/\* Design , develop and implement a program in C for the following operations on graph G of cities

a. create a graph of N cities using adjacency matrix

b. print all the nodes reachable from a given starting node in a digraph using DFS/BFS method. \*/

#include <stdio.h>

#include <stdlib.h>

int a[20][20],q[20],visited[20],reach[10],n,i,j,f=0,r= -1,count=0;

void bfs(int v)

{

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

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

q[++r]=i;

if(f<=r)

{

visited[q[f]]=1;

bfs(q[f++]);

}

}

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);

count++;

dfs(i);

}

}

}

void main()

{

int v, choice;

while(1)

{

printf("1.Create a graph of N cities\n");

printf("2.BFS\n");

printf("3.DFS\n");

printf("4.Exit\n");

printf("Enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("\n Enter the number of cities:");

scanf("%d",&n);

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

{

q[i]=0;

visited[i]=0;

}

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

reach[i]=0;

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

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

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

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

break;

case 2: printf("Print using BFS...\n");

printf("\n Enter the starting vertex:");

scanf("%d",&v);

bfs(v);

if((v<1)||(v>n))

{

printf("\n Bfs is not possible");

}

else

{

printf("\n The nodes which are reachable from %d:\n",v);

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

if(visited[i])

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

}

break;

case 3: printf("Print using DFS and check for connected or not connected\n");

dfs(1);

if(count==n-1)

printf("\n Graph is connected\n");

else

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

break;

case 4: exit(0);

default: printf("Enter correct choice\n");

break;

}

}

}