

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

```
#include <stdio.h>

#define MAX 10

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

/* 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()
{
    if (front > rear)
        return -1;
    return queue[front++];
}

/* BFS function */
void BFS(int graph[MAX][MAX], int n, 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])
            {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}
```

```

    }
}
}

int main()
{
    int graph[MAX][MAX], 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]);

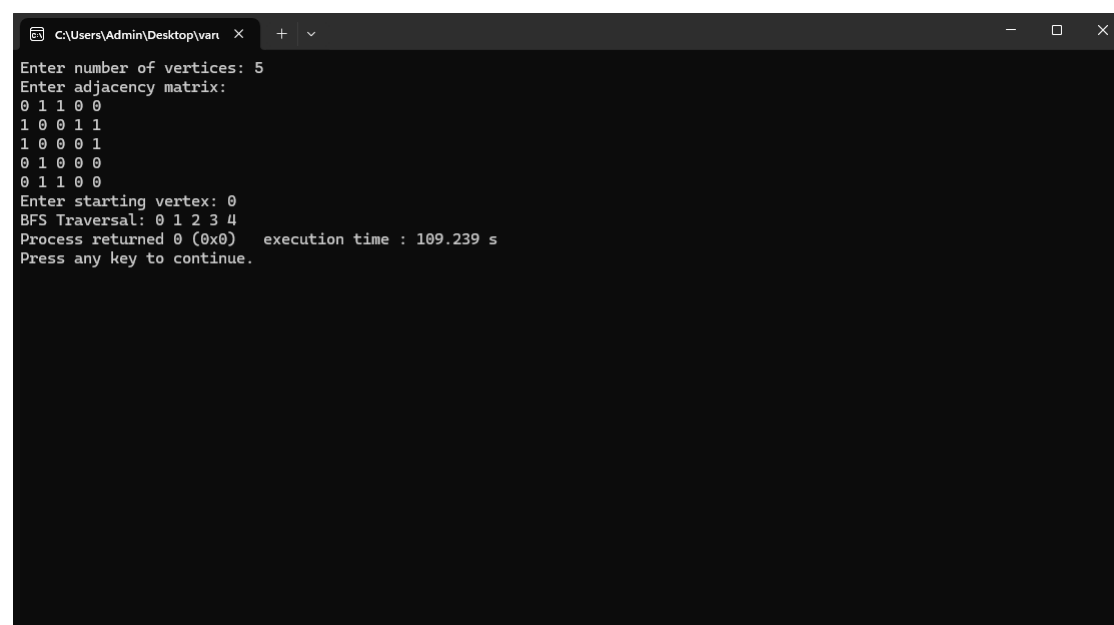
    printf("Enter starting vertex: ");
    scanf("%d", &start);

    BFS(graph, n, start);

    return 0;
}

```

Output:



```

C:\Users\Admin\Desktop\van x + v
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 0 1 1
1 0 0 0 1
0 1 0 0 0
0 1 1 0 0
Enter starting vertex: 0
BFS Traversal: 0 1 2 3 4
Process returned 0 (0x0)    execution time : 109.239 s
Press any key to continue.

```