Experiment No.10

Graph Traversal

You are designing a navigation system for a campus with multiple buildings. The system should explore possible paths (routes) using BFS or DFS.

Create a graph using an adjacency matrix and implement Breadth-First Search and Depth-First Search to explore the building connectivity.

```
#include <stdio.h>
#define MAX 10

int graph[MAX][MAX], visited[MAX];
int n; // number of buildings (nodes)

// DFS (recursive)
void dfs(int start) {
   printf("%d ", start);
   visited[start] = 1;
   for (int i = 0; i < n; i++) {
      if (graph[start][i] && !visited[i]) {
         dfs(i);
      }
   }
}

// BFS (using queue)
void bfs(int start) {
   int queue[MAX], 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] && !visited[i]) {
            visited[i] = 1;
            queue[rear++] = i;
         }
      }
   }
}

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

   printf("Enter number of buildings (nodes): ");
```

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

    printf("Enter adjacency matrix (0/1):\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &graph[i][j]);
        }
    }

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

    printf("DFS Traversal: ");
    for (i = 0; i < n; i++) visited[i] = 0;
    dfs(start);

    printf("\nBFS Traversal: ");
    for (i = 0; i < n; i++) visited[i] = 0;
    bfs(start);

    return 0;
}
```