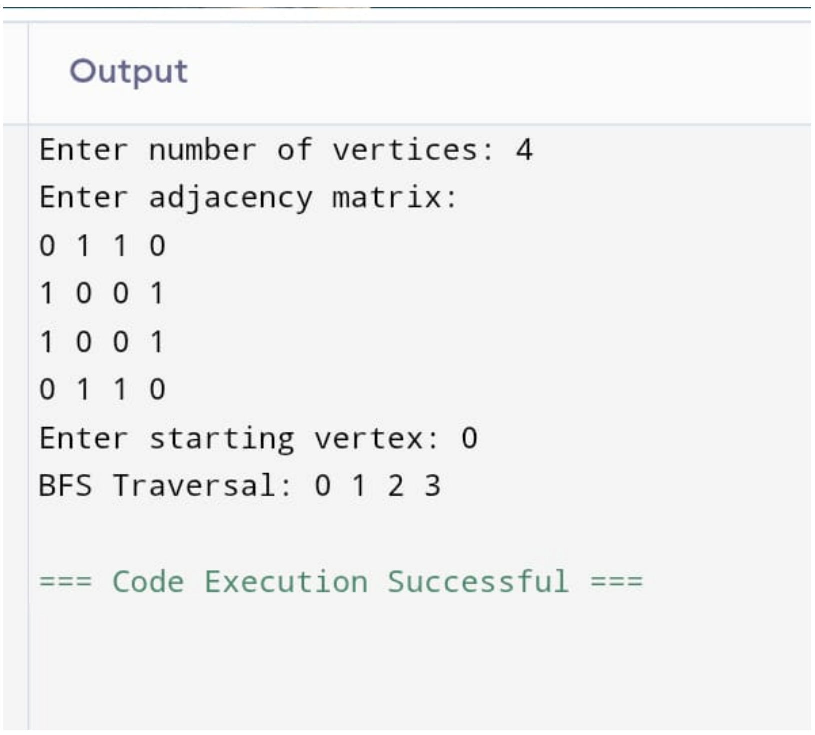
CSA 0317 DATA STRUCTURES

PROGRAM 21

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

int queue[MAX], front = -1, rear = -1;
int visited[MAX];

void enqueue(int vertex) {
    if (rear == MAX - 1)
        return;
    if (front == -1)
        front = 0;
    queue[++rear] = vertex;
}

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

void BFS(int adj[MAX][MAX], int n, int start) {
    int i, vertex;
    for (i = 0; i < n; i++)
        visited[i] = 0;

    enqueue(start);
    visited[start] = 1;
    while ((vertex = dequeue()) != -1) {
        printf("%d ", vertex);
        for (i = 0; i < n; i++) {
            if (adj[vertex][i] == 1 && !visited[i]) {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}

int main() {
    int n, adj[MAX][MAX], i, j, start;

    printf("Enter number of vertices: ");
    scanf("%d", &n);

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

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

    printf("BFS Traversal: ");
    BFS(adj, n, start);
    return 0;
}
```

Output:


Output

```
Enter number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 0 1
1 0 0 1
0 1 1 0
Enter starting vertex: 0
BFS Traversal: 0 1 2 3

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