

AI LAB ASSIGNMENT-6

AP20110010812

K.MAHESH BABU

CSE-L

PROGRAM:

```
#include<stdio.h>
```

```
int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
```

```
int delete();
```

```
void add(int item);
```

```
void bfs(int s,int n);
```

```
void dfs(int s,int n);
```

```
void push(int item);
```

```
int pop();
```

```
void main()
```

```
{
```

```
    int n,i,s,ch,j;
```

```
    char c,dummy;
```

```
    printf("Enter number of nodes : ");
```

```
    scanf("%d",&n);
```

```
    printf("\n");
```

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

```

{
    for(j=1;j<=n;j++)
    {
        printf("Enter 1 if %d has a node with %d else 0 : ",i,j);
        scanf("%d",&a[i][j]);
    }
}
printf("\n\nAdjacency matrix :\n");
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        printf(" %d ",a[i][j]);
    }
    printf("\n");
}

do
{
    for(i=1;i<=n;i++)
    vis[i]=0;
    printf("\nMenu\n 1.B.F.S\n 2.D.F.S\n Enter your choice : ");
    scanf("%d",&ch);
    printf("\n Enter source node : ");
    scanf("%d",&s);

```

```

switch(ch)
{
    case 1: bfs(s,n);
        break;
    case 2: dfs(s,n);
        break;
}
printf("\nDo you want to continue(Y/N) : ");
scanf("%c",&dummy);
scanf("%c",&c);
} while((c == 'y') || (c == 'Y'));
}

```

void bfs(int s,int n) //BFS

```

{
    int p,i;
    add(s);
    vis[s]=1;
    p=delete();
    if(p!=0)
        printf(" %d",p);
    while(p!=0)
    {
        for(i=1;i<=n;i++)

```

```
        if((a[p][i]!=0)&&(vis[i]==0))
        {
            add(i);
            vis[i]=1;
        }
        p=delete();
        if(p!=0)
            printf(" %d ",p);
    }
    for(i=1;i<=n;i++)
        if(vis[i]==0)
            bfs(i,n);
}
```

```
void add(int item)
{
    if(rear==19)
        printf("QUEUE FULL");
    else
    {
        if(rear==-1)
        {
            q[++rear]=item;
            front++;
        }
        else
```

```
        q[++rear]=item;
    }
}
```

```
int delete()
{
    int k;
    if((front>rear) || (front==-1))
        return(0);
    else
    {
        k=q[front++];
        return(k);
    }
}
```

```
void dfs(int s,int n) //DFS
{
    int i,k;
    push(s);
    vis[s]=1;
    k=pop();
    if(k!=0)
        printf(" %d ",k);
    while(k!=0)
    {
```

```

    for(i=1;i<=n;i++)
        if((a[k][i]!=0)&&(vis[i]==0))
        {
            push(i);
            vis[i]=1;
        }
    k=pop();
    if(k!=0)
        printf(" %d ",k);
}
for(i=1;i<=n;i++)
    if(vis[i]==0)
        dfs(i,n);
}

```

```

void push(int item)
{
    if(top==19)
        printf("Stack overflow ");
    else
        stack[++top]=item;
}

```

```

int pop()
{
    int k;

```

```

    if(top== -1)
        return(0);
    else
    {
        k=stack[top--];
        return(k);
    }
}

```

OUTPUT:

```

input
Enter number of nodes : 3

Enter 1 if 1 has a node with 1 else 0 : 0
Enter 1 if 1 has a node with 2 else 0 : 1
Enter 1 if 1 has a node with 3 else 0 : 0
Enter 1 if 2 has a node with 1 else 0 : 1
Enter 1 if 2 has a node with 2 else 0 : 0
Enter 1 if 2 has a node with 3 else 0 : 1
Enter 1 if 3 has a node with 1 else 0 : 0
Enter 1 if 3 has a node with 2 else 0 : 1
Enter 1 if 3 has a node with 3 else 0 : 0

Adjacency matrix :
0 1 0
1 0 1
0 1 0

Menu
1.B.F.S
2.D.F.S
Enter your choice : 1

Enter source node : 2
2 1 3
Do you want to continue(Y/N) : y

Menu
1.B.F.S
2.D.F.S
Enter your choice : 2

Enter source node : 3
3 2 1
Do you want to continue(Y/N) : 

```