

```

1  #include <stdio.h>
2  int graph[20][20], visited[20], n;
3
4  void BFS(int start) {
5      int queue[20], front = 0, rear = 0;
6      visited[start] = 1;
7      queue[rear++] = start;
8      while (front < rear) {
9          int node = queue[front++];
10         printf("%d ", node);
11         for (int i = 0; i < n; i++) {
12             if (graph[node][i] == 1 && !visited[i]) {
13                 visited[i] = 1;
14                 queue[rear++] = i;
15             }
16         }
17     }
18 }
19 int main() {
20     int start;
21     printf("Enter number of vertices: ");
22     scanf("%d", &n);
23     printf("Enter adjacency matrix:\n");
24     for (int i = 0; i < n; i++) {
25         for (int j = 0; j < n; j++) scanf("%d", &graph[i][j]);
26     }
27     for (int i = 0; i < n; i++) visited[i] = 0;
28     printf("Enter starting vertex: ");
29     scanf("%d", &start);
30     printf("BFS Traversal: ");
31     BFS(start);
32     return 0;
33 }

```

Enter number of vertices: 4

Enter adjacency matrix:

0 1 1 0

1 0 0 1

1 0 0 0

0 1 0 0

Enter starting vertex: 0

BFS Traversal: 0 1 2 3

Lab
a) BFS

```
Code: #include <stdio.h>
```

```
int graph[20][20], visited[20], n;
```

```
void BFS(int start) {
```

```
    int queue[20], front=0, rear=0;
```

```
    visited[start] = 1;
```

```
    queue[rear++] = start;
```

```
    while (front < rear) {
```

```
        int node = queue[front++];
```

```
        printf(" %d", node);
```

```
        for (int i=0; i<n; i++) {
```

```
            if (graph[node][i] == 1 && !visited[i]) {
```

```
                visited[i] = 1;
```

```
                queue[rear++] = i;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main() {
```

```
    int start;
```

```
    printf("Enter number of vertices:");
```

```
    scanf("%d", &n);
```

```
    printf("Enter adjacency matrix:");
```

```
    for (int i=0; i<n; i++) {
```

```
        for (int j=0; j<n; j++) {
```

```
            scanf("%d", &graph[i][j]);
```

```
        }
```

```
    }
```

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

```
    printf("Enter starting vertex:");
```

```
    scanf("%d", &start);
```

```
    printf("BFS Traversal:");
```

```
    BFS(start);
```

```
    return 0;
```

```
}
```

Q/p:-

Enter number of vertices: 4

Enter Adjacency matrix:

0 1 1 0

1 0 0 1

1 0 0 0

0 1 0 0

Enter starting vertex: 0

BFS Traversal: 0 1 2 3