

现在我们需要查出一些作弊的问答社区中的 ID，作弊有两种：1.A 回答了 B 的问题，同时 B 回答了 A 的问题。那么 A 和 B 都是作弊。2.作弊 ID 用户 A 和作弊 ID 用户 B 同时回答了 C 的问题，那么 C 也是作弊。已知每个用户的 ID 是一串数字，一个问题可能有多个人回答。

```
1 #include <iostream>
2 #include <map>
3 #include <set>
4 #include <vector>
5 using namespace std;
6 /*
7 *利用 map 和 set，每个问题 a 对应一个 set b, 之后对 b 中的每个元素根据其 key 值找到其对应的 b'，然
8 后看 a 是否在里面。
9 *因为是 map 和 set, 不用排序，查找时间复杂度也低
10*刚提交，目前显示排名第一，，
11*/
12int main()
13{
14    int N;
15    while(cin>>N)
16    {
17        map<int,set<int>>> pairs;
18        while(N-->0)
19        {
20            int a,count,b;
21            cin>>a>>count;
22            set<int> bs;
23            while(count-->0)
```

```
24         {
25             cin>>b;
26             pairs[a].insert(b);
27         }
28         //pairs[a]=bs;
29     }
30     vector<pair<int,int>> all;
31     set<int> res;
32     for(map<int,set<int>>::iterator it=pairs.begin();it!=pairs.end();it++)
33     {
34         int a=it->first;
35         set<int> b=it->second;
36         for(set<int>::iterator i=b.begin();i!=b.end();i++)
37         {
38             if(a!=*i&&pairs[*i].find(a)!=pairs[*i].end())
39             {
40                 all.push_back(pair<int,int>(a,*i));
41                 res.insert(a);
42                 res.insert(*i);
43             }
44         }
45     }
46     for(map<int,set<int>>::iterator it=pairs.begin();it!=pairs.end();it++)
47     {
48         int a=it->first;
49         set<int> b=it->second;
```



```
50         for(int i=0;i<all.size();i++)
51             if(b.find(all[i].first)!=b.end() && b.find(all[i].second)!=b.end())
52                 res.insert(a);
53         }
54
55         cout<<res.size()<<endl;
56         for(set<int>::iterator it=res.begin();it!=res.end();it++)
57             {
58                 cout<<*it<<endl;
59             }
60     }
61     return 0;
62 }
```

$A[n,m]$ 是一个  $n$  行  $m$  列的矩阵,  $a[i,j]$ 表示  $A$  的第  $i$  行  $j$  列的元素, 定义  $x[i,j]$ 为  $A$  的第  $i$  行和第  $j$  列除了  $a[i,j]$ 之外所有元素(共  $n+m-2$  个)的乘积, 即  $x[i,j]=a[i,1]*a[i,2]*\dots*a[i,j-1]*\dots*a[i,m]*a[1,j]*a[2,j]*\dots*a[i-1,j]*a[i+1,j]*\dots*a[n,j]$ , 现输入非负整形的矩阵  $A[n,m]$ , 求  $\text{MAX}(x[i,j])$ , 即所有的  $x[i,j]$  中的最大值。

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main()
6  {
7      int n,m;
8      while(cin>>n>>m) {
```



```
9      vector<vector<int>> a(n, vector<int>(m, 0));
10     for(int i = 0; i < n; i++)
11         for(int j = 0; j < m; j++)
12             cin>>a[i][j];
13
14     long res = 0;
15     for(int i = 0; i < n; i++){
16         for(int j = 0; j < m; j++){
17             long sum = 1;
18             for(int row = 0; row < n; row++){
19                 if(row == i)
20                     continue;
21                 sum = sum*a[row][j];
22             }
23
24             for(int col = 0; col < m; col++){
25                 if(col==j)
26                     continue;
27                 sum = sum*a[i][col];
28             }
29
30             res = max(res, sum);
31         }
32     }
33     cout<<res<<endl;
34 }
```

```
35         return 0;  
36     }
```