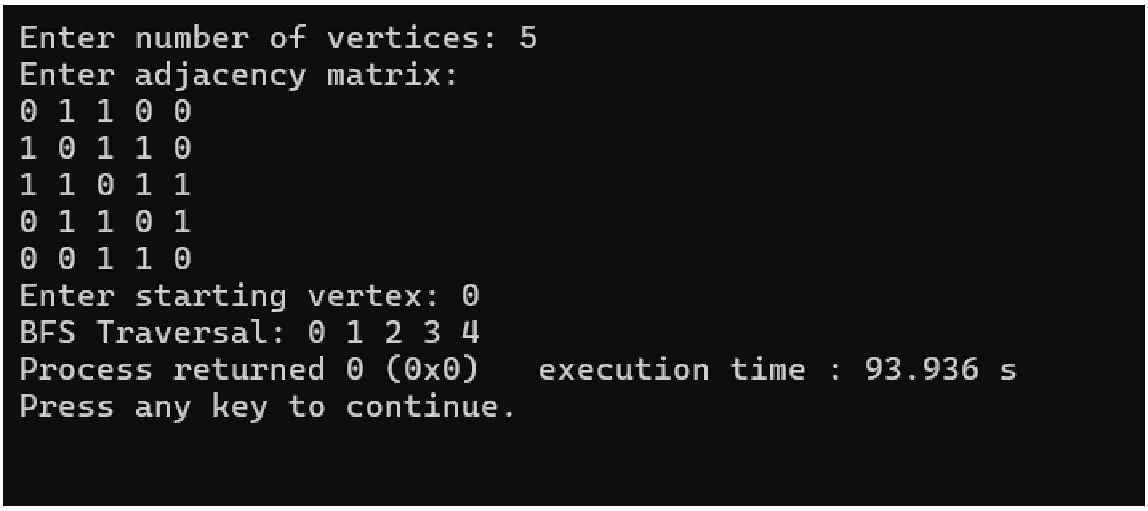
a) Write a program to traverse a graph using BFS method.

```
#include <stdio.h>
int queue[50], front = -1, rear = -1;
int visited[50], n, graph[50][50];
void enqueue(int v) {
    if (rear == 49)
        return;
    if (front == -1)
        front = 0;
    queue[++rear] = v;
}
int dequeue() {
    return queue[front++];
}
void bfs(int start) {
    int i, v;
    enqueue(start);
    visited[start] = 1;
    printf("BFS Traversal: ");
    while (front <= rear) {
        v = dequeue();
        printf("%d ", v);
        for (i = 0; i < n; i++) {
            if (graph[v][i] == 1 && visited[i] == 0) {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}
int main() {
    int 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", &graph[i][j]);
        }
        visited[i] = 0;
    }
    printf("Enter starting vertex: ");
    scanf("%d", &start);
    bfs(start);
    return 0;
}
```



```
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 1 1 0
1 1 0 1 1
0 1 1 0 1
0 0 1 1 0
Enter starting vertex: 0
BFS Traversal: 0 1 2 3 4
Process returned 0 (0x0)   execution time : 93.936 s
Press any key to continue.
```