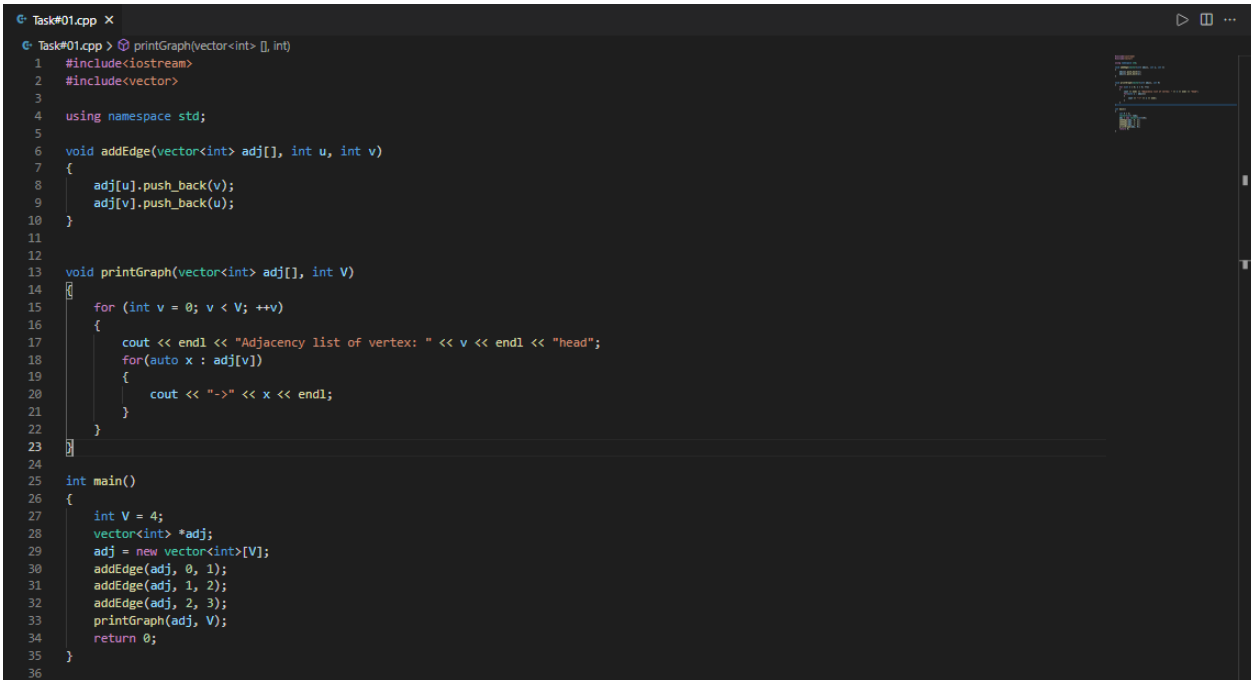
LAB TASK # 12

MOHAMMAD BASIL ALI KHAN

20K-0477

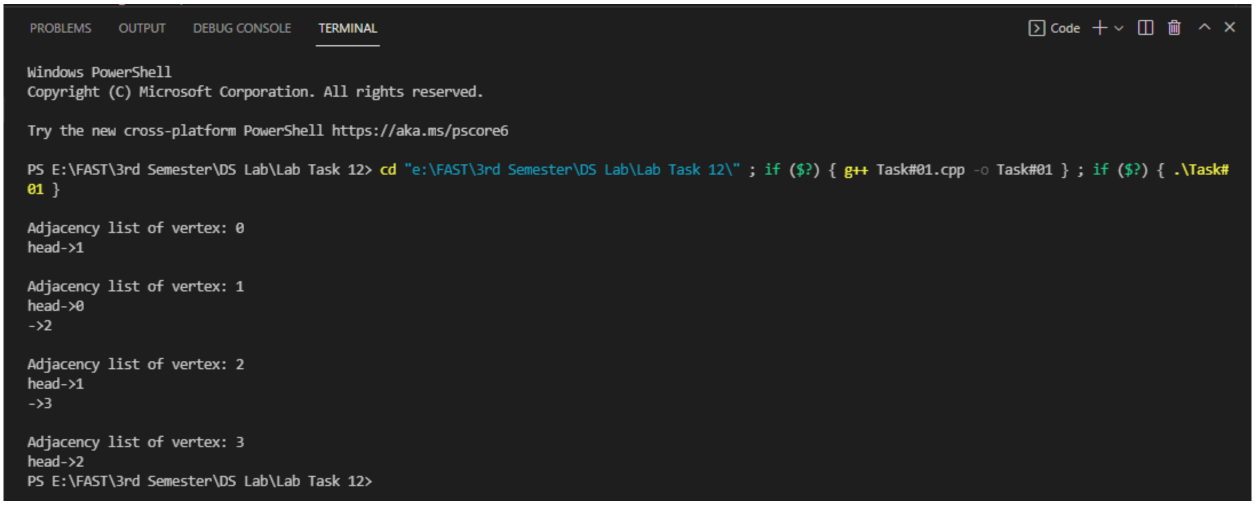
Task#01

- **Code**



```cpp
#include<iostream>
#include<vector>

using namespace std;

void addEdge(vector<int> adj[], int u, int v)
{
    adj[u].push_back(v);
    adj[v].push_back(u);
}


void printGraph(vector<int> adj[], int V)
{
    for (int v = 0; v < V; ++v)
    {
        cout << endl << "Adjacency list of vertex: " << v << endl << "head";
        for(auto x : adj[v])
        {
            cout << "->" << x << endl;
        }
    }
}

int main()
{
    int V = 4;
    vector<int> *adj;
    adj = new vector<int>[V];
    addEdge(adj, 0, 1);
    addEdge(adj, 1, 2);
    addEdge(adj, 2, 3);
    printGraph(adj, V);
    return 0;
}
```

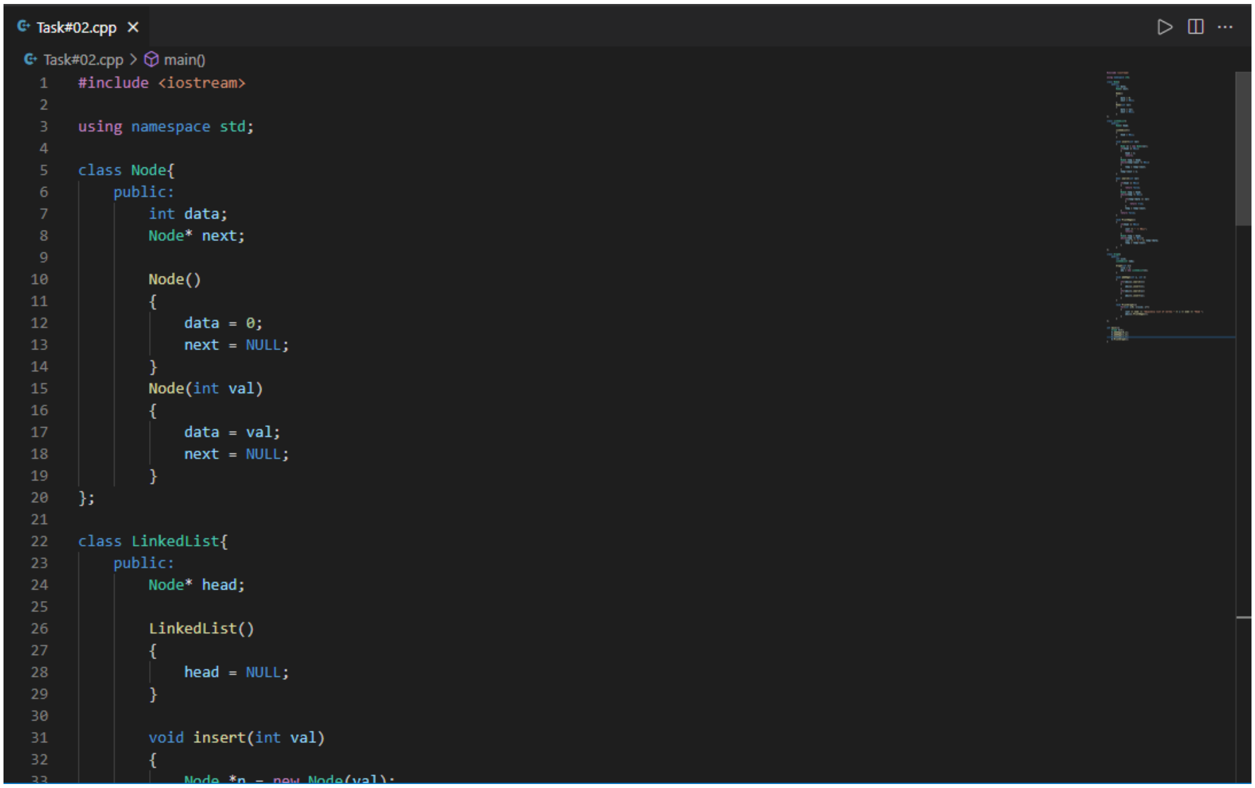
- **Output**



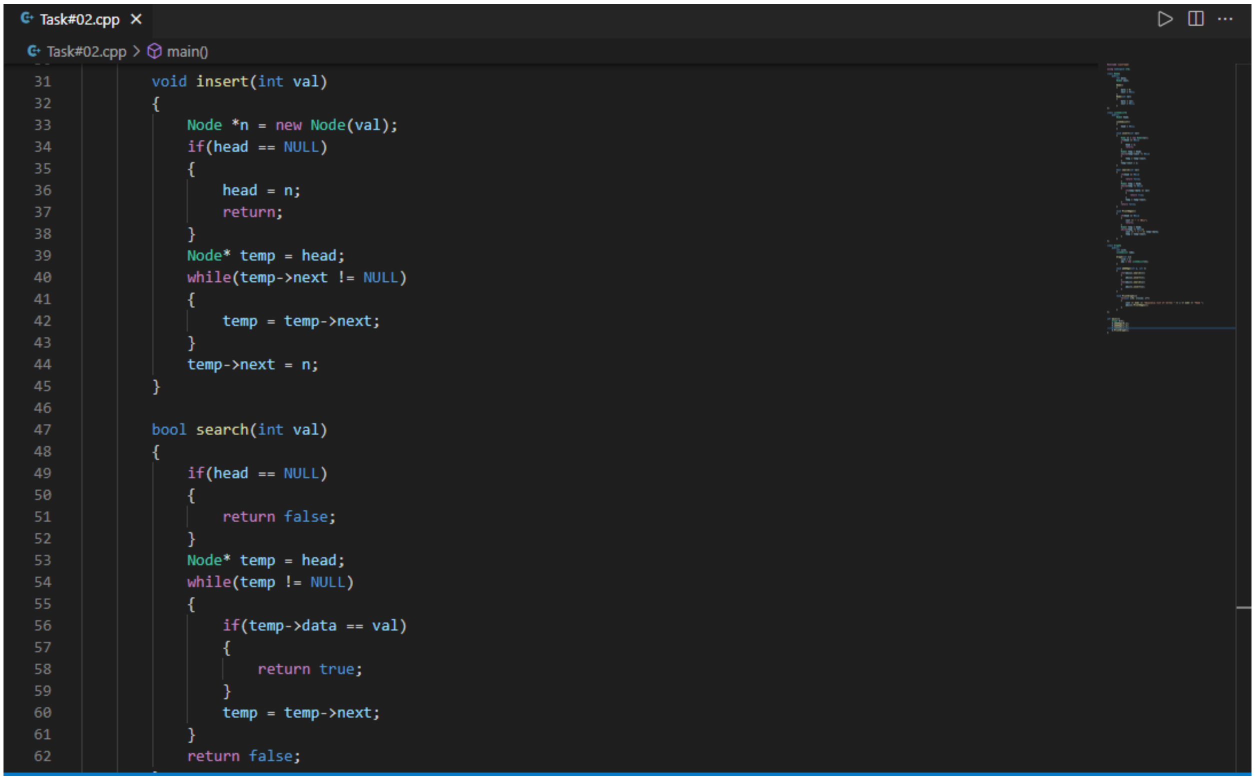
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#01.cpp -o Task#01 } ; if ($?) { .\Task#
01 }

Adjacency list of vertex: 0
head->1

Adjacency list of vertex: 1
head->0
->2

Adjacency list of vertex: 2
head->1
->3

Adjacency list of vertex: 3
head->2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

Task#02:

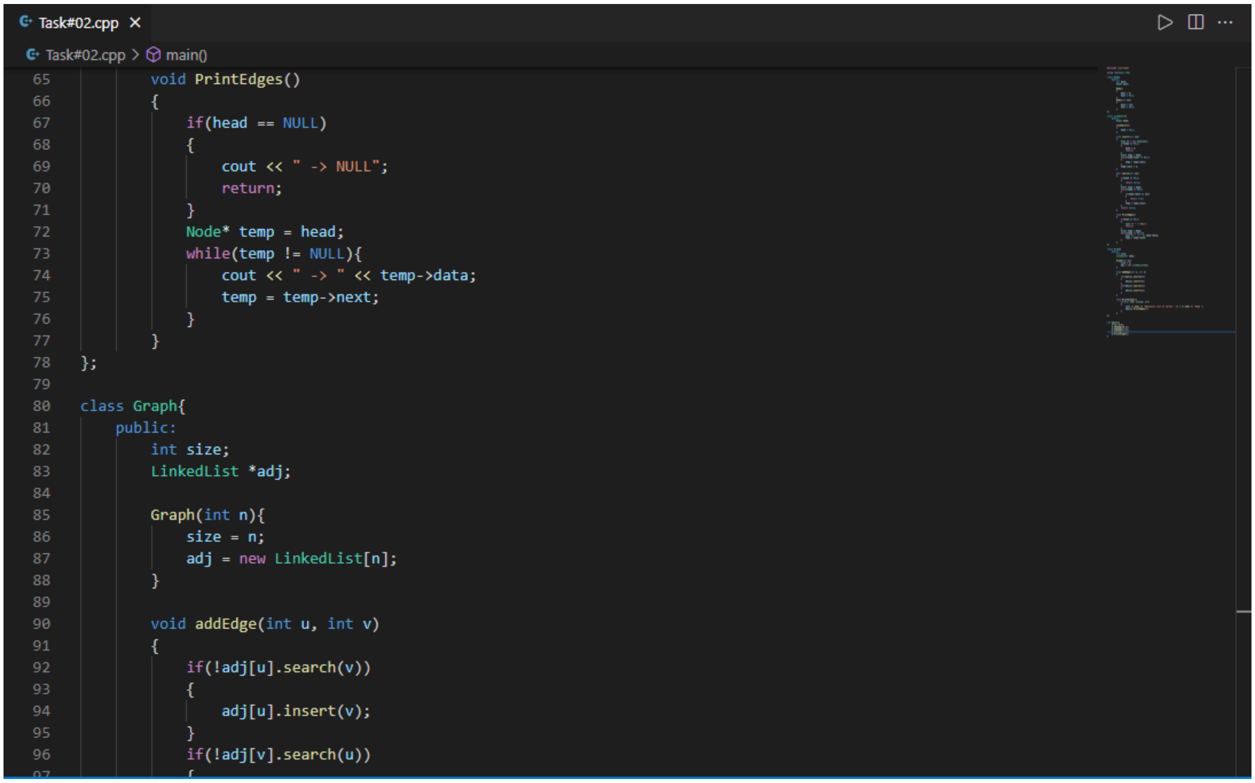
- **Code**



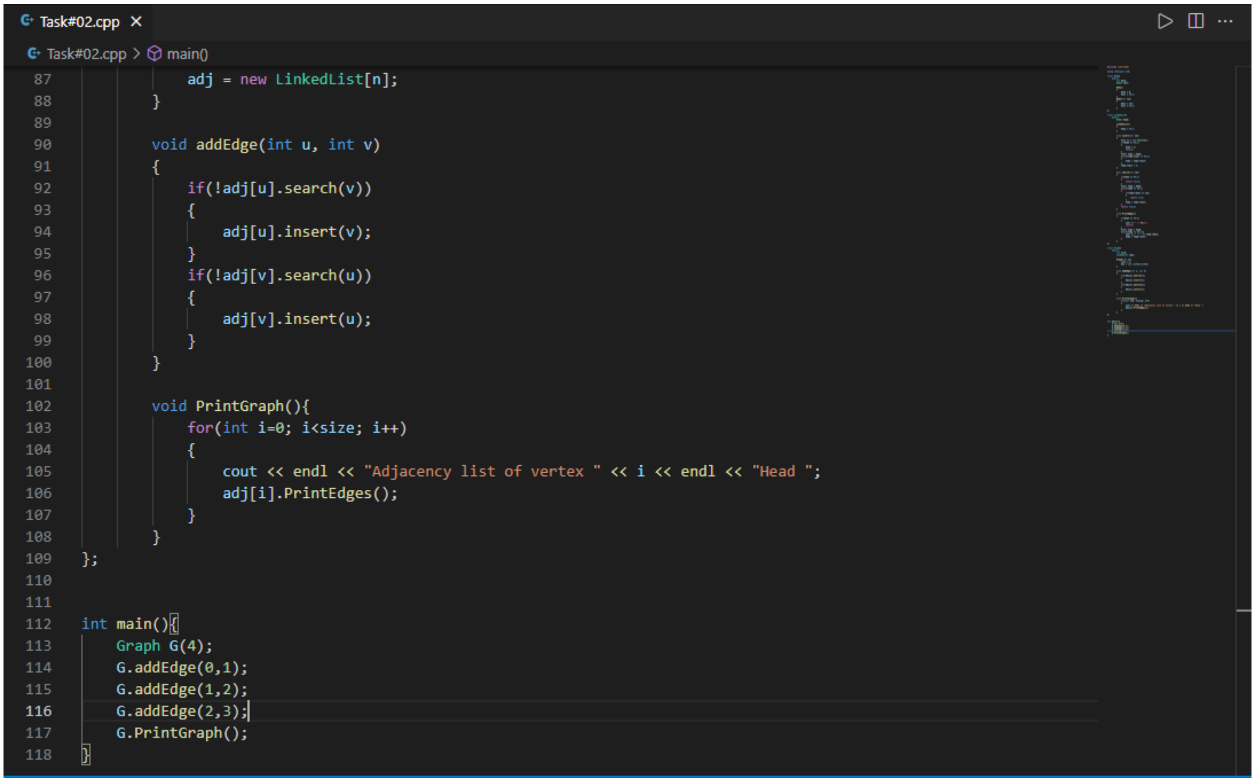
```cpp
#include <iostream>

using namespace std;

class Node{
    public:
        int data;
        Node* next;

        Node()
        {
            data = 0;
            next = NULL;
        }
        Node(int val)
        {
            data = val;
            next = NULL;
        }
};

class LinkedList{
    public:
        Node* head;

        LinkedList()
        {
            head = NULL;
        }

        void insert(int val)
        {
            Node *n = new Node(val);
```



```cpp
        void insert(int val)
        {
            Node *n = new Node(val);
            if(head == NULL)
            {
                head = n;
                return;
            }
            Node* temp = head;
            while(temp->next != NULL)
            {
                temp = temp->next;
            }
            temp->next = n;
        }

        bool search(int val)
        {
            if(head == NULL)
            {
                return false;
            }
            Node* temp = head;
            while(temp != NULL)
            {
                if(temp->data == val)
                {
                    return true;
                }
                temp = temp->next;
            }
            return false;
```

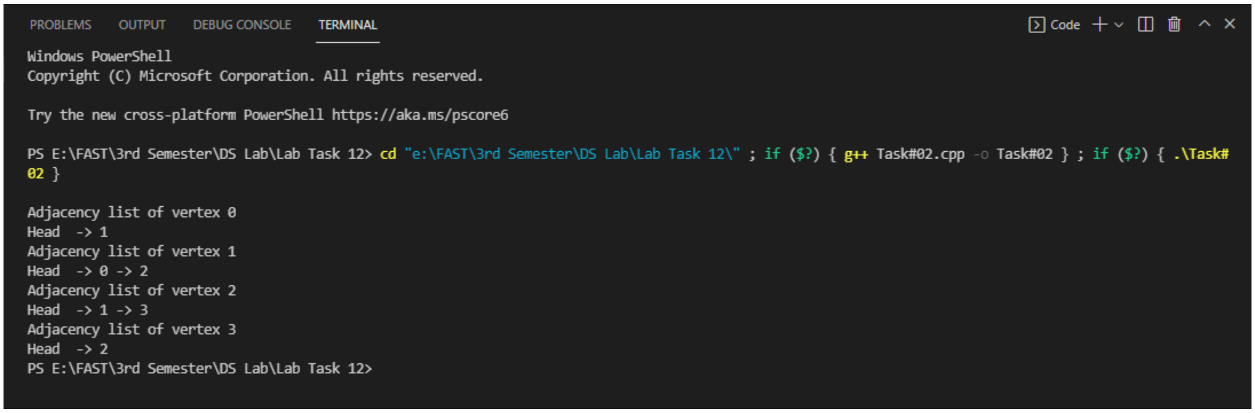


```
        void PrintEdges()
        {
            if(head == NULL)
            {
                cout << " -> NULL";
                return;
            }
            Node* temp = head;
            while(temp != NULL){
                cout << " -> " << temp->data;
                temp = temp->next;
            }
        }
};

class Graph{
    public:
        int size;
        LinkedList *adj;

        Graph(int n){
            size = n;
            adj = new LinkedList[n];
        }

        void addEdge(int u, int v)
        {
            if(!adj[u].search(v))
            {
                adj[u].insert(v);
            }
            if(!adj[v].search(u))
            {
```



```
            adj = new LinkedList[n];
        }

        void addEdge(int u, int v)
        {
            if(!adj[u].search(v))
            {
                adj[u].insert(v);
            }
            if(!adj[v].search(u))
            {
                adj[v].insert(u);
            }
        }

        void PrintGraph(){
            for(int i=0; i<size; i++)
            {
                cout << endl << "Adjacency list of vertex " << i << endl << "Head ";
                adj[i].PrintEdges();
            }
        }
};


int main(){
    Graph G(4);
    G.addEdge(0,1);
    G.addEdge(1,2);
    G.addEdge(2,3);
    G.PrintGraph();
}
```

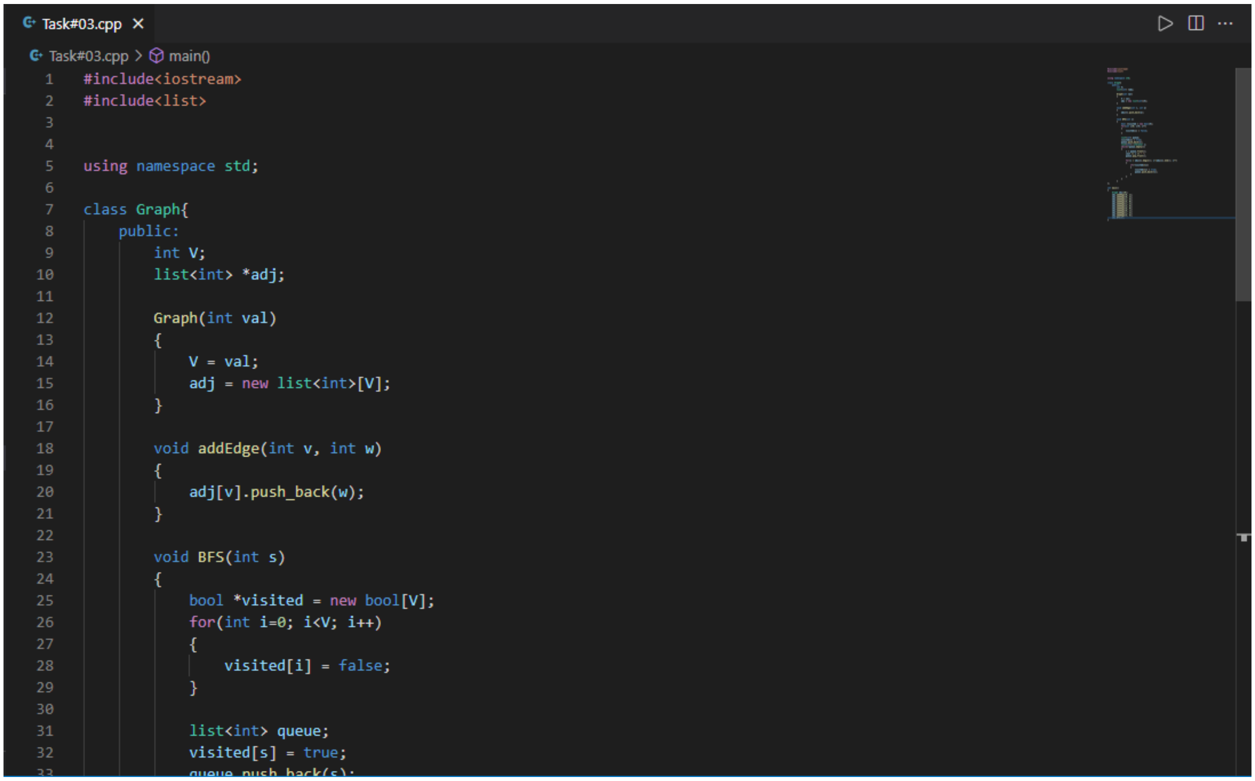
- **Output**



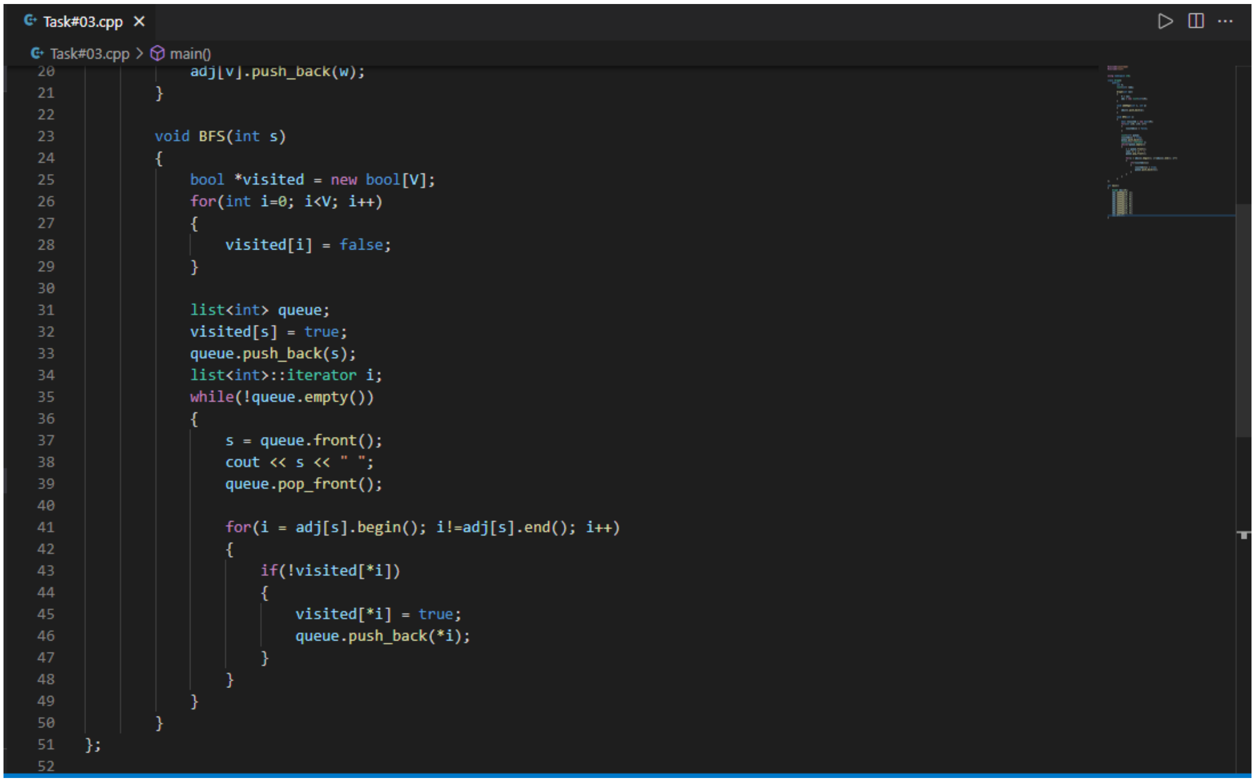
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#02.cpp -o Task#02 } ; if ($?) { .\Task#
02 }

Adjacency list of vertex 0
Head  -> 1
Adjacency list of vertex 1
Head  -> 0 -> 2
Adjacency list of vertex 2
Head  -> 1 -> 3
Adjacency list of vertex 3
Head  -> 2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

Task#03:

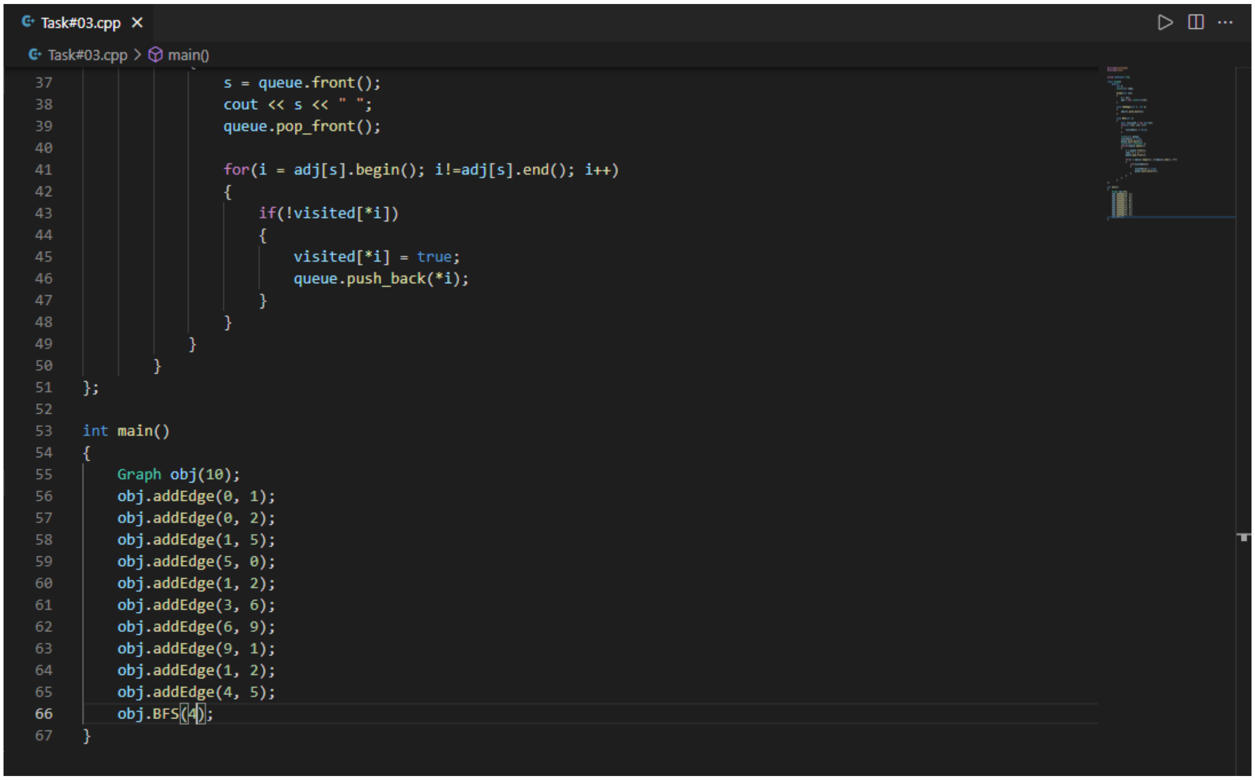
- **Code**



```cpp
#include<iostream>
#include<list>


using namespace std;

class Graph{
    public:
        int V;
        list<int> *adj;

        Graph(int val)
        {
            V = val;
            adj = new list<int>[V];
        }

        void addEdge(int v, int w)
        {
            adj[v].push_back(w);
        }

        void BFS(int s)
        {
            bool *visited = new bool[V];
            for(int i=0; i<V; i++)
            {
                visited[i] = false;
            }

            list<int> queue;
            visited[s] = true;
            queue.push_back(s);
```



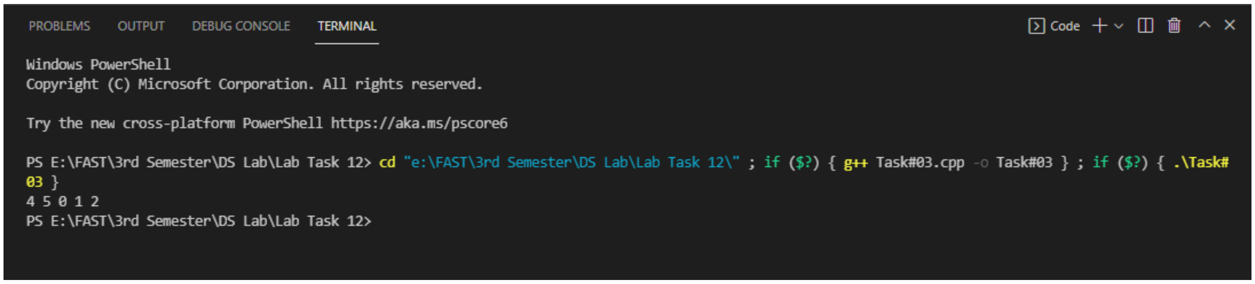
```cpp
            adj[v].push_back(w);
        }

        void BFS(int s)
        {
            bool *visited = new bool[V];
            for(int i=0; i<V; i++)
            {
                visited[i] = false;
            }

            list<int> queue;
            visited[s] = true;
            queue.push_back(s);
            list<int>::iterator i;
            while(!queue.empty())
            {
                s = queue.front();
                cout << s << " ";
                queue.pop_front();

                for(i = adj[s].begin(); i!=adj[s].end(); i++)
                {
                    if(!visited[*i])
                    {
                        visited[*i] = true;
                        queue.push_back(*i);
                    }
                }
            }
        }
};

```



```cpp
                s = queue.front();
                cout << s << " ";
                queue.pop_front();

                for(i = adj[s].begin(); i!=adj[s].end(); i++)
                {
                    if(!visited[*i])
                    {
                        visited[*i] = true;
                        queue.push_back(*i);
                    }
                }
            }
        }
};
```

```cpp

int main()
{
    Graph obj(10);
    obj.addEdge(0, 1);
    obj.addEdge(0, 2);
    obj.addEdge(1, 5);
    obj.addEdge(5, 0);
    obj.addEdge(1, 2);
    obj.addEdge(3, 6);
    obj.addEdge(6, 9);
    obj.addEdge(9, 1);
    obj.addEdge(1, 2);
    obj.addEdge(4, 5);
    obj.BFS(4);
}
```

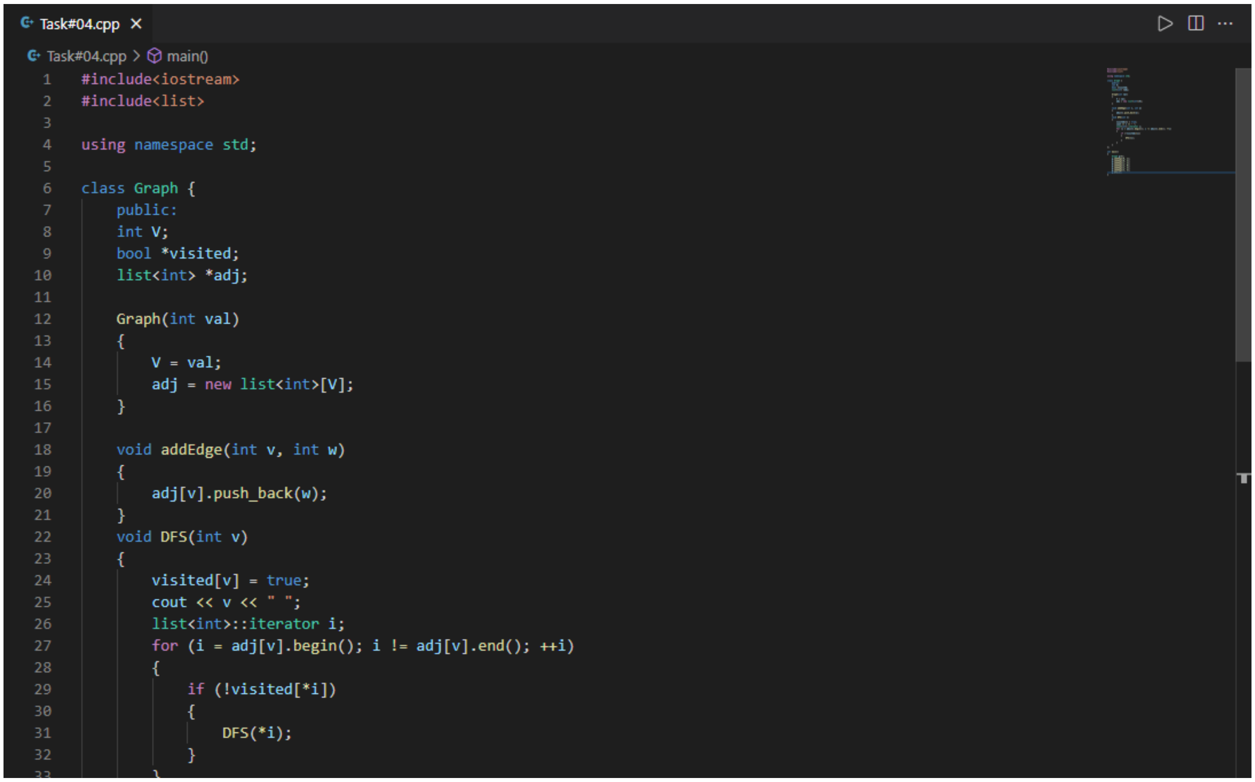
- **Output**



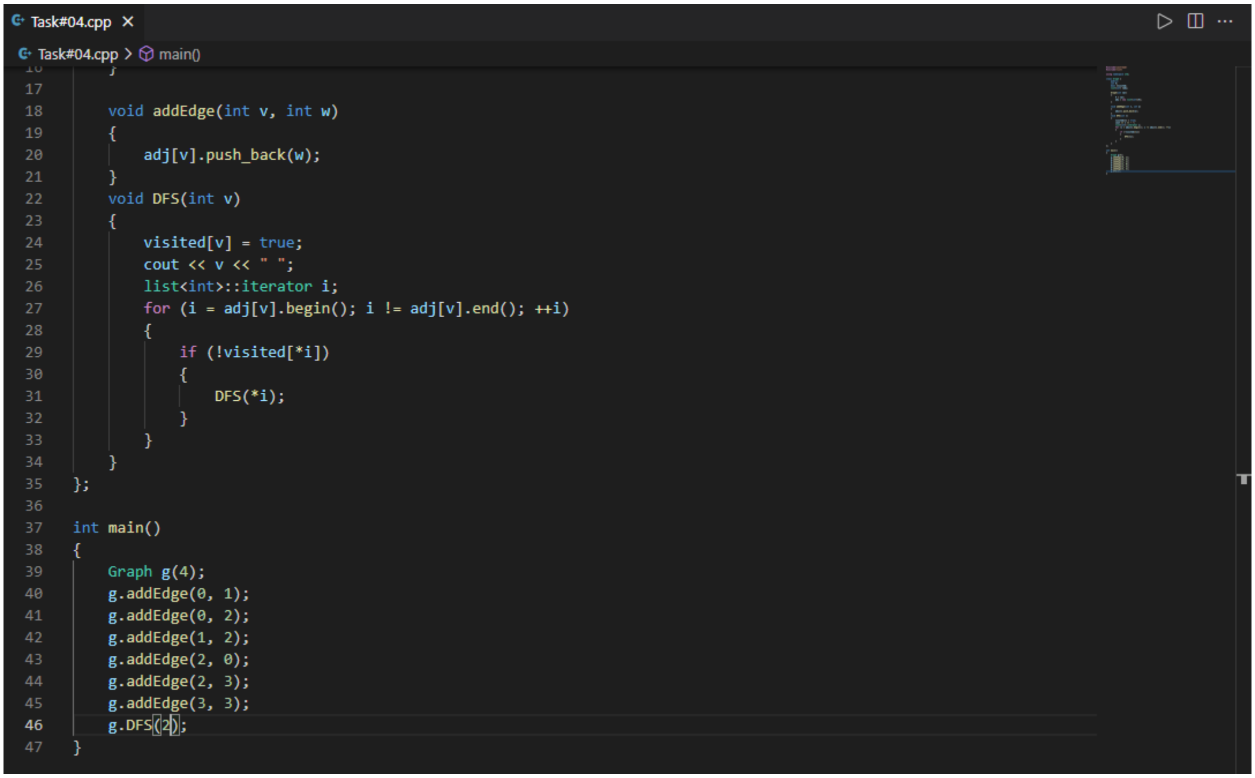
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#03.cpp -o Task#03 } ; if ($?) { .\Task#
03 }
4 5 0 1 2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

Task#04:

- **Code**

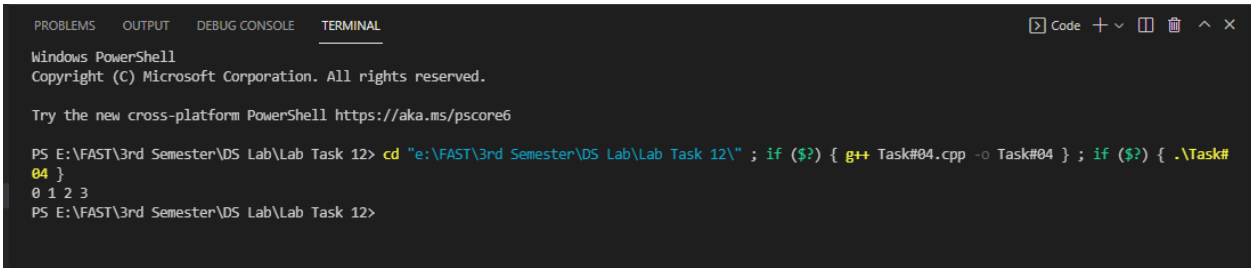


```cpp
#include<iostream>
#include<list>

using namespace std;

class Graph {
    public:
    int V;
    bool *visited;
    list<int> *adj;

    Graph(int val)
    {
        V = val;
        adj = new list<int>[V];
    }

    void addEdge(int v, int w)
    {
        adj[v].push_back(w);
    }
    void DFS(int v)
    {
        visited[v] = true;
        cout << v << " ";
        list<int>::iterator i;
        for (i = adj[v].begin(); i != adj[v].end(); ++i)
        {
            if (!visited[*i])
            {
                DFS(*i);
            }
        }
```



```cpp
    }

    void addEdge(int v, int w)
    {
        adj[v].push_back(w);
    }
    void DFS(int v)
    {
        visited[v] = true;
        cout << v << " ";
        list<int>::iterator i;
        for (i = adj[v].begin(); i != adj[v].end(); ++i)
        {
            if (!visited[*i])
            {
                DFS(*i);
            }
        }
    }
};

int main()
{
    Graph g(4);
    g.addEdge(0, 1);
    g.addEdge(0, 2);
    g.addEdge(1, 2);
    g.addEdge(2, 0);
    g.addEdge(2, 3);
    g.addEdge(3, 3);
    g.DFS(2);
}
```

- **Output**



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#04.cpp -o Task#04 } ; if ($?) { .\Task#
04 }
0 1 2 3
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```