BFS traversal

#include <stdio.h>

#define max 100

int a[max][max], visited[max];

void bfs(int adj[max][max], int V, int s) {

int q[MAX], front = 0, rear = 0;

visited[s] = 1;

q[rear++] = s;

while (front < rear) {

int curr = q[front++];

printf("%d ", curr);

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

if (adj[curr][i] == 1 && visited[i] == 0) {

visited[i] = 1;

q[rear++] = i;

}

}

}

}

int main() {

int n, s;

printf(“Ruqaiyya Mahreen 1BM23EE044\n”);

printf("Enter number of vertices: ");

scanf("%d", &n);

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

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

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

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

}

}

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

visited[i] = 0;

}

printf("Enter the starting vertex: ");

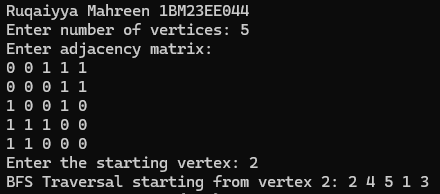
scanf("%d", &s);

printf("BFS Traversal starting from vertex %d: ", s);

bfs(a, n, s);

return 0;

}



Checking if graph is connected using DFS

#include <stdio.h>

#define max 100

int adj[max][max];

int visited[max];

int vertices;

void DFS(int vertex) {

visited[vertex] = 1;

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

if (adj[vertex][i]==1&&visited[i]==0) {

DFS(i);

}

}

}

int isConnected() {

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

visited[i] = 0;

}

DFS(1);

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

if (visited[i]==0) {

return 0;

}

}

return 1;

}

int main() {

int n=0,s=0,a[max][max];

printf(“Ruqaiyya Mahreen 1BM23EE044”);

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

scanf("%d",&n);

int i=0,j=0;

printf("Enter the adjacency matrix: ");

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

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

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

}

}

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

visited[i]=0;

printf("Enter the starting vertex: ");

scanf("%d",&s);

if(isConnected()){

printf("Graph is connected");

}

else{

printf("Graph not connected");

}

return 0;

}

