

OUTPUT FOR DFS

The screenshot shows two windows of the OnlineCCompiler. The top window displays the C code for DFS:

```
main.c
1 #include<stdio.h>
2 #include<conio.h>
3 int a[20][20],reach[20],n;
4 void dfs(int v)
5 {
6     int i;
7     reach[v]=1;
8     for(i=1;i<n;i++)
9         if(a[v][i] && !reach[i])
10        {
11            printf("\n %d->%d",v,i);
12            dfs(i);
13        }
14    }
15 void main()
16 {
17     int i,j,count=0,s;
18     printf("\n Enter number of vertices:");
19     scanf("%d",&n);
20     for(i=1;i<n;i++)
21     {
22         reach[i]=0;
23         for(j=1;j<n;j++)
24             a[i][j]=0;
25     }
26     printf("\n Enter the adjacency matrix:\n");
27     for(i=1;i<n;i++)
28     {
29         printf("Row %d\n",i);
30         for(j=1;j<n;j++)
31             scanf("%d",&a[i][j]);
32     }
33     printf("\n Enter source vertex to check from:");
34     scanf("%d",&s);
35     dfs(s);
36     printf("\n");
37     for(i=1;i<n;i++)
38     {
39         if(reach[i])
40             count++;
41     }
42     if(count==n)
43         printf("\n Graph is connected");
44     else
45         printf("\n Graph is not connected");
46 }
```

The bottom window shows the execution of the code. It prompts for the number of vertices (4), the adjacency matrix (represented by three rows: Row 1 [1, 0, 0, 0], Row 2 [0, 22, 0, 0], Row 3 [0, 0, 3, 0], Row 4 [0, 0, 0, 4]), and the source vertex (2). The output shows the graph is connected.

```
Enter number of vertices:4
Enter the adjacency matrix:
Row 1
1
0
0
0
Row 2
0
22
0
0
Row 3
0
0
3
0
Row 4
0
0
0
4
Enter source vertex to check from:2
2->1
2->3
Graph is connected
..Program finished with exit code 0
Press ENTER to exit console.
```

CODE FOR DFS

```
#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,s;
    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++)
    {printf("Row %d\n",i);
        for(j=1;j<=n;j++)
            scanf("%d",&a[i][j]);
    }
    printf("\n Enter source vertex to check from:");
    scanf("%d",&s);
    dfs(s);
}

```

```
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 FOR BFS

```
#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;
        r++;
    visited[q[f]]=1;
    bfs(q[f+1]);
}
void main()
{
    int v;
    printf("\nEnter the number of vertices:");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        q[i]=0;
        visited[i]=0;
    }
    printf("\nEnter graph data in matrix form:");
    for(i=1;i<n;i++)
    {
        printf("Row %d\n",i);
        for(j=1;j<n;j++)
        scanf("%d",&a[i][j]);
    }
    printf("\nEnter the starting vertex:");
    scanf("%d",&v);
    bfs(v);
    printf("\nThe node which are reachable are:\n");
    for(i=1;i<n;i++)
    if(visited[i])
        printf("%d\t",i);
    getch();
}
```

```
Enter the number of vertices:2
Row 1
1
Enter graph data in matrix form:
Row 1
1
Row 2
1
2
Enter the starting vertex:
The node which are reachable are:
1      2
...program finished with exit code 0
Press ENTER to exit console.
```

CODE FOR BFS

```
#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++)
    { printf("Row %d\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();  
}
```