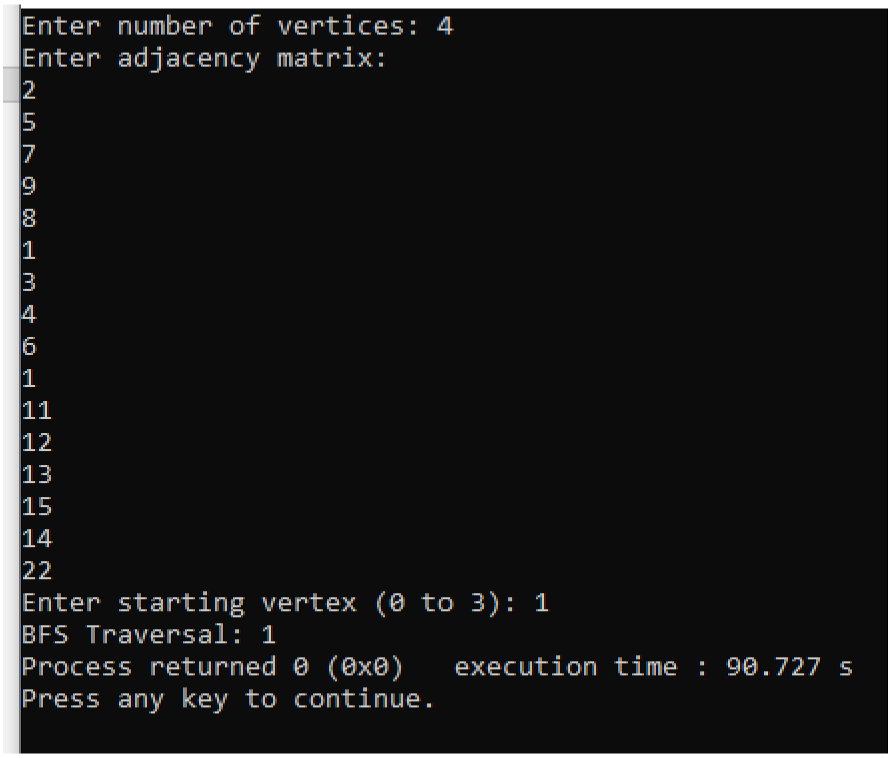
9a. Write a program to traverse a graph using BFS method.

```
#include <stdio.h>

#define MAX 10

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

/* Function to insert into queue */
void enqueue(int v) {
    if (rear == MAX - 1)
        return;
    if (front == -1)
        front = 0;
    queue[++rear] = v;
}

/* Function to delete from queue */
int dequeue() {
    return queue[front++];
}

/* BFS function */
void bfs(int start, int n) {
    int i;
    printf("BFS Traversal: ");

    enqueue(start);
    visited[start] = 1;
    while (front <= rear) {
        int v = dequeue();
        printf("%d ", v);

        for (i = 0; i < n; i++) {
            if (graph[v][i] == 1 && !visited[i]) {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}

/* Main function */
int main() {
    int n, 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 (0 to %d): ", n - 1);
    scanf("%d", &start);

    bfs(start, n);
    return 0;

}
```

OUTPUT:



```
Enter number of vertices: 4
Enter adjacency matrix:
2
5
7
9
8
1
3
4
6
1
11
12
13
15
14
22
Enter starting vertex (0 to 3): 1
BFS Traversal: 1
Process returned 0 (0x0)   execution time : 90.727 s
Press any key to continue.
```