```
11…#include <stdio.h>
int a[20][20];     // Adjacency Matrix
int visited[20];  // Visited array
int n;              // Number of cities

/* DFS function */
void dfs(int source)
{
    int v;
    visited[source] = 1;
    printf("%d ", source);

    for (v = 1; v <= n; v++)
    {
        if (a[source][v] == 1 && visited[v] == 0)
        {
            dfs(v);
        }
    }
}

/* BFS function */
void bfs(int source)
{
    int queue[20], front = 0, rear = 0;
    int v;
    visited[source] = 1;
    queue[rear++] = source;
    while (front < rear)
    {
        source = queue[front++];
        printf("%d ", source);

        for (v = 1; v <= n; v++)
        {
            if (a[source][v] == 1 && visited[v] == 0)
            {
                visited[v] = 1;
                queue[rear++] = v;
            }
        }
    }
}
```

```
int main()
{
    int i, j, start, choice;

    printf("Enter number of cities: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix (directed graph):\n");
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }

    printf("Enter starting city (node): ");
    scanf("%d", &start);

    for (i = 1; i <= n; i++)
        visited[i] = 0;

    printf("\nChoose Traversal Method:\n");
    printf("1. DFS\n2. BFS\n");
    scanf("%d", &choice);

    printf("\nNodes reachable from city %d:\n", start);

    if (choice == 1)
        dfs(start);
    else if (choice == 2)
        bfs(start);
    else
        printf("Invalid choice");

    return 0;
}
/*output
Enter number of cities: 4
Enter adjacency matrix (directed graph):
0 1 1 0
0 0 1 0
0 0 0 1
0 0 0 0
```

```
Enter starting city (node): 1

Choose Traversal Method:
1. DFS
2. BFS
1

Nodes reachable from city 1:
1 2 3 4
```