**//Program 3: Write and execute a menu driven program to print all the nodes reachable from a given starting node in a graph using DFS and BFS method.**

#include<stdio.h>

#include<conio.h>

void main()

{

int n,a[10][10],i,j,source,s[10],ch;

clrscr();

printf("Enter the Number of Nodes \n");

scanf("%d",&n);

printf("Enter the Adjacency Matrix \n");

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

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

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

printf("Enter the Source Node \n");

scanf("%d",&source);

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

s[i]=0;

printf("1: Depth first Traversal 2:Breadth first Traversal \n");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("The DFS order is \n");

Dfs(n,a,source,s);

break;

case 2: printf("The BFS order is \n");

Bfs(n,a,source,s);

break;

default: exit(0);

}

}

//---------------------------------------------------------------

Dfs(int n, int a[10][10], int source, int s[10])

{

int i;

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

s[source]=1;

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

if( s[i] == 0 && a[source][i] == 1)

Dfs(n,a,i,s);

}

//---------------------------------------------------------------

Bfs(int n, int a[10][10], int source, int s[10])

{

int q[10], f=1, r=1, i, u, v;

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

s[source]=1;

q[r] = source;

while ( f <= r )

{

u = q[f];

f = f + 1;

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

if( s[i] == 0 && a[u][i] == 1)

{

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

s[i] = 1;

r = r+1;

q[r] = i;

}

}

}

/\* OutPut

Enter the Number of Nodes

5

Enter the Adjacency Matrix

0 0 0 1 0

1 0 1 0 0

1 0 0 1 1

0 1 0 0 0

0 0 0 0 0

Enter the Source Node

1

The DFS order is

1 4 2 3 5

Enter the Number of Nodes

5

Enter the Adjacency Matrix

0 0 0 1 0

1 0 1 0 0

1 0 0 1 1

0 1 0 0 0

0 0 0 0 0

Enter the Source Node

1

1: Depth first Traversal 2:Breadth first Traversal

2

The BFS order is

1 4 2 3