

```
1. // strongly connected component (short) by Shadman
2.
3. #include <bits/stdc++.h>
4. using namespace std;
5. int indeg[2001],outdeg[2001];
6. vector<int>v1[2002],dg[2002];
7.
8. int pre[2001], lowlink[2001], sccno[2001], dfs_clock, scc_cnt;
9. stack<int> S;
10.
11. void dfs_scc(int u) {
12.     pre[u] = lowlink[u] = ++dfs_clock;
13.     S.push(u);
14.     for (int i = 0; i < v1[u].size(); i++) {
15.         int v = v1[u][i];
16.         if (!pre[v]) {
17.             dfs_scc(v);
18.             lowlink[u] = min(lowlink[u], lowlink[v]);
19.         } else if (!sccno[v])
20.             lowlink[u] = min(lowlink[u], pre[v]);
21.     }
22.     if (lowlink[u] == pre[u]) {
23.         scc_cnt++;
24.         while (!S.empty()) {
25.             int x = S.top(); S.pop();
26.             sccno[x] = scc_cnt;
27.             if (x == u) break;
28.         }
29.     }
30. }
31.
32. void find_scc(int n) {
33.     dfs_clock = scc_cnt = 0;
34.     memset(sccno, 0, sizeof(sccno));
35.     memset(pre, 0, sizeof(pre));
36.     for (int i = 0; i < n; i++)
37.         if (!pre[i]) dfs_scc(i);
38. }
39. int cnt(int node)
40. {
41.     if(outdeg[node])return outdeg[node];
42.
43.     int i,e,mx=0,x1;
44.     e=dg[node].size();
45.     for(i=0;i<e;i++)
46.     {
47.         x1=dg[node][i];
48.         mx=max(mx,cnt(x1));
49.     }
50.     outdeg[node]=mx+indeg[node];
51.     return outdeg[node];
52.
53.
54. }
55.
56. int main()
57. {
58.     int a,b,c,d,e,f,i,j,k,p,q,T;
```

```
59.     scanf("%d",&T);
60.     for(i=1;i<=T;i++)
61.     {   memset(indeg,0,sizeof(indeg));
62.         memset(outdeg,0,sizeof(outdeg));
63.         scanf("%d %d",&a,&b);
64.
65.
66.         for(j=0;j<=2000;j++){
67.             v1[j].clear();
68.             dg[j].clear();
69.         }
70.
71.
72.
73.         for(j=1;j<=b;j++)
74.         {
75.             scanf("%d %d",&c,&d);
76.             c--;
77.             d--;
78.
79.             //v2.push_back(pr);
80.             v1[c].push_back(d);
81.         }
82.         find_scc(a);
83.
84.         for(j=0;j<a;j++)
85.         {
86.             indeg[sccno[j]]++;
87.         }
88.
89.         for(j=0;j<a;j++)
90.         {
91.             e=v1[j].size();
92.             for(k=0;k<e;k++)
93.             {
94.                 f=v1[j][k];
95.                 if(sccno[j]!=sccno[f])
96.                 {
97.                     dg[sccno[j]].push_back(sccno[f]);
98.                 }
99.             }
100.        }
101.
102.        q=0;
103.        for(j=0;j<a;j++)
104.        {
105.            q=max(q,cnt(sccno[j]));
106.        }
107.
108.        printf("%d\n", q);
109.    }
110.    return 0;
111.
112.
113. }
```