

C++-字符串哈希表排序-现有两门选修课，每门选修课都有一部分学生选修

题目描述：

现有两门选修课，每门选修课都有一部分学生选修，每个学生都有选修课的成绩，需要你找出同时选修了两门选修课的学生，先按照班级进行划分，班级编号小的先输出，每个班级按照两门选修课成绩和的降序排序，成绩相同时按照学生的学号升序排序。

输入描述：

第一行为第一门选修课学生的成绩，第二行为第二门选修课学生的成绩，每行数据中学生之间以英文分号分隔，每个学生的学号和成绩以英文逗号分隔，学生学号的格式为 8 位数字(2 位院系编号+入学年份后 2 位+院系内部 1 位专业编号+所在班级 3 位学号)，学生成绩的取值范围为[0,100]之间的整数，两门选修课选修学生数的取值范围为[1-2000]之间的整数。

输出描述：

同时选修了两门选修课的学生的学号，如果没有同时选修两门选修课的学生输出 NULL，否则，先按照班级划分，班级编号小的先输出，每个班级先输出班级编号(学号前五位)，然后另起一行输出这个班级同时选修两门选修课的学生学号，学号按照要求排序(按照两门选修课成绩和的降序，成绩和相同时按照学号升序)，学生之间以英文分号分隔。

补充说明：

示例 1

输入：

01202021,75;01201033,95;01202008,80;01203006,90;01203088,100
01202008,70;01203088,85;01202111,80;01202021,75;01201100,88

输出：

01202
01202008;01202021
01203
01203088

说明：

同时选修了两门选修课的学生 01202021、01202008、01203088，这三个学生两门选修课的成绩和分别为 150、150、185，01202021、01202008 属于 01202 班的学生，按照成绩和降序，成绩相同时按学号升序输出的结果为 01202008;01202021,01203088 属于 01203 班的学生，按照成绩和降序，成绩相同时按学号升序输出的结果为 01203088，01202 的班级编号小于 01203 的班级编号，需要先输出。

示例 2

输入：

01201022,75;01202033,95;01202018,80;01203006,90;01202066,100
01202008,70;01203102,85;01202111,80;01201021,75;01201100,88

输出：

NULL

说明：

没有同时选修了两门选修课的学生，输出 NULL。

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

```

    int num;
    int score;
    node(int a, int b){
        num = a;
        score = b;
    }
};

bool cmp(const node a, const node b){
    if(a.score == b.score){
        return a.num < b.num;
    }
    return a.score > b.score;
}

map<string, int> one;
map<string, int> grade;
map<string, vector<node>> mp;
string get_cla(string s){
    string t = "";
    for(int i = 0; i < 5; i++) t += s[i];
    return t;
}

int get_number(string s){
    int sum = 0;
    for(int i = 5; i < 8; i++) sum = sum * 10 + (s[i] - '0');
    return sum;
}

string change(int n){
    int a[3];
    memset(a, 0, sizeof(a));
    for(int i = 0; i < 3; i++){
        a[i] = n % 10;
        n /= 10;
    }
    string s = "";
    for(int i = 2; i >= 0; i--) s += ('0' + a[i]);
    return s;
}

int main() {
    bool is_null = true;
    string s;
    cin >> s;
    bool num = true;
    string t = "";
    int sum = 0;

```

```

for(char ch : s){
    if(ch == ';'){
        one[t] = sum;
        sum = 0, t.clear(), num = true;
        continue;
    }
    if(ch == ','){
        num = false;
        continue;
    }
    if(num) t += ch;
    else sum = sum * 10 + (ch - '0');
}
//cout << t << " " << sum << endl;
one[t] = sum;
sum = 0, t.clear(), num = true;
cin >> s;
for(char ch : s){
    if(ch == ';'){
        if(one[t]){
            is_null = false;
            grade[t] = one[t] + sum;
        }
        sum = 0, t.clear(), num = true;
        continue;
    }
    if(ch == ','){
        num = false;
        continue;
    }
    if(num) t += ch;
    else sum = sum * 10 + (ch - '0');
}
if(one[t]){
    is_null = false;
    grade[t] = one[t] + sum;
}
if(is_null){
    cout << "NULL" << endl;
    return 0;
}
for(auto it : grade){
    string name = it.first;
    int score = it.second;
}

```

```
    string cla = get_cla(name);
    int number = get_number(name);
    mp[cla].push_back(node(number, score));
}
for(auto it : mp){
    sort(it.second.begin(), it.second.end(), cmp);
    cout << it.first << endl;
    int cou = 0;
    for(node x : it.second){
        if(cou) cout << ",";
        cout << it.first;
        cout << change(x.num);
        cou++;
    }
    cout << endl;
}
}
```