**Experiment No.11**

**Aim: To implement** traversal of a directed graph through BFS and DFS

**Code:**

BFS:

#include<stdio.h>

int a[20][20],q[20],visited[20],n,f=-1,r=-1;

void bfs(int v)

{

int i;

for (i=0;i<n;i++) // check all the vertices in the graph

{

if(a[v][i] != 0 && visited[i] == 0) // adjacent to v and not visited

{

r=r+1;

q[r]=i; // insert them into queue

visited[i]=1; // mark the vertex visited

printf("%d ",i);

}

}

f=f+1; // remove the vertex at front of the queue

if(f<=r) // as long as there are elements in the queue

bfs(q[f]); // peform bfs again on the vertex at front of the queue

}main()

{

int v,i,j;

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

scanf("%d",&n);

for (i=0;i<n;i++) // mark all the vertices as not visited

{

visited[i]=0;

}

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

for (i=0;i<n;i++)

for (j=0;j<n;j++)

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

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

scanf("%d",&v);

f=r=0;

q[r]=v;

printf("\n BFS traversal is:\n");

visited[v]=1; // mark the starting vertex as visited

printf("%d ",v);

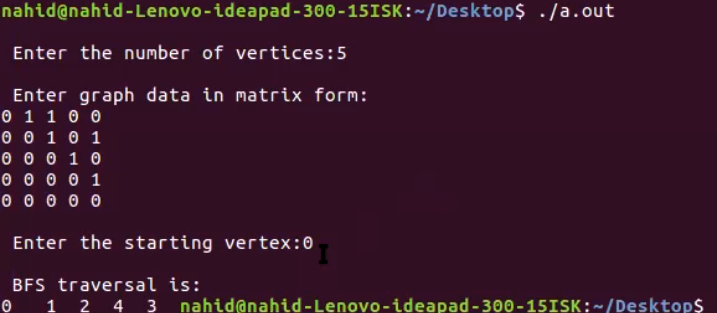
bfs(v);

if(r != n-1)

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

}

OUTPUT:



**DFS:**

#include<stdio.h>

void DFS(int);

int G[10][10],visited[10],n; //n is no of vertices and graph is sorted in array G[10][10]

void main()

{

int i,j;

printf("Enter number of vertices:");

scanf("%d",&n);

//read the adjecency matrix

printf("\nEnter adjecency matrix of the graph:");

for(i=0;i<n;i++)

for(j=0;j<n;j++)

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

//visited is initialized to zero

for(i=0;i<n;i++)

visited[i]=0;

DFS(0);

}

void DFS(int i)

{

int j;

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

visited[i]=1;

for(j=0;j<n;j++)

if(!visited[j]&&G[i][j]==1)

DFS(j)

}

