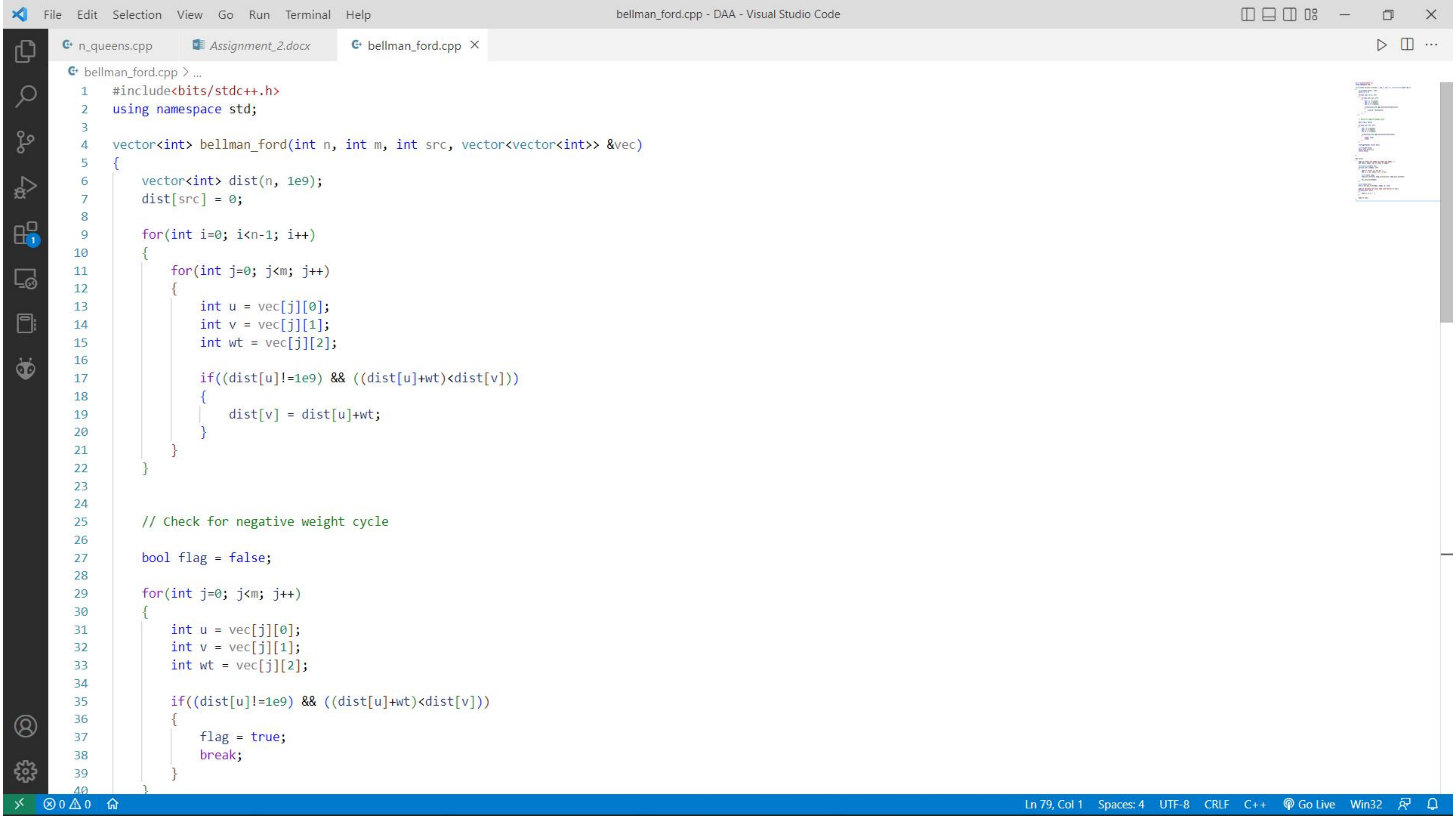
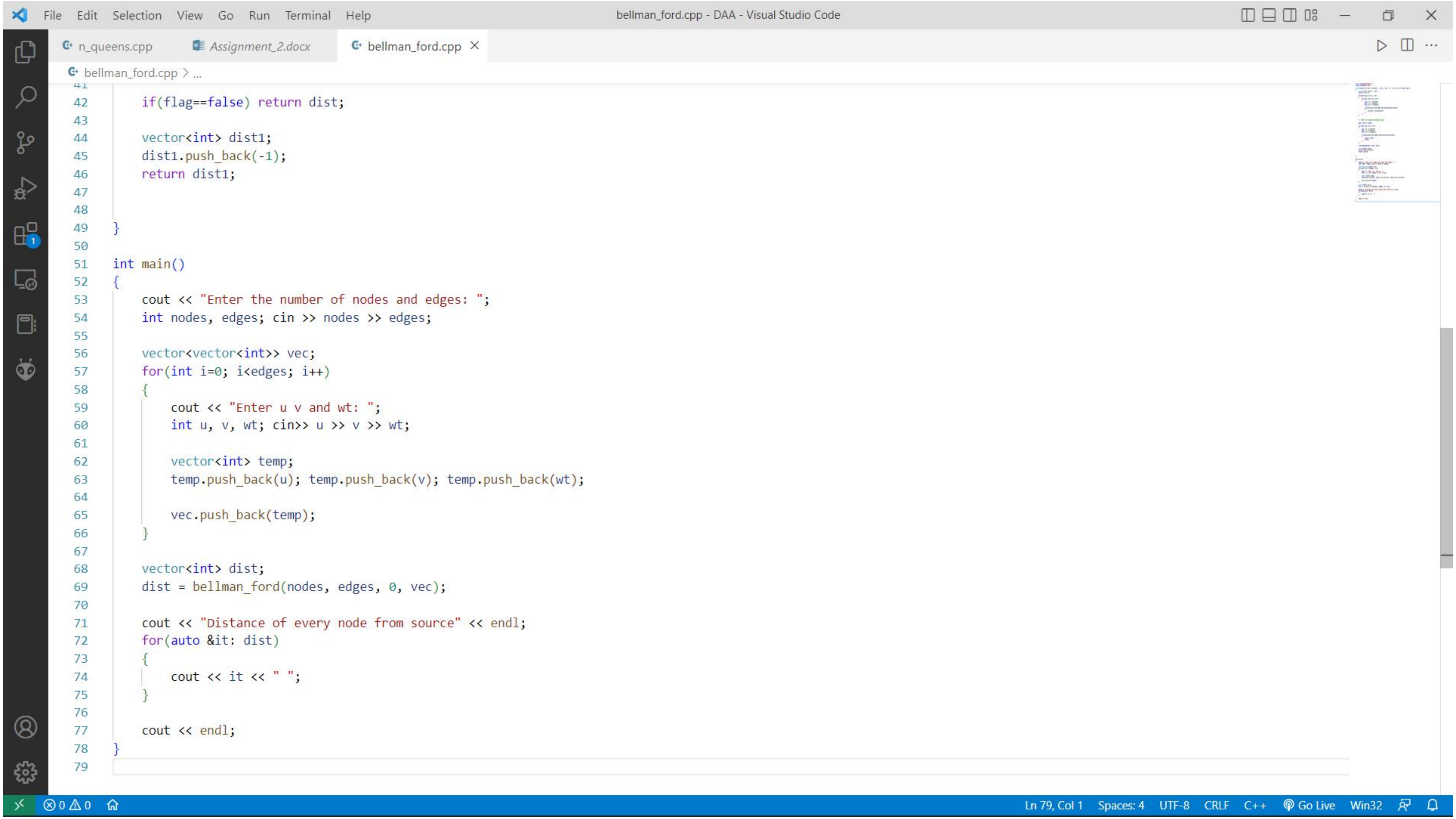


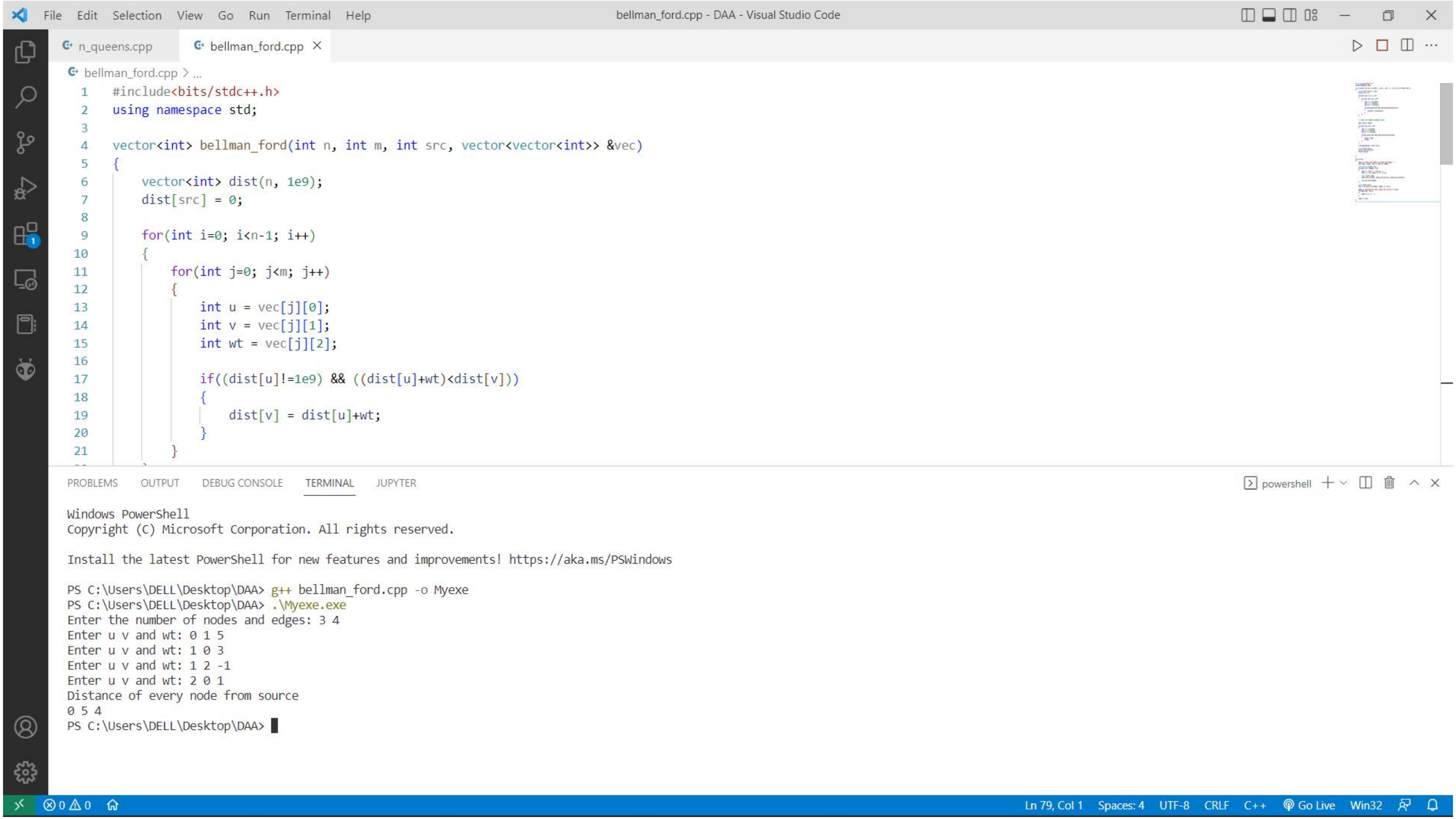
File Edit Selection View Go Run Terminal Help
bellman_ford.cpp - DAA - Visual Studio Code
n_queens.cpp
Assignment_2.docx
bellman_ford.cpp
bellman_ford.cpp > ...

```cpp
#include<bits/stdc++.h>
using namespace std;

vector<int> bellman_ford(int n, int m, int src, vector<vector<int>> &vec)
{
    vector<int> dist(n, 1e9);
    dist[src] = 0;

    for(int i=0; i<n-1; i++)
    {
        for(int j=0; j<m; j++)
        {
            int u = vec[j][0];
            int v = vec[j][1];
            int wt = vec[j][2];

            if((dist[u]!=1e9) && ((dist[u]+wt)<dist[v]))
            {
                dist[v] = dist[u]+wt;
            }
        }
    }


    // Check for negative weight cycle

    bool flag = false;

    for(int j=0; j<m; j++)
    {
        int u = vec[j][0];
        int v = vec[j][1];
        int wt = vec[j][2];

        if((dist[u]!=1e9) && ((dist[u]+wt)<dist[v]))
        {
            flag = true;
            break;
        }
```

⊗ 0 ⚠ 0
Ln 79, Col 1 Spaces: 4 UTF-8 CRLF C++ Go Live Win32



```cpp
    if(flag==false) return dist;

    vector<int> dist1;
    dist1.push_back(-1);
    return dist1;


}

int main()
{
    cout << "Enter the number of nodes and edges: ";
    int nodes, edges; cin >> nodes >> edges;

    vector<vector<int>> vec;
    for(int i=0; i<edges; i++)
    {
        cout << "Enter u v and wt: ";
        int u, v, wt; cin>> u >> v >> wt;

        vector<int> temp;
        temp.push_back(u); temp.push_back(v); temp.push_back(wt);

        vec.push_back(temp);
    }

    vector<int> dist;
    dist = bellman_ford(nodes, edges, 0, vec);

    cout << "Distance of every node from source" << endl;
    for(auto &it: dist)
    {
        cout << it << " ";
    }

    cout << endl;
}

```



```cpp
#include<bits/stdc++.h>
using namespace std;

vector<int> bellman_ford(int n, int m, int src, vector<vector<int>> &vec)
{
    vector<int> dist(n, 1e9);
    dist[src] = 0;

    for(int i=0; i<n-1; i++)
    {
        for(int j=0; j<m; j++)
        {
            int u = vec[j][0];
            int v = vec[j][1];
            int wt = vec[j][2];

            if((dist[u]!=1e9) && ((dist[u]+wt)<dist[v]))
            {
                dist[v] = dist[u]+wt;
            }
        }
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\DELL\Desktop\DAA> g++ bellman_ford.cpp -o Myexe
PS C:\Users\DELL\Desktop\DAA> .\Myexe.exe
Enter the number of nodes and edges: 3 4
Enter u v and wt: 0 1 5
Enter u v and wt: 1 0 3
Enter u v and wt: 1 2 -1
Enter u v and wt: 2 0 1
Distance of every node from source
0 5 4
PS C:\Users\DELL\Desktop\DAA>
```