s2;  
}
```

```
int main() {  
    char s[1000010];  
    scanf("%s", s+1);  
    int m = strlen(s+1), n=1;  
    for(int i=1; i<=m; ++i){  
        if(s[i] == ',')  
            ++n;  
        else if(e[n].s1.size()<2)  
            e[n].s1+=s[i];  
        else  
            e[n].s2+=s[i];  
    }  
    sort(e+1, e+n+1, cmp);  
    cout<<e[1].s1<<e[1].s2;  
    for(int i=2; i<=n; ++i)  
        cout<<" "<<e[i].s1<<e[i].s2;  
    return 0;  
}  
// 64 位输出请用 printf("%lld")
```