

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef vector<vector<int> > vvi;
5  typedef vector<int> vi;
6  typedef vector<pair<int,int> > vii;
7  vvi Grafo(1005);
8  vii pontes;
9  int dfs low[1005];
10 int dfs num[1005];
11 int dfs parent[1005];
12 bool articulation vertex[1005];
13 int dfsNumberCounter,Children,dfsRoot,n,arti;
14 void print dfs(){
15     cout << pontes.size() << " critical links\n";
16     for(int i=0;i<pontes.size();i++){
17         cout << pontes[i].first << " - " << pontes[i].second << "\n";
18     }
19     cout << "\n";
20 }
21 void dfs(int u){
22     dfs low[u] = dfs num[u] = dfsNumberCounter++;
23     for(int i=0;i<Grafo[u].size();i++){
24         int v = Grafo[u][i];
25         if(dfs num[v]==-1){
26             dfs parent[v] = u;
27             if(u==dfsRoot)
28                 Children++;
29             dfs(v);
30
31             if(dfs low[v]>=dfs num[u]){
32                 articulation vertex[u] = true;}
33 //             if(dfs low[v]>dfs num[u])
34 //                 pontes.push back(make pair(u,v));
35             dfs low[u] = min(dfs low[u],dfs low[v]);
36         }
37         else if(v!=dfs parent[u])
38             dfs low[u] = min(dfs low[u],dfs num[v]);
39     }
40 }
41 void reset(){
42     for(int i=0;i<n;i++){
43         Grafo[i].clear();
44         dfs num[i] = -1;
45         dfs low[i] = 0;
46         dfs parent[i] = 0;
47         articulation vertex[i] = false;
48     }
49     //pontes.clear();
50     dfsNumberCounter = 0;
51     arti = 0;
52 }
53 void solve(){
54     for(int i=0;i<n;i++){
55         if(dfs num[i]==-1){
56             dfsRoot = i;
57             Children = 0;
58             dfs(i);
59             articulation vertex[i] = (Children>1);
60         }
61     }
62     for(int i=0;i<n;i++){
63         if(articulation vertex[i])
64             arti++;
65     }
66     cout << arti << "\n";
67 }
68
69 main(){
70     int i,j,k,from,to,m;

```

```
71
72     while(cin >> n and n){
73         reset();
74         for(i=0;i<n+1;i++){
75             cin >> from;
76             if(from==0)
77                 break;
78             while(cin >> to and to){
79                 Grafo[from-1].push back(to-1);
80                 Grafo[to-1].push back(from-1);
81                 if(getchar()=='\n')
82                     break;
83             }
84         }
85         solve();
86     }
87 }
88
```