

```
1 //日期: 2018/ 时间:
2 #include <iostream>
3 #include <string>
4 #include <map>
5 using namespace std;
6 const int maxn = 2010; //总人数
7 const int INF = 1000000000; //无穷大
8
9 map<int,string> intToString; //编号->姓名
10 map<string,int> stringToInt; //姓名->编号
11 map<string,int> Gang; //head->人数
12
13 int G[maxn][maxn] = {0},weight[maxn] = {0}; //邻接矩阵G,点权weight
14 int n,k,numPerson = 0; //变数n,下限k,总人数numPerson
15 bool vis[maxn] = {false}; //标记是否被访问
16
17 //DFS函数访问单个连通块, nowVisit为当前访问的编号
18 //head为头目, numMember为成员编号, totalValue为连通块的总边权
19 void DFS(int nowVisit,int &head,int &numMember,int &totalValue){
20     numMember++; //成员人数加一
21     vis[nowVisit] = true; //标记nowVisit已访问
22     if(weight[nowVisit] > weight[head]){
23         head = nowVisit; //当前访问节点的点权大于头目的点权, 则更新头目
24     }
25     for(int i=0;i<numPerson;i++){ //枚举所有人
26         if(G[nowVisit][i] > 0){ //如果从nowVisit能到达i
27             totalValue += G[nowVisit][i]; //连通块的总边权增加该边权
28             G[nowVisit][i] = G[i][nowVisit] = 0; //删除这条边, 防止回头
29             if(vis[i] == false){ //如果i未被访问, 则递归访问i
30                 DFS(i,head,numMember,totalValue);
31             }
32         }
33     }
34 }
35
36 //DFSTrave函数遍历整个图, 获取每个连通块的信息
37 void DFSTrave(){
38     for(int i=0;i<numPerson;i++){ //枚举所有人
39         if(vis[i] == false){ //如果i未被访问
40             int head = i,numMember = 0,totalValue = 0; //头目, 成员数, 总边权
41             DFS(i,head,numMember,totalValue); //遍历i所在的连通块
42             if(numMember > 2 && totalValue > k){ //成员数大于2且总边权大于k
43                 //head人数为numMember
44                 Gang[intToString[head]] = numMember;
45             }
46         }
47     }
48 }
49
50 //change函数返回姓名str对应的编号
51 int change(string str){
52     if(stringToInt.find(str) != stringToInt.end()){ //如果str已经出现过
53         return stringToInt[str]; //返回编号
54     } else {
55         stringToInt[str] = numPerson; //str的编号为numPerson
56         intToString[numPerson] = str; //numPerson对应str
57     }
58 }
```

```
57     return numPerson++;           //总人数加一
58 }
59 }
60
61
62 int main(){
63     int w;
64     string str1,str2;
65     cin >> n >> k;                //人数和阈值
66     for(int i=0;i<n;i++){
67         cin >> str1 >> str2 >> w;
68         int id1 = change(str1);    //将str1转换为编号id1
69         int id2 = change(str2);    //将str2转换为编号id2
70         weight[id1] += w;          //id1的点权增加w
71         weight[id2] += w;          //id2的点权增加w
72         G[id1][id2] += w;          //边id1->id2的边权增加w
73         G[id2][id1] += w;          //边id2->id1的边权增加w
74     }
75
76     DFSTrave();                    //遍历整个图的所有连通块，获取Gang的信息
77     cout << Gang.size() << endl;   //Gang的个数
78     map<string,int>::iterator it;
79     for(it= Gang.begin(); it != Gang.end();it++){//遍历所有Gang
80         cout << it->first << " " << it->second << endl; //输出信息
81     }
82
83     return 0;
84 }
85
86
```