

MEET PATEL
SY IT – 43
DSA EXP NO.7

CODE:

The screenshot displays a Linux desktop environment with a dark theme. The top panel shows the 'Activities' button, a search bar, and the system clock indicating 'Sep 12, 14:57'. The main workspace contains a single window titled 'Text Editor' with the filename 'meet7.c'. The code in the editor is a C++ program that implements a Depth-First Search (DFS) algorithm to find the shortest path in a graph. The program uses a 2D array 'G' to represent the graph and a 'visited' array to track visited nodes. The 'main' function prompts the user to enter the number of edges and vertices, followed by the edges themselves. It then calls the 'DFS' function starting from a source node. The output shows the shortest path from node 1 to node 4, which is 1 2 3 4.

```
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     int j;
7     visited[i]=1;
8     printf("%d->", i+1);
9     for(j = 0; j<V;j++){
10         if(G[i][j]==1&&visited[j]==0)
11             DFS(j);
12     }
13 }
14
15 int main(){
16     int i,j,v1,v2;
17     printf("\t\t\tGraphs\n");
18     printf("enter no of edges ");
19     scanf("%d",&E);
20     printf("enter no of vertices ");
21     scanf("%d",&V);
22     for(i=0;i<V;i++){
23         for(j=0;j<V;j++){
24             G[i][j] = 0;
25         }
26     }
27     for(i=0;i<E;i++){
28         printf("enter the edges (format:V1 V2) : ");
29         scanf("%d%d",&v1,&v2);
30         G[v1-1][v2-1] = 1;
31     }
32     for(i=0;i<V;i++){
33         for(j=0;j<V;j++){
34             printf("\t %d ",G[i][j]);
35         }
36         printf("\n");
37     }
38     printf("Enter the source :");
39     scanf("%d",&source);
40     DFS(source-1);
41     return 0;
42 }
43
44 }
```

The bottom status bar shows 'C' (language), 'Tab Width: 8', 'Ln 35, Col 25', and 'INS' (insert mode).

OUTPUT:

```

Enter the source :7
7->lab@lab-HP-ProDesk-400-G7-Microtower-PC:~$ gedit meet7.c
lab@lab-HP-ProDesk-400-G7-Microtower-PC:~$ ./a.out
Graphs

enter no of edges 7
enter no of vertices 8
enter the edges (format:V1 V2) : 1 5
enter the edges (format:V1 V2) : 4 8
enter the edges (format:V1 V2) : 1 2
enter the edges (format:V1 V2) : 9 6
enter the edges (format:V1 V2) : 1 3
enter the edges (format:V1 V2) : 2 4
enter the edges (format:V1 V2) : 5 4
01101000
00010000
00000000
00000001
00010000
00000000
00000000
00000000
00000000
Enter the source :7

```