**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 BFS method**

**c. Check whether a given graph is connected or not using DFS 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;

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

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 matrix form:\n");

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

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

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

printf("1.BFS\n 2.DFS\n 3.Exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1: 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 2: dfs(1);

if(count==n-1)

printf("\n Graph is connected");

else

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

break;

case 3: exit(0);

}

}

**Output:**

root:~/dslab #gedit bfs.c

root:~/dslab #cc bfs.c

root:~/dslab # ./a.out

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 0 1 0

1 0 1 0 1

0 1 0 1 0

1 0 1 0 0

0 1 0 0 0

1.BFS

2.DFS

3.Exit

2

1->2

2->3

3->4

2->5

Graph is connected

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 0 1 0

1 0 1 0 0

0 1 0 1 0

1 0 1 0 0

0 0 0 0 0

1.BFS

2.DFS

3.Exit

2

1->2

2->3

3->4

Graph is not connected

Enter the number of vertices:5

Enter graph data in matrix form:

0 1 1 0 0

0 0 0 1 0

0 0 0 0 0

0 0 1 0 0

0 0 1 0 0

1.BFS

2.DFS

3.Exit

1

Enter the starting vertex:1

The nodes which are reachable from 1:

2 3 4

Enter graph data in matrix form:

0 1 1 0 0

0 0 0 1 0

0 0 0 0 0

0 0 1 0 0

0 0 1 0 0

1.BFS

2.DFS

3.Exit

1

Enter the starting vertex:0

BFS is not possible