CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Course No: CSE- 244

Course Title:
Algorithm Design & Analysis (Sessional)

Experiment No: 06

Name Of the Experiment: : Graph Traversal.

Identity Details

Name of the Student: Md. Safayat Bin Nasir

Student Id Number: 2004099

Level: 2 Term: 2 Section: B

Group: B1

Date of performance: 05-11-2023

Date of Submission: 19-11-2023

Remarks

Program 1: Given a bi-directional graph, traverse the full graph using DFS. Print the nodes during traversal..

```cpp
#include <bits/stdc++.h>
using namespace std;
bool visited[100000];
long long dist[100000];
vector<long long> graph[1000];
void dfs(long long source)
{
    visited[source] = true;
    for(auto &next:graph[source]){
        if(visited[next]) continue;
        cout << next << " ";
        dfs(next);
    }
}
int main()
{
    long long node, edge;
    cin >> node >> edge;
    for (long long i = 1; i <= edge; i++)
    {
        long long u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }
    cout  << 1 << " "; dfs(1);
}
```

Output:

TC 1 Passed 185ms

Input:

```
6 8
1 2
1 3
2 4
2 5
3 5
4 5
4 6
5 6
```

Expected Output:

```
1 2 4 5 3 6
```

Received Output:

```
1 2 4 5 3 6
```

Program 2: Given a bi-directional graph, traverse the full graph using BFS. Print the nodes during traversal.

```cpp
#include <bits/stdc++.h>
using namespace std;
bool visited[100000];
vector<long long> graph[1000] , dist(100000);
void bfs(long long source)
{
    queue<long long> q;
    visited[source] = 1;
    dist[source] = 0;
    q.push(source);
    cout << source << " " ;
    while (!q.empty())
    {
        int node = q.front();
        q.pop();
        for (long long i = 0; i < graph[node].size(); i++)
        {
            long long next = graph[node][i];
            if (visited[next] == 0)
            {
                cout << next << " ";
                visited[next] = 1;
                q.push(next);
            }
        }
    }
}
int main()
{
    long long node, edge;
    cin >> node >> edge;
    for (long long i = 1; i <= edge; i++)
    {
        long long u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }
    bfs(1);
}
```

Output:

TC 1 Passed 24ms

Input:

```
6 8
1 2
1 3
2 4
2 5
3 5
4 5
4 6
5 6
```

Expected Output:

```
1 2 3 4 5 6
```

Received Output:

```
1 2 3 4 5 6
```