

```
1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int cost[10][10], i, j, k, n, qu[10], front, rear, v, visit[10], visited[10];
7 int stk[10], top, visit1[10], visited1[10];
8
9 int main()
10 {
11     int m;
12
13     cout << "Enter number of vertices : ";
14     cin >> n;
15
16     cout << "Enter number of edges : ";
17     cin >> m;
18
19     cout << "\nEDGES :\n";
20
21     for (k = 1; k <= m; k++)
22     {
23         cin >> i >> j;
24         cost[i][j] = 1;
25         cost[j][i] = 1;
26     }
27
28     // Display function
29     cout << "The adjacency matrix of the graph is : " << endl;
30
31     for (i = 0; i < n; i++)
32     {
33         for (j = 0; j < n; j++)
34         {
35             cout << " " << cost[i][j];
36         }
37         cout << endl;
38     }
39
40     cout << "Enter initial vertex : ";
41     cin >> v;
42
43     cout << "The BFS of the Graph is\n";
44     cout << v << endl;
45     visited[v] = 1;
46     k = 1;
47
48     while (k < n)
49     {
50         for (j = 1; j <= n; j++)
51             if (cost[v][j] != 0 && visited[j] != 1 && visit[j] != 1)
52             {
53                 visit[j] = 1;
54                 qu[rear++] = j;
55             }
56     }
57 }
```

```

56
57         v = qu[front++];
58         cout << v << " ";
59         k++;
60         visit[v] = 0;
61         visited[v] = 1;
62     }
63
64     cout << endl << "Enter initial vertex : ";
65     cin >> v;
66
67     cout << "The DFS of the Graph is\n";
68     cout << v << endl;
69     visited[v] = 1;
70     k = 1;
71
72     while (k < n)
73     {
74         for (j = n; j >= 1; j--)
75             if (cost[v][j] != 0 && visited1[j] != 1 && visit1[j] != 1)
76             {
77                 visit1[j] = 1;
78                 stk[top] = j;
79                 top++;
80             }
81
82         v = stk[--top];
83         cout << v << " ";
84         k++;
85         visit1[v] = 0;
86         visited1[v] = 1;
87     }
88
89     return 0;
90 }
91

```

## Output :

```

D:\SE Computer\LAB CODES\DSA\DSA6.exe
Enter number of vertices : 5
Enter number of edges : 6
EDGES :
0 1
0 2
1 3
1 4
2 4
3 4
The adjacency matrix of the graph is :
0 1 1 0 0
1 0 0 1 1
1 0 0 0 1
0 1 0 0 1
0 1 1 1 0
Enter initial vertex : 0
The BFS of the Graph is
0
1 2 3 4
Enter initial vertex : 0
The DFS of the Graph is
0
1 3 4 2
-----
Process exited after 46.43 seconds with return value 0
Press any key to continue . . .

```