BICOLORING/CYCLE DETECTING IN GRAPH :

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
vector<ll>g[1000005];
ll vis[1000005],level[1000005],nw,val=0;

void bfs(ll src)
{

    queue<ll>q;
    q.push(src);
    while(!q.empty())
    {
        ll y=q.front();
        q.pop();
        for(ll i=0;i<g[y].size();i++)
        {
            ll x=g[y][i];
            if(level[x]==level[y])
                    {
                        val=1;
                        return;
                    }
            if(level[x]==-1)
            {
                if(level[y]==0)
                    level[x]=1;
                else
                    level[x]=0;
                q.push(x);
            }

        }
    }

}
```

```
int main()
{
    ll t,cs=0;
    cin >> t;
    while(t--)
    {
        for(ll i=0;i<1000005;i++)
        {
            g[i].clear();
            level[i]=-1;
        }
        ll n,m;
        cin >> n >> m;//n=node number m=edge number
        for(ll i=0;i<m;i++)
        {
            ll u,v;
            scanf("%lld%lld",&u,&v);
            g[u].push_back(v);
            g[v].push_back(u);
        }
        ll f=0;
        printf("Scenario #%lld:\n",++cs);
        for(ll i=1;i<=n;i++)
        {
            if(level[i]==-1)
            {
                //cout << i << endl;
            level[i]=0;
            }
                bfs(i);
                if(val==1)
                {
                    printf("Suspicious bugs found!\n");
                    f=1;
                    break;
                }
        }
        if(f==0)
            printf("No suspicious bugs found!\n");


        val=0;
    }
}
```