**Cycle Detection In Undirected Graph**

#include <bits/stdc++.h>>

bool bfs(vector<int> adj[], vector<int>& vis, int node) {

queue<pair<int, int>> q;

vis[node] = 1;

q.push({node, -1});

while (!q.empty()) {

int node = q.front().first;

int parent = q.front().second;

q.pop();

for (auto x : adj[node]) {

if (!vis[x]) {

vis[x] = 1;

q.push({x, node});

} else if (x != parent) {

return true;

}

}

}

return false;

}

string cycleDetection(vector<vector<int>>& edges, int n, int m) {

vector<int> adj[n + 1];

for (int i = 0; i < m; i++) {

adj[edges[i][0]].push\_back(edges[i][1]);

adj[edges[i][1]].push\_back(edges[i][0]);

}

vector<int> vis(n + 1, 0);

for (int i = 1; i <= n; i++) {

if (!vis[i]) {

if (bfs(adj, vis, i))

return "Yes";

}

}

return "No";

}