**Aaditya Sharma 199302117**BFS and DFS Traversal

#include<bits/stdc++.h>

using namespace std;

void edge(vector<int>ad[],int u,int v){

  ad[u].push\_back(v);

}

void bfs(int s,vector<int>adj[],bool visit[]){

  queue<int>q;

  q.push(s);

  visit[s]=true;

  while(!q.empty()){

    int u=q.front();

    cout<<u<<" ";

    q.pop();

    for(int i=0;i<adj[u].size();i++){

      if(!visit[adj[u][i]]){

        q.push(adj[u][i]);

        visit[adj[u][i]]=true;

      }

    }

  }

}

void dfs(int s,vector<int>adj[],bool visit[]){

  stack<int>stk;

  stk.push(s);

  visit[s]=true;

  while(!stk.empty()){

    int u=stk.top();

    cout<<u<<" ";

    stk.pop();

    for(int i=0;i<adj[u].size();i++){

      if(!visit[adj[u][i]]){

        stk.push(adj[u][i]);

        visit[adj[u][i]]=true;

      }

    }

  }

}

int main(){

  vector<int>adj[5];

  bool visit[5];

    for(int i=0;i<5;i++){

    visit[i]=false;

  }

  edge(adj,0,1);

  edge(adj,0,2);

  edge(adj,1,3);

  edge(adj,2,0);

  edge(adj,2,3);

  edge(adj,2,4);

  cout<<"BFS traversal:"<<"  ";

  bfs(0,adj,visit);

  cout<<endl;

  for(int i=0;i<5;i++){

    visit[i]=false;

  }

  cout<<"DFS traversal:"<<"  ";

  dfs(0,adj,visit);

  return 0;

}

OUTPUT

