

Name : JAYKUMAR GOR  
Roll no : 16

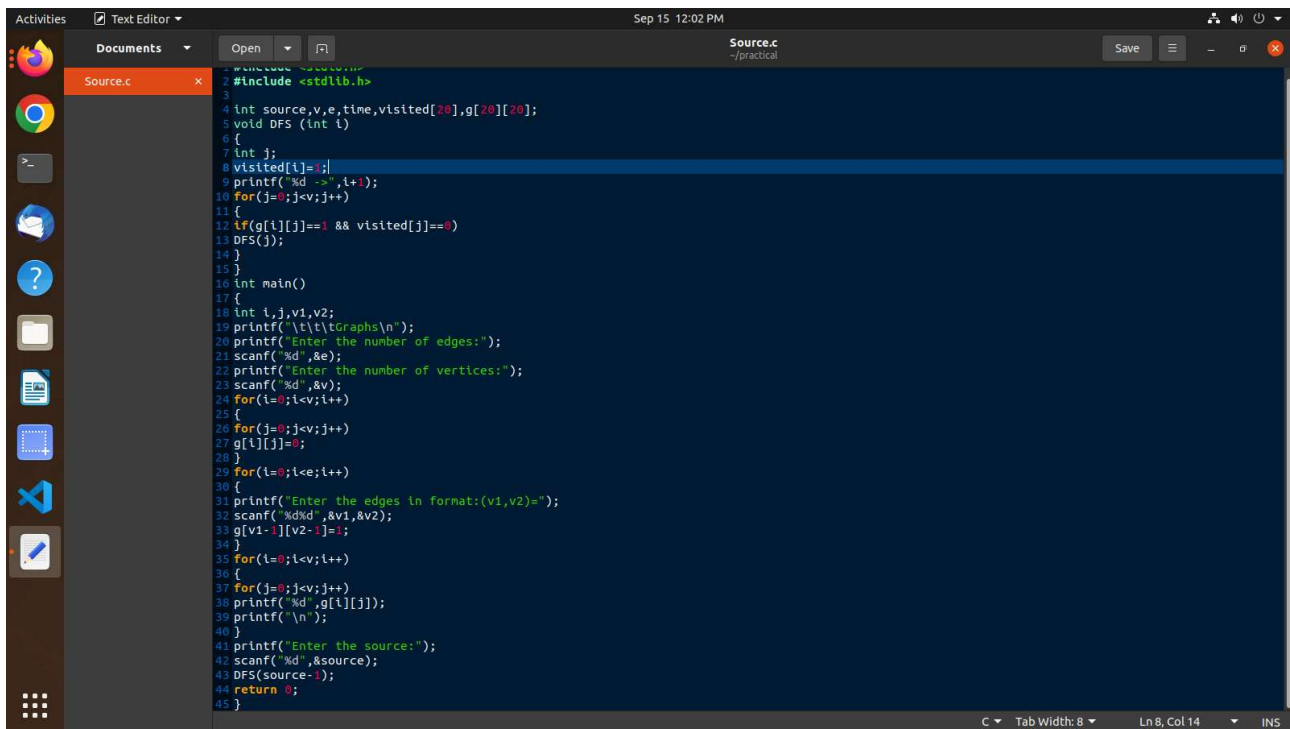
Code :

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

int source,v,e,time,visited[20],g[20][20];
void DFS (int i)
{
    int j;
    visited[i]=1;
    printf("%d ->",i+1);
    for(j=0;j<v;j++)
    {
        if(g[i][j]==1 && visited[j]==0)
            DFS(j);
    }
}

int main()
{
    int i,j,v1,v2;
    printf("\t\t\tGraphs\n");
    printf("Enter the number of edges:");
    scanf("%d",&e);
    printf("Enter the number of vertices:");
    scanf("%d",&v);
    for(i=0;i<v;i++)
    {
        for(j=0;j<v;j++)
            g[i][j]=0;
    }
    for(i=0;i<e;i++)
    {
        printf("Enter the edges in format:(v1,v2)=");
        scanf("%d%d",&v1,&v2);
        g[v1-1][v2-1]=1;
    }
    for(i=0;i<v;i++)
    {
        for(j=0;j<v;j++)
            printf("%d",g[i][j]);
        printf("\n");
    }
    printf("Enter the source:");
    scanf("%d",&source);
    DFS(source-1);
    return 0;
}
```

## Output ScreenShort

A screenshot of a Linux desktop environment with a text editor open. The text editor has a dark theme and shows a C program for Depth-First Search (DFS). The code includes headers for stdio and stdlib, defines a 20x20 adjacency matrix 'g' and a visited array 'visited'. The main function prompts the user for the number of vertices 'n' and the source vertex 'source'. It then calls the DFS function starting from the source vertex. The DFS function recursively visits all adjacent vertices that have not been visited yet, printing the vertex number and the current state of the adjacency matrix. The status bar at the bottom indicates the current line is 8 and column is 14.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int source,v,e,time,visited[20],g[20][20];
5 void DFS (int i)
6 {
7     int j;
8     visited[i]=1;
9     printf("%d -> ",i+1);
10    for(j=0;j<v;j++)
11    {
12        if(g[i][j]==1 && visited[j]==0)
13            DFS(j);
14    }
15 }
16 int main()
17 {
18     int i,j,v1,v2;
19     printf("\t\t\tGraphs\n");
20     printf("Enter the number of edges:");
21     scanf("%d",&e);
22     printf("Enter the number of vertices:");
23     scanf("%d",&v);
24     for(i=0;i<v;i++)
25     {
26         for(j=0;j<v;j++)
27             g[i][j]=0;
28     }
29     for(i=0;i<e;i++)
30     {
31         printf("Enter the edges in format:(v1,v2)=");
32         scanf("%d%d",&v1,&v2);
33         g[v1-1][v2-1]=1;
34     }
35     for(i=0;i<v;i++)
36     {
37         for(j=0;j<v;j++)
38             printf("%d",g[i][j]);
39         printf("\n");
40     }
41     printf("Enter the source:");
42     scanf("%d",&source);
43     DFS(source-1);
44     return 0;
45 }
```

I\_1

```
#include<stdio.h>
#include<stdlib.h>
int a[20][20],q[20],visit[20],n,f=-1,r=-1;
```

```
void bfs(int v)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(a[v][i] !=0 && visit[i]==0)
        {
            r=r+1;
            q[r]=i;
            visit[i]=1;
            printf("%d",i);
        }
    }
    f=f+1;
    bfs(q[f]);
}
```

```
int main()
{
    int v,i,j;
    printf("\n Enter no. of vertices: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
```

```

    visit[i]=0;
}
printf("\n\nEnter graph data in matrix form:\n");
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
printf("\n\nEnter the starting vertex: ");
scanf("%d",&v);
bfs(v);
if(r!=n-1)
printf("\nBFS not possible");
printf("\n\n");
return 0;
}

```

```

dl0415@itadmn: ~/practical
Source.c:39:16: error: expected ';' before '}' token
39 |         return 0;
   |                ^
40 |     }
   |     ^
dl0415@itadmn:~/practical$ gcc exp7.c
gcc: error: exp7.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
dl0415@itadmn:~/practical$ gcc Source.c
dl0415@itadmn:~/practical$ ./a.out
Graphs
Enter the number of edges: 5
Enter the number of vertices: 6
Enter the edges in format:(v1,v2)=9,4
Segmentation fault (core dumped)
dl0415@itadmn:~/practical$ gcc Source.c
dl0415@itadmn:~/practical$ ./a.out
Graphs
Enter the number of edges: 5
Enter the number of vertices: 5
Enter the edges in format:(v1,v2)=(6,7)
Segmentation fault (core dumped)
dl0415@itadmn:~/practical$ gcc Source.c
dl0415@itadmn:~/practical$ ./a.out
Graphs
Enter the number of edges: 7
Enter the number of vertices: 6
Enter the edges in format:(v1,v2)= 5 4
Enter the edges in format:(v1,v2)= 3 2
Enter the edges in format:(v1,v2)= 1 6
Enter the edges in format:(v1,v2)= 3 2
Enter the edges in format:(v1,v2)= 5 8
Enter the edges in format:(v1,v2)= 9 7
Enter the edges in format:(v1,v2)= 4 3
000001
000000
010000
001000
000100
000000
Enter the source: 5
5 -> 4 -> 3 -> 2 -> 1
dl0415@itadmn:~/practical$

```