

ADA LAB PROGRAM 1

AIM: Write a program to obtain the following:

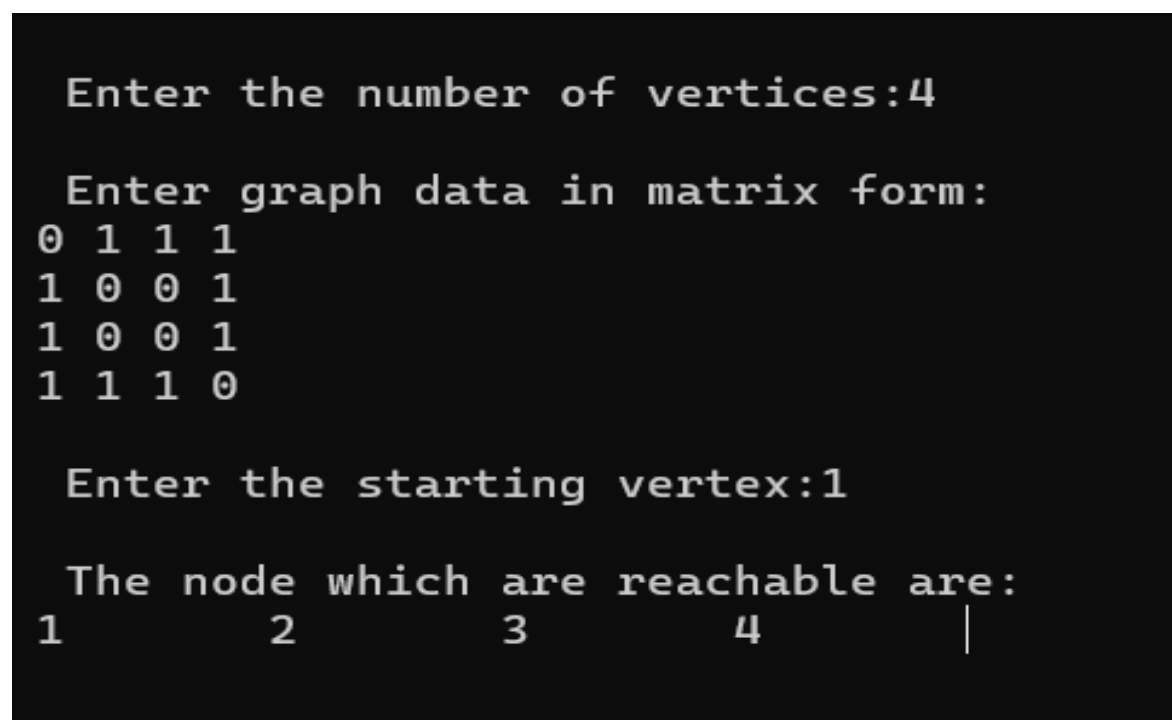
- a) Print all the nodes reachable from a given starting node in a diagraph using BFS method.
- b) Check weather a given graph is connected or not using DFS method.

SOURCE CODE: BFS METHOD

```
#include<stdio.h>
#include<conio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v)
{
for(i=1;i<=n;i++)
if(a[v][i] && !visited[i])
q[++r]=i;
if(f<=r)
{
visited[q[f]]=1;
bfs(q[f++]);
}
}
void main()
{
int v;
printf("\n Enter the number of vertices:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
q[i]=0;
visited[i]=0;
```

```
}  
printf("\n Enter graph data in matrix form:\n");  
for(i=1;i<=n;i++)  
for(j=1;j<=n;j++)  
scanf("%d",&a[i][j]);  
printf("\n Enter the starting vertex:");  
scanf("%d",&v);  
bfs(v);  
printf("\n The node which are reachable are:\n");  
for(i=1;i<=n;i++)  
if(visited[i])  
printf("%d\t",i);  
getch();  
}
```

OUTPUT SCREENSHOT



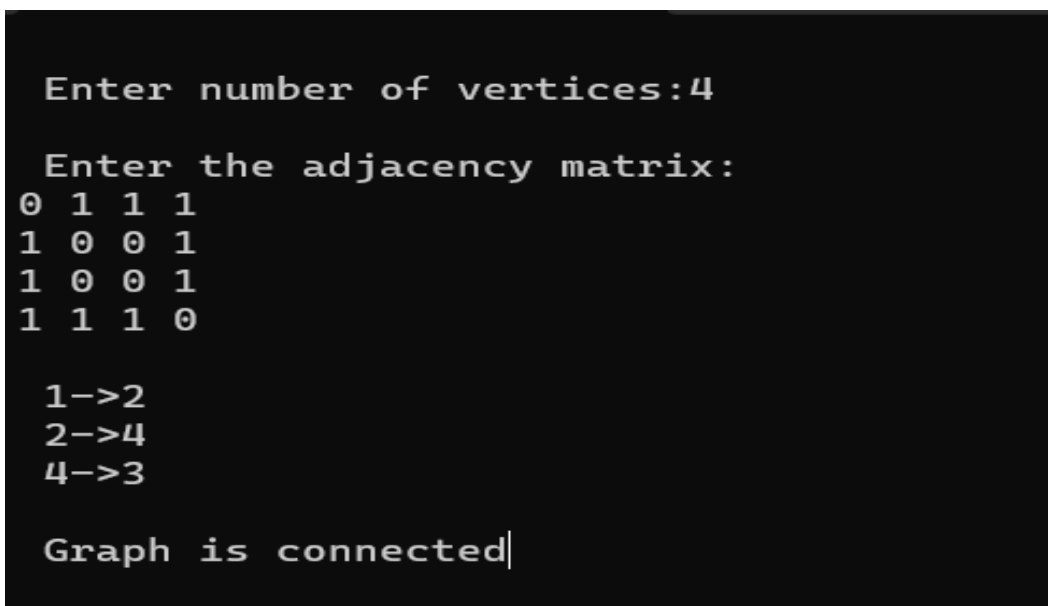
```
Enter the number of vertices:4  
  
Enter graph data in matrix form:  
0 1 1 1  
1 0 0 1  
1 0 0 1  
1 1 1 0  
  
Enter the starting vertex:1  
  
The node which are reachable are:  
1      2      3      4      |
```

SOURCE CODE: DFS METHOD

```
#include<stdio.h>
#include<conio.h>
int a[20][20],reach[20],n;
void dfs(int v)
{
    int i;
    reach[v]=1;
    for(i=1;i<=n;i++)
        if(a[v][i] && !reach[i])
        {
            printf("\n %d->%d",v,i);
            dfs(i);
        }
}
void main()
{
    int i,j,count=0;
    printf("\n Enter number of vertices:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        reach[i]=0;
        for(j=1;j<=n;j++)
            a[i][j]=0;
    }
    printf("\n Enter the adjacency matrix:\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            scanf("%d",&a[i][j]);
    dfs(1);
```

```
printf("\n");
for(i=1;i<=n;i++)
{
if(reach[i])
count++;
}
if(count==n)
printf("\n Graph is connected");
else
printf("\n Graph is not connected");
getch();
}
```

OUTPUT SCREENSHOT



```
Enter number of vertices:4
Enter the adjacency matrix:
0 1 1 1
1 0 0 1
1 0 0 1
1 1 1 0

1->2
2->4
4->3

Graph is connected|
```

The screenshot shows the execution of a C program that checks if a graph is connected. It prompts the user to enter the number of vertices (4) and the adjacency matrix. The matrix is entered as four rows of four values each. The program then outputs the edges: 1->2, 2->4, and 4->3, and finally states that the graph is connected.