

Lab 9A

The screenshot shows a C IDE interface with a code editor containing a C program for Breadth-First Search (BFS). The code is as follows:

```
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
20 int main(){
21     int start;
22     printf("Enter number of vertices: ");
23     scanf("%d",&n);
24     printf("Enter adjacency matrix:\n");
25     for(int i=0;i<n;i++)
26         for(int j=0;j<n;j++)
27             scanf("%d",&graph[i][j]);
28     for(int i=0;i<n;i++)
29         visited[i]=0;
30     printf("Enter starting vertex: ");
31     scanf("%d",&start);
32     printf("BFS Traversal: ");
33     BFS(start);
34     return 0;
35 }
36 }
```

The code defines a `BFS` function that takes a starting vertex index. It initializes a queue and marks the start vertex as visited. Then it enters a loop where it prints the current node from the queue, marks its neighbors as visited, and adds them to the queue. The `main` function reads the number of vertices and the adjacency matrix from the user, and then calls the `BFS` function with the first vertex as the starting point.

```
2 | int graph[20][20], visited[20], n;
3 |
4 | "C:\Users\BMSCE\Downloads" X + v
5 | Enter number of vertices: 4
6 | Enter adjacency matrix:
7 | 0 1 1 0
8 | 1 0 0 1
9 | 1 0 0 1
10 | 0 1 1 0
11 | Enter starting vertex: 2
12 | BFS Traversal: 2 0 3 1
13 | Process returned 0 (0x0) execution time : 37.006 s
14 | Press any key to continue.
```

The screenshot shows a C code editor interface with the following details:

- Toolbar:** Standard file operations (New, Open, Save, Print, etc.) and search functions.
- Project Explorer:** Shows three projects: "Start here", "LAB 9A.C", and "LAB 9B.C".
- Symbols View:** A sidebar titled "FSymbols" containing a tree view of symbols defined in the current file.
- Code Editor:** The main window displays the following C code:

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

The code implements Depth-First Search (DFS) on an undirected graph represented by an adjacency matrix. It starts at vertex 0 and explores all reachable vertices.

The screenshot shows a terminal window titled "C:\Users\BMSCE\Downloads\". The window displays the following text:

```
Enter number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 0 1
1 0 0 1
0 1 1 0
DFS Traversal starting from vertex 0:
0 1 3 2
Process returned 0 (0x0)   execution time : 33.014 s
Press any key to continue.
```