9a) Write a program to traverse a graph using BFS method.

#include <stdio.h>

int n, i, j, visited[10], queue[10], front = -1, rear = -1;

int adj[10][10];

void bfs(int v)

{

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

if (adj[v][i] && !visited[i])

queue[++rear] = i;

if (front <= rear)

{

visited[queue[front]] = 1;

bfs(queue[front++]);

}

}

void main()

{

int v;

printf("Enter the number of vertices: ");

scanf("%d", &n);

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

{

queue[i] = 0;

visited[i] = 0;

}

printf("Enter graph data in matrix form: \n");

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

for (j = 1; j <= n; j++)

scanf("%d", &adj[i][j]);

printf("Enter the starting vertex: ");

scanf("%d", &v);

bfs(v);

printf("The node which are reachable are: \n");

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

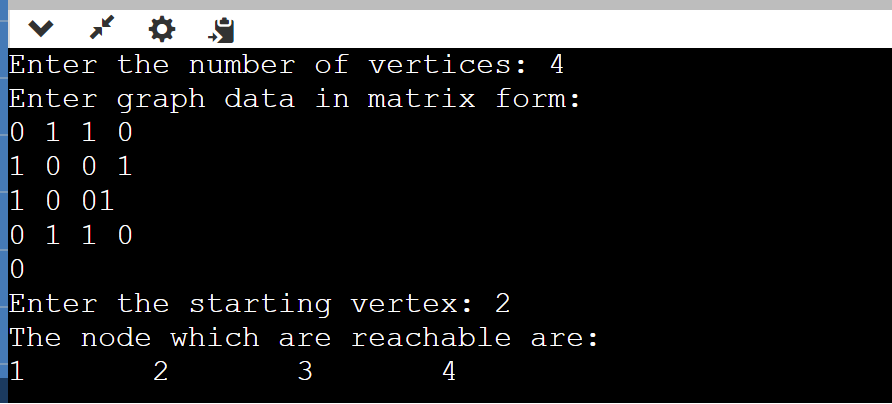
if (visited[i])

printf("%d\t", i);

else

printf("BFS is not possible. Not all nodes are reachable");

}



9b) Write a program to check whether given graph is connected or not using DFS method.

#include<stdio.h>

#include<conio.h>

int a[20][20], reach[20], n;

void dfs(int v) {

int i;

reach[v] = 1;

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

if (a[v][i] && !reach[i]) {

printf("\n %d->%d", v, i);

dfs(i);

}

}

int main(int argc, char \*\*argv) {

int i, j, count = 0;

printf("\n Enter number of vertices:");

scanf("%d", &n);

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

reach[i] = 0;

for (j = 1; j <= n; j++)

a[i][j] = 0;

}

printf("\n Enter the adjacency matrix:\n");

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

for (j = 1; j <= n; j++)

scanf("%d", &a[i][j]);

dfs(1);

printf("\n");

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

if (reach[i])

count++;

}

if (count == n)

printf("\n Graph is connected");

else

printf("\n Graph is not connected");

return 0;

}

