EXPERIMENT NO. 12 (B-2)

//The Code Is As Follows :-

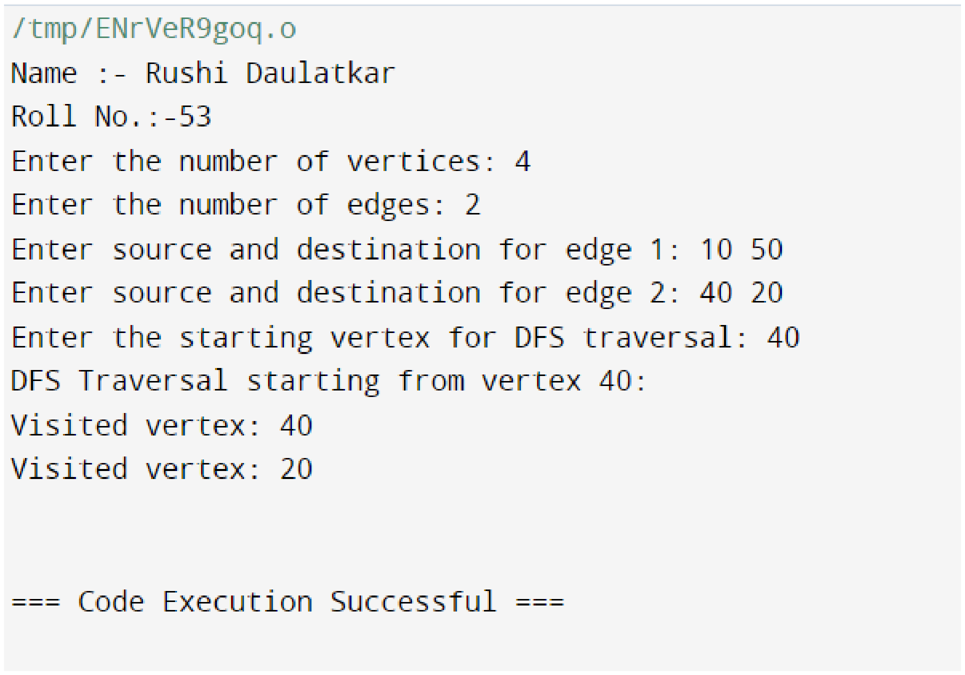
```
#include <stdio.h>

#include <stdlib.h>


// Define structure for a node in the adjacency list

struct Node {

    int vertex;

    struct Node* next;

};


// Define structure for graph

struct Graph {

    int numVertices;

    struct Node** adjLists;

    int* visited;

};


// Function to create a new node with given vertex

struct Node* createNode(int v) {

    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));

    newNode->vertex = v;

    newNode->next = NULL;

    return newNode;
```

```
}

// Function to create a graph with given number of vertices
struct Graph* createGraph(int vertices) {
    struct Graph* graph = (struct Graph*)malloc(sizeof(struct Graph));
    graph->numVertices = vertices;

    graph->adjLists = (struct Node**)malloc(vertices * sizeof(struct Node*));
    graph->visited = (int*)malloc(vertices * sizeof(int));

    for (int i = 0; i < vertices; i++) {
        graph->adjLists[i] = NULL;
        graph->visited[i] = 0;
    }

    return graph;
}

// Function to add an edge between two vertices
void addEdge(struct Graph* graph, int src, int dest) {
    // Add edge from src to dest
    struct Node* newNode = createNode(dest);
    newNode->next = graph->adjLists[src];
    graph->adjLists[src] = newNode;
    // For undirected graph, add edge from dest to src as well
    newNode = createNode(src);
    newNode->next = graph->adjLists[dest];
    graph->adjLists[dest] = newNode;
}

// DFS traversal function
void dfs(struct Graph* graph, int vertex) {
    // Mark the current vertex as visited
    graph->visited[vertex] = 1;
    printf("Visited vertex: %d\n", vertex);

    // Traverse adjacent vertices
    struct Node* adjList = graph->adjLists[vertex];
    while (adjList != NULL) {
        int adjVertex = adjList->vertex;
        if (graph->visited[adjVertex] == 0) {
            dfs(graph, adjVertex); // Recursive call for unvisited adjacent vertices
        }
        adjList = adjList->next;
    }
}

// Main function
int main() {
```

```
printf("Name :- Rushi Daulatkar \n");
printf("Roll No.:-53\n");

int numVertices, numEdges;
printf("Enter the number of vertices: ");
scanf("%d", &numVertices);

// Create graph with given number of vertices
struct Graph* graph = createGraph(numVertices);

printf("Enter the number of edges: ");
scanf("%d", &numEdges);

// Add edges
for (int i = 0; i < numEdges; i++) {
    int src, dest;
    printf("Enter source and destination for edge %d: ", i + 1);
    scanf("%d %d", &src, &dest);
    addEdge(graph, src, dest);
}

int startVertex;
printf("Enter the starting vertex for DFS traversal: ");
scanf("%d", &startVertex);
```

```
    // Perform DFS traversal

    printf("DFS Traversal starting from vertex %d:\n", startVertex);

    dfs(graph, startVertex);


    return 0;

}
```

OUTPUT:-



```
/tmp/ENrVeR9goq.o
Name :- Rushi Daulatkar
Roll No.:-53
Enter the number of vertices: 4
Enter the number of edges: 2
Enter source and destination for edge 1: 10 50
Enter source and destination for edge 2: 40 20
Enter the starting vertex for DFS traversal: 40
DFS Traversal starting from vertex 40:
Visited vertex: 40
Visited vertex: 20


=== Code Execution Successful ===
```