

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int a[20][20], q[20], visited[20], reach[10], n, i, j, f = 0, r = -1, count =
    0;
5
6  void bfs(int v) {
7      for (i = 1; i ≤ n; i++) {
8          if (a[v][i] && !visited[i])
9              q[++r] = i;
10     }
11     if (f ≤ r) {
12         visited[q[f]] = 1;
13         bfs(q[f++]);
14     }
15 }
16
17 void dfs(int v) {
18     int i;
19     reach[v] = 1;
20     for (i = 1; i ≤ n; i++) {
21         if (a[v][i] && !reach[i]) {
22             printf("\n %d→%d", v, i);
23             count++;
24             dfs(i);
25         }
26     }
27 }
28
29 void main() {
30     int v, choice;
31     printf("\n Enter the number of vertices: ");
32     scanf("%d", &n);
33     for (i = 1; i ≤ n; i++) {
34         q[i] = 0;
35         visited[i] = 0;
36     }
37     for (i = 1; i ≤ n - 1; i++)
38         reach[i] = 0;
39     printf("\n Enter graph data in matrix form:\n");
40     for (i = 1; i ≤ n; i++)
41         for (j = 1; j ≤ n; j++)
42             scanf("%d", &a[i][j]);
43     printf("1. BFS\n2. DFS\n3. Exit\n");
44     scanf("%d", &choice);
45     switch (choice) {
46         case 1:
47             printf("\n Enter the starting vertex: ");
48             scanf("%d", &v);
49             bfs(v);
50             if ((v < 1) || (v > n)) {
51                 printf("\n BFS is not possible");
52             } else {
```

```
53         printf("\n The nodes which are reachable from %d:\n", v);
54         for (i = 1; i ≤ n; i++) {
55             if (visited[i])
56                 printf("%d\t", i);
57         }
58     }
59     break;
60 case 2:
61     dfs(1);
62     if (count == n - 1)
63         printf("\n Graph is connected");
64     else
65         printf("\n Graph is not connected");
66     break;
67 case 3:
68     exit(0);
69 }
70 }
71
```