**Name: Sakshi Rajerndra Chaudhari**

**Roll No: 39**

**Practical Name : Implementation of Program based on breadth First Search**.

#include "iostream.h"

#include "conio.h"

class GRAPH

{

int n, G[][9];

public :

GRAPH(int);

void READ\_GRAPH();

void SHOW\_GRAPH();

void BFS(int);

};

GRAPH::GRAPH(int par)

{

n = par;

}

void GRAPH::READ\_GRAPH()

{

cout<<endl<<"Enter Adjecney Matrix: \n";

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

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

cin>>G[i][j];

}

void GRAPH::SHOW\_GRAPH()

{

cout<<endl<<"Adjecency matrix is:\n";

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

{

cout<<endl;

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

cout<<G[i][j]<<" ";

}

}

void GRAPH::BFS(int v)

{

int u,VISITED[10],QUE[10],rear,front;

rear=front=0;

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

VISITED[i]=0;

VISITED[v] =1;

u = v;

do

{

cout<<u<<" ";

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

{

if(G[u][i] == 1 && VISITED[i] ==0)

{

if(front == 0)

front = 1;

rear = rear + 1;

QUE[rear] = i;

VISITED[i] = 1;

}

}

if(front==0)

break;

else

{

u = QUE[front];

if(front == rear)

front = rear = 0;

else

front = front +1;

}

}while(1);

}

void main()

{

clrscr();

int v, n;

cout<<endl<<"Enter of An Vertices: ";

cin>>n;

GRAPH obj(n);

obj.READ\_GRAPH();

cout<<endl<<"Enter source Vertices: ";

cin>>v;

obj.BFS(v);

getch();

}