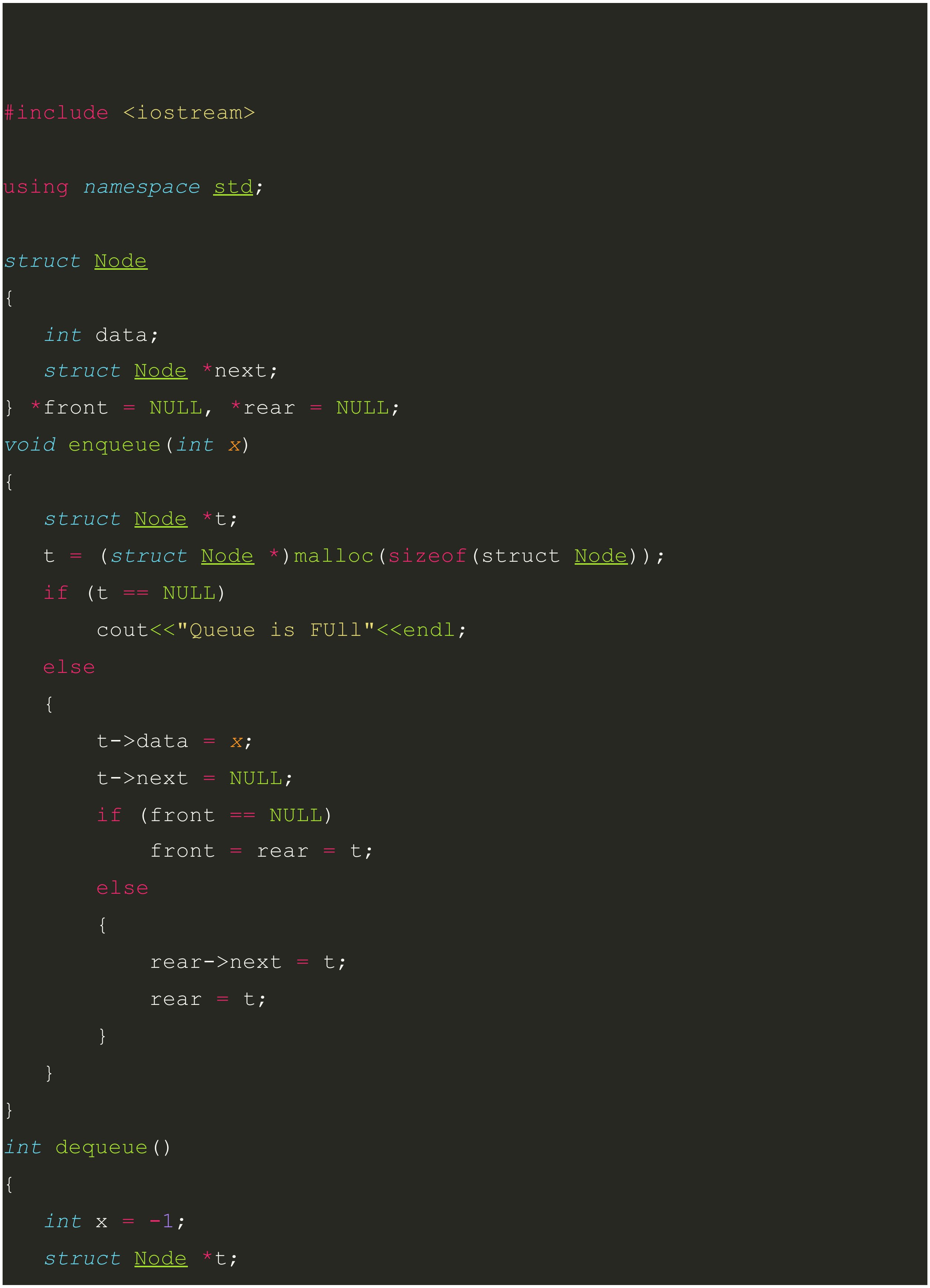
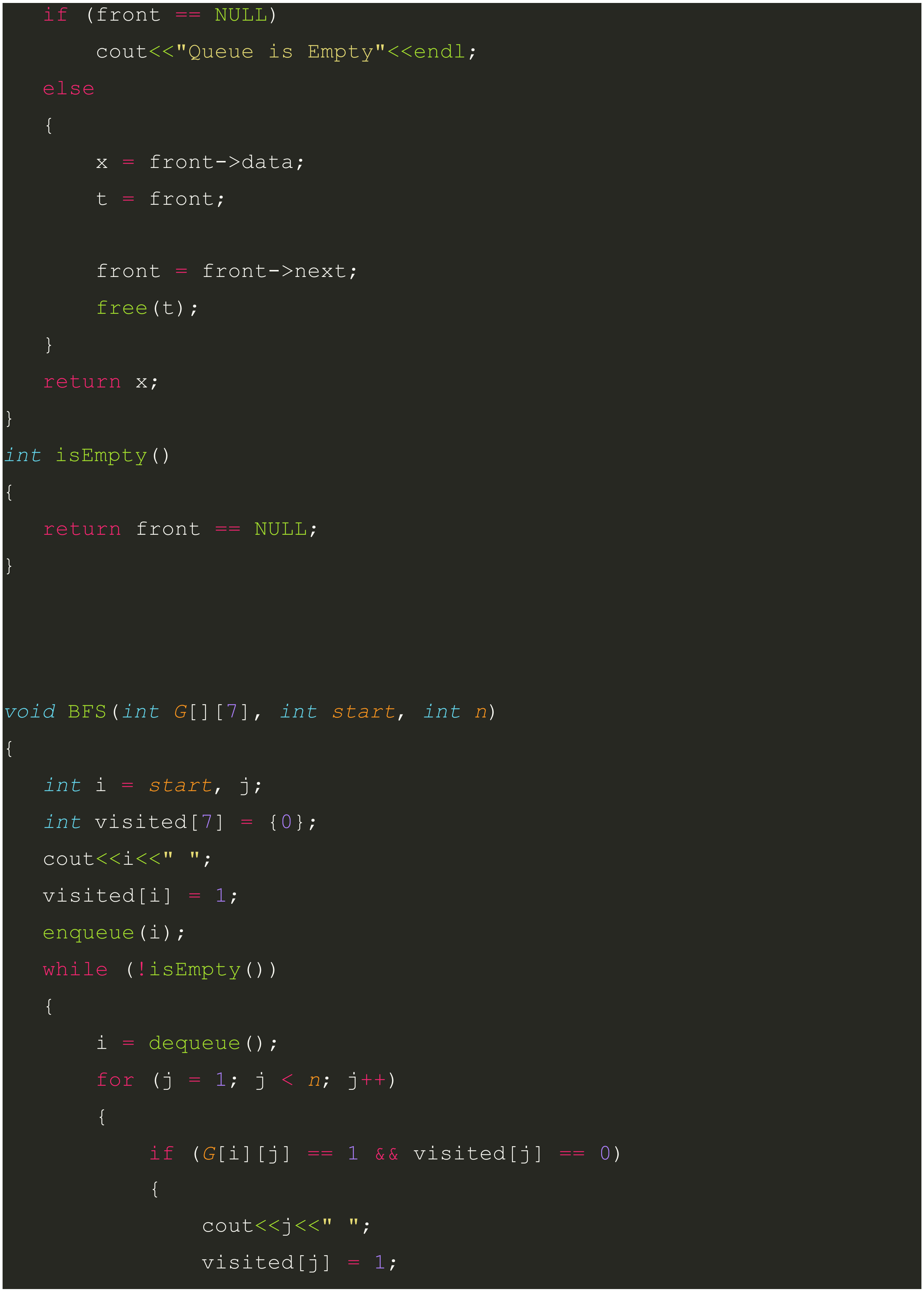
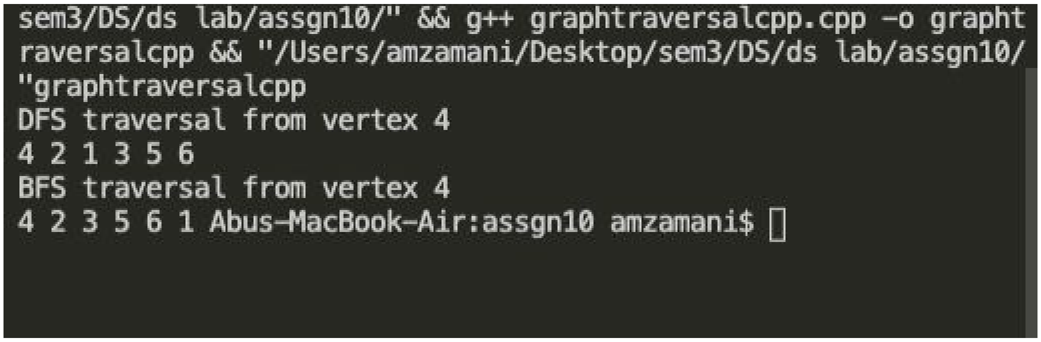


```
#include <iostream>

using namespace std;

struct Node
{
    int data;
    struct Node *next;
} *front = NULL, *rear = NULL;
void enqueue(int x)
{
    struct Node *t;
    t = (struct Node *)malloc(sizeof(struct Node));
    if (t == NULL)
        cout<<"Queue is FUll"<<endl;
    else
    {
        t->data = x;
        t->next = NULL;
        if (front == NULL)
            front = rear = t;
        else
        {
            rear->next = t;
            rear = t;
        }
    }
}
int dequeue()
{
    int x = -1;
    struct Node *t;
```



```
    if (front == NULL)
        cout<<"Queue is Empty"<<endl;
    else
    {
        x = front->data;
        t = front;

        front = front->next;
        free(t);
    }
    return x;
}
int isEmpty()
{
    return front == NULL;
}



void BFS(int G[][7], int start, int n)
{
    int i = start, j;
    int visited[7] = {0};
    cout<<i<<" ";
    visited[i] = 1;
    enqueue(i);
    while (!isEmpty())
    {
        i = dequeue();
        for (j = 1; j < n; j++)
        {
            if (G[i][j] == 1 && visited[j] == 0)
            {
                cout<<j<<" ";
                visited[j] = 1;
                enqueue(j);
            }
        }
    }
}

void DFS(int G[][7], int start, int n)
{

    static int visited[7] = {0};
    int j;
    if (visited[start] == 0)
    {
        cout<<start<<" ";
        visited[start] = 1;
        for (j = 1; j < n; j++)
        {
            if (G[start][j] == 1 && visited[j] == 0)
                DFS(G, j, n);
        }
    }
}

int main()
{
    int G[7][7] = {{0, 0, 0, 0, 0, 0, 0},
                  {0, 0, 1, 1, 0, 0, 0},
                  {0, 1, 0, 0, 1, 0, 0},
                  {0, 1, 0, 0, 1, 0, 0},
                  {0, 0, 1, 1, 0, 1, 1},
                  {0, 0, 0, 0, 1, 0, 0},
                  {0, 0, 0, 0, 1, 0, 0}};
    cout<<"DFS traversal from vertex 4"<<endl;
    DFS(G, 4, 7);
    cout<<endl;
    cout<<"BFS traversal from vertex 4"<<endl;

    BFS(G, 4, 7);
    cout<<endl;
    return 0;
}
```

OUTPUT :



```
sem3/DS/ds lab/assgn10/" && g++ graphtraversalcpp.cpp -o grapht
raversalcpp && "/Users/amzamani/Desktop/sem3/DS/ds lab/assgn10/
"graphtraversalcpp
DFS traversal from vertex 4
4 2 1 3 5 6
BFS traversal from vertex 4
4 2 3 5 6 1 Abus-MacBook-Air:assgn10 amzamani$ 
```