

```

1  #include <iostream>
2  #include <vector>
3  #include <queue>
4
5  using namespace std;
6
7  #define MAX 10000
8
9  bool visited[MAX];
10
11 void dfs(vector<vector<int>> n, int node)
12 {
13     for (int neighbor : n[node])
14     {
15         if (!visited[neighbor])
16         {
17             dfs(n, neighbor);
18         }
19     }
20     visited[node] = true;
21     cout << node << " ";
22 }
23
24 void bfs(vector<vector<int>> v)
25 {
26     queue<int> q;
27     int start = 0;
28     q.push(start);
29     visited[start] = true;
30
31     while (!q.empty())
32     {
33         int first = q.front();
34         cout << first << " ";
35         for (int n : v[first])
36         {
37             if (!visited[n])
38             {
39                 q.push(n);
40                 visited[n] = 1;
41             }
42         }
43         q.pop();
44     }
45 }
46
47 int main()
48 {
49     int n;
50     cout << "Enter how many node you have : ";
51
52     cin >> n;
53     vector<vector<int>> v;
54
55     // vector input
56     for (int i = 0; i < n; i++)
57     {
58         vector<int> n;
59         cout << "Enter how many neighbour for node " << i << " : ";
60
61         int neighbour;
62         cin >> neighbour;
63
64         for (int j = 0; j < neighbour; j++)

```

```
67     {
68         int val;
69         cin >> val;
70         n.push_back(val);
71     }
72     v.push_back(n);
73 }
74
75 dfs(v, 0);
76
77 bfs(v);
78 return 0;
79 }
```