

## LAB PROGRAMS:

3. Write a C program depth first search (DFS) using array.

```
#include<stdio.h>
int G[10][10],n,visited;
void DFS(int k)
{
    int j;
    printf("%d",k);
    visited[k]=1;
    for(j=0;j<n;j++)
    {
        if(!visited[j]&&G[k][j]==1)
        {
            DFS(j);
        }
    }
}
int main()
{
    int k,j;
    printf("Enter the number of vertices: ");
    scanf("%d",&n);
    printf("\nEnter the adj graph matrix: ");
    for(k=0;k<n;k++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&G[k][j]);
        }
    }
    for(k=0;k<n;k++)
    {
        visited[k]=0;
        DFS(0);
    }
    return 0;
}
```

4. Write a C program breath first search (BFS) using array.

```
#include<stdio.h>
int a[100][100],b[100],visited[50],n,i,j,x=0,y=-1;
void bfs(int z)
{
    for(i=0;i<=n;i++)
```

```

{
if(a[z][i]&&!visited[i])
b[++y]=i;
if(x<=y){
visited[b[y]]=1;
bfs[b[y++]];
}
}
}
void main()
{
int z;
printf("Enter the number of vertices: ");
scanf("%d",&n);
for(i=1;i<n;i++)
{
b[i]=0;
visited[i]=0;
}
printf("Enter the graph data in the matrix form: ");
for (i=1;i<=n;i++){
for (j=1;j<=n;j++){
scanf("%d",&a[i][j]);
}
}
printf("Enter the vertex:");
scanf("%d",&z);
bfs(z);
printf("The nodes reachable are:\n");
for (i=1;i<=n;i++){
if(visited[i]){
printf("%d\t",i);}
else{
printf("Bfs is impossible");}
}
}
}
}
}

```