Experiment 21: Graph Traversal using BFS

Aim:
To write a C program to perform Breadth First Search (BFS) traversal of a graph.

Algorithm:

1. Start the program.
2. Represent the graph using adjacency matrix/list.
3. Use a queue to store visited nodes.
4. Mark the starting node as visited.
5. Dequeue a node, print it, enqueue all its unvisited neighbors.
6. Repeat until queue is empty.
7. Stop.

Code:

```
#include <stdio.h>
#define SIZE 10
int queue[SIZE], front = -1, rear = -1;
void enqueue(int v) {
    if (rear == SIZE - 1) return;
    if (front == -1) front = 0;
    queue[++rear] = v;
}

int dequeue() {
    if (front == -1 || front > rear) return -1;
    return queue[front++];
}

void bfs(int adj[SIZE][SIZE], int n, int start) {
    int visited[SIZE] = {0};
    enqueue(start);
    visited[start] = 1;
```

```
    while (front <= rear) {
        int v = dequeue();
        printf("%d ", v);
        for (int i = 0; i < n; i++) {
            if (adj[v][i] && !visited[i]) {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}

int main() {
    int n = 4;
    int adj[SIZE][SIZE] = {
        {0,1,1,0},
        {1,0,0,1},
        {1,0,0,1},
        {0,1,1,0}
    };
    printf("BFS starting from vertex 0: ");
    bfs(adj, n, 0);
    return 0;
}
```

Sample Output:

```
BFS starting from vertex 0: 0 1 2 3

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

Result:
The program successfully traverses a graph using BFS.