ANSWER - 01

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e5;
vector<int>adj_list[N];
int main()
{
    int n;cin>>n;
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            int temp;cin>>temp;
            if(temp == 1)
                adj_list[i].push_back(j);
        }
    }
    for(int i=0; i<n; i++){
        cout<<i<<": ";
        for(auto x : adj_list[i]){
            cout<<x<<" ";
        }
        cout<<endl;
    }
    return 0;
}
```

ANSWER - 02

```
#include<bits/stdc++.h>
using namespace std;

const int N = 1e5;
vector<int>adj_list[N];
int visited[N];
int level[N];

void bfs(int src)
{
    queue<int>q;
    visited[src] = 1;
    level[src] = 0;
    q.push(src);

    while(!q.empty())
    {
        int head = q.front();
        q.pop();

        for(auto nxt : adj_list[head])
        {
            if(visited[nxt] == 0){
                visited[nxt] = 1;
                level[nxt] = level[head] + 1;
                q.push(nxt);
            }

        }



    }

}

int main()
{
    int v, e;cin>>v>>e;

    for(int i=0; i<e; i++){
        int f, t;cin>>f>>t;
        adj_list[f].push_back(t);
    }

    bfs(0);

    for(int i=0; i<v; i++){
        cout<<"node "<<i<<"--> "<<"level: "<<level[i]<<endl;
    }

    return 0;
}
```

ANSWER - 03

```
#include<bits/stdc++.h>
using namespace std;

const int N = 1e5;
vector<int>adj_list[N];
int visited[N];

bool dfs(int src)
{
    visited[src] = 1;

    for(auto nxt : adj_list[src]){
        if(visited[nxt] == 0){
            if(dfs(nxt)){
                return true;
            }
        }
        else if(visited[nxt] == 1){
            return true;
        }
    }

    return false;
}

int main()
{
    int v, e;cin>>v>>e;

    for(int i=0; i<e; i++){
        int f, t;cin>>f>>t;
        adj_list[f].push_back(t);
    }

    bool is_cyclic = false;

    for(int i=0; i<v; i++){
        if(visited[i] == 0){
            if(dfs(i)){
                is_cyclic = true;
                break;
            }
        }
    }

    if(is_cyclic){
        cout<<"YES"<<endl;
    }
    else
        cout<<"NO"<<endl;

    return 0;
}
```

ANSWER - 04

```
#include<bits/stdc++.h>
using namespace std;

const int N = 1e5;
int visited[N];
int color[N];
vector<int>adj_list[N];

bool dfs(int src)
{
    visited[src] = 1;

    for(auto nxt : adj_list[src])
    {
        if(visited[nxt] == 0){
            if(color[src] == 1) color[nxt] = 2;
            else color[nxt] = 1;

            if(!dfs(nxt)){
                return false;
            }
        }
        else if(color[src] == color[nxt])return false;
    }

    return true;
}



int main()
```

```
{
    int v, e;cin>>v>>e;

    for(int i=0; i<e; i++){
        int f, t;cin>>f>>t;
        adj_list[f].push_back(t);
        adj_list[t].push_back(f);
    }

    for(int i=0; i<v; i++){
        if(visited[i] == 0){
            color[i] = 1;
            if(!dfs(i)){
                cout<<"NO"<<endl;
                return 0;
            }
        }
    }

    cout<<"YES"<<endl;

    return 0;
}
```

ANSWER - 05

```
#include<bits/stdc++.h>
using namespace std;

const int N = 1e5;
int visited[N];
vector<int>adj_list[N];
```

```
void dfs(int src)
{
    visited[src] = 1;

    for(auto nxt : adj_list[src])
    {
        if(visited[nxt] == 0){
            visited[nxt] = 1;
            dfs(nxt);
        }
    }

}


int main()
{
    int v, e;cin>>v>>e;
    for(int i=0; i<e; i++){
        int f, t;cin>>f>>t;
        adj_list[f].push_back(t);
        adj_list[t].push_back(f);
    }
    int cnt = 0;
    for(int i=0; i<v; i++)
    {
        if(visited[i] == 0){
            cnt ++;
            dfs(i);
        }
    }

    cout<<cnt<<endl;

    return 0;
```

}