

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n,m;
5  map<string,int> mapa;
6  vector<int> parent,r;
7  vector<int> filhos;
8  int findSet(int i) {
9      return (parent[i] == -1) ? i : parent[i] = findSet(parent[i]);
10 }
11 bool isSameSet(int i, int j) {
12     return findSet(i) == findSet(j);
13 }
14 void unionF(int i,int j){
15     if (!isSameSet(i, j)) {
16         int x = findSet(i), y = findSet(j);
17         if (r[x] > r[y]) {
18             filhos[findSet(x)] += filhos[findSet(y)];
19             parent[y] = x;
20         } else {
21             filhos[findSet(y)] += filhos[findSet(x)];
22             parent[x] = y;
23             if (r[x] == r[y])
24                 r[y]++;
25         }
26     }
27 }
28 main(){
29     int i,j,k;
30     string a,b;
31
32     cin >> n;
33
34     for(i=0;i<n;i++){
35         cin >> m;
36         for(j=0;j<m;j++){
37             cin >> a >> b;
38             if(!mapa.count(a)){
39                 int aux = mapa.size();
40                 mapa[a] = aux;
41                 parent.push back(-1);
42                 filhos.push back(1);
43                 r.push back(0);
44             }
45             if(!mapa.count(b)){
46                 int aux = mapa.size();
47                 mapa[b] = aux;
48                 parent.push back(-1);
49                 filhos.push back(1);
50                 r.push back(0);
51             }
52             unionF(mapa[a],mapa[b]);
53             cout << filhos[findSet(mapa[a])] << endl;
54         }
55         mapa.clear();
56         parent.clear();
57         filhos.clear();
58         r.clear();
59     }
60 }
61
```