

OUTPUT FOR DFS

The image displays two screenshots of the OnlineGDB IDE. The top screenshot shows the C source code for a Depth-First Search (DFS) algorithm. The code includes headers for `stdio.h` and `conio.h`, declares a 2D array `a` of size 20x20, and a `reach` array of size 20. It defines a recursive `dfs` function and a `main` function that prompts the user for the number of vertices, the adjacency matrix, and the source vertex. The bottom screenshot shows the execution output of the program. The user has entered 3 for the number of vertices, followed by the adjacency matrix: Row 1: 1 2 3, Row 2: 1 2 2, Row 3: 2 2 1. The source vertex is 2. The output indicates that the graph is connected and the program finished with exit code 0.

```
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     s=dfs(s);
35     printf("\n");
36     for(i=1;i<=n;i++)
37     {
38         if(reach[i])
39             count++;
40     }
41     if(count==n)
42         printf("\n Graph is connected");
43     else
44         printf("\n Graph is not connected");
45     getch();
46 }
```

Enter number of vertices:3

Enter the adjacency matrix:

Row 1

1 2 3

Row 2

1 2 2

Row 3

2 2 1

Enter source vertex to check from:2

2->1

1->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;
```

```

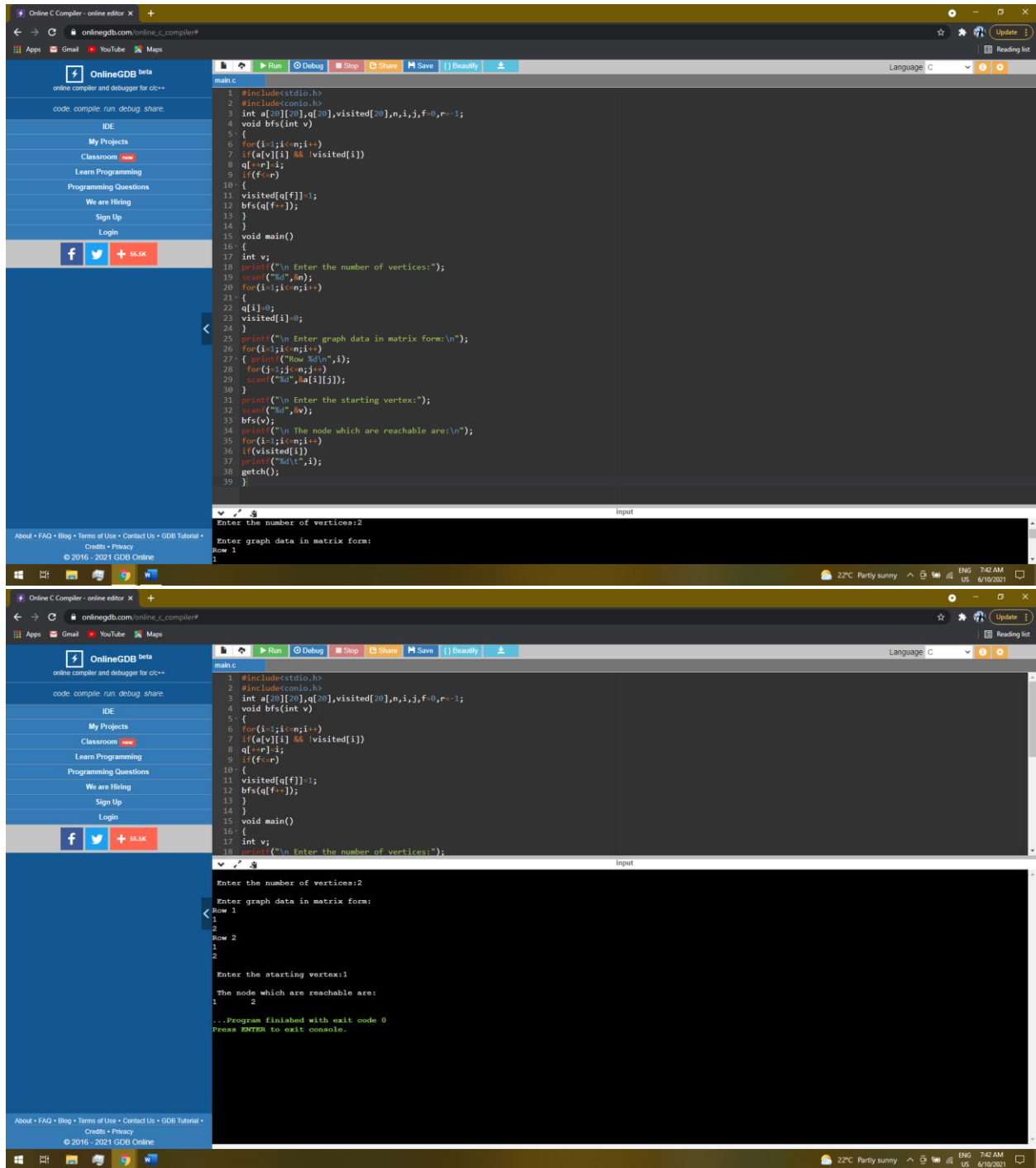
void dfs(int v)
{
    int i;
    reach[v]=1;
    for(i=1;i<=n;i++)
    if(a[v][i] && !reach[i])
    {
        printf("\n %d->%d",v,i);
        dfs(i);
    }
}

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



```
1 #include<stdio.h>
2 #include<conio.h>
3 int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
4 void bfs(int v)
5 {
6     for(i=1;i<=n;i++)
7     if(a[v][i] && !visited[i])
8     {
9         q[f]=i;
10        if(f==r)
11        {
12            visited[q[f]]=1;
13            bfs(q[f++]);
14        }
15    }
16 }
17 void main()
18 {
19     int v;
20     printf("\n Enter the number of vertices:");
21     scanf("%d",&n);
22     for(i=1;i<=n;i++)
23     {
24         q[i]=0;
25         visited[i]=0;
26     }
27     printf("\n Enter graph data in matrix form:\n");
28     for(i=1;i<=n;i++)
29     {
30         printf("Row %d\n",i);
31         for(j=1;j<=n;j++)
32         {
33             scanf("%d",&a[i][j]);
34         }
35     }
36     printf("\n Enter the starting vertex:");
37     scanf("%d",&v);
38     bfs(v);
39     printf("\n The nodes which are reachable are:\n");
40     for(i=1;i<=n;i++)
41     if(visited[i])
42     printf("%d\t",i);
43     getch();
44 }
```

Enter the number of vertices:2

Enter graph data in matrix form:

Row 1

1

Row 2

1

2

Enter the starting vertex:1

The nodes 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();  
}
```