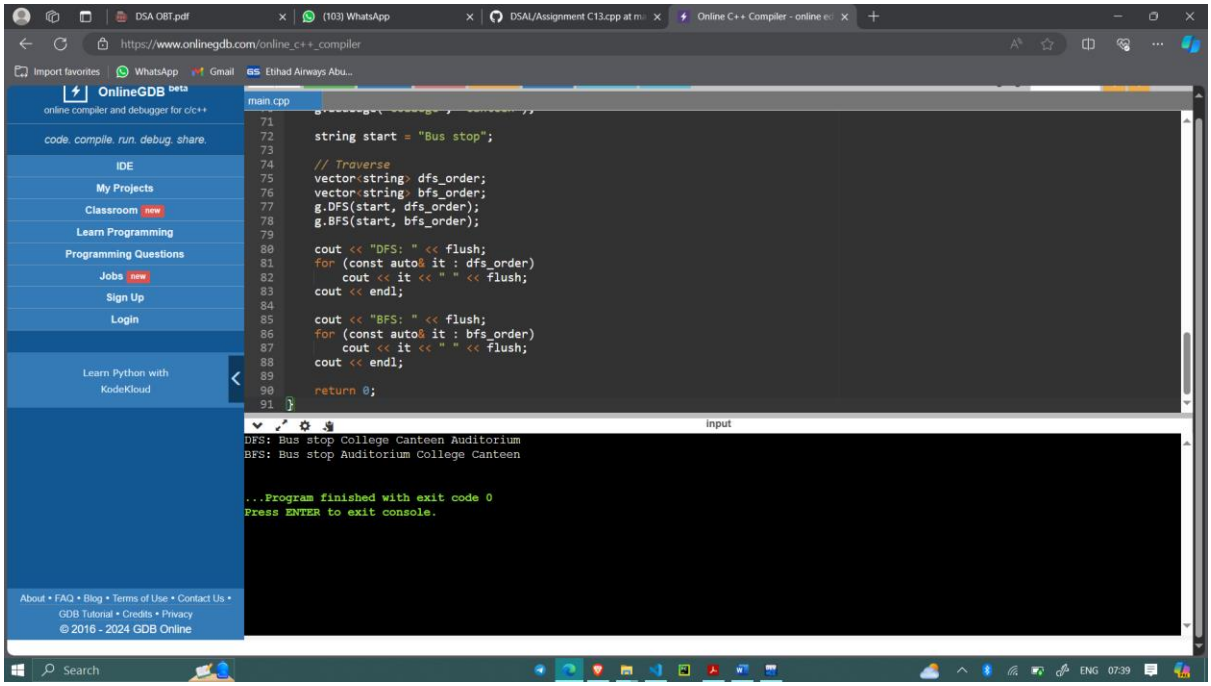
ASSIGNMENT 13

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

class Graph {
private:
    unordered_map<string, vector<string>> adjList;

public:
    Graph() {};

    void DFS(const string& start, vector<string>& order) {
        stack<string> s;
        unordered_map<string, bool> visited = {};

        s.push(start);

        while (!s.empty()) {
            auto curr = s.top();
            s.pop();

            if (!visited[curr]) {
                visited[curr] = true;
                order.push_back(curr);
            }

            for (const auto& neighbour : adjList[curr]) {
                if (!visited[neighbour]) {
                    s.push(neighbour);
                }
```

```
        }
    }
}

void BFS(const string& start, vector<string>& order) {
    queue<string> q;
    unordered_map<string, bool> visited = {};
    string curr;

    q.push(start);

    while(!q.empty()) {
        curr = q.front();
        q.pop();

        if (!visited[curr]) {
            visited[curr] = true;
            order.push_back(curr);
        }

        for (const auto& neighbour : adjList[curr]) {
            if (!visited[neighbour])
                q.push(neighbour);
        }
    }
}

void addEdge(const string& parent, const string& child) {
    adjList[parent].push_back(child);
    adjList[child].push_back(parent);
}
```

```
};

int main() {

    // Create Graph
    Graph g;
    g.addEdge("Bus stop", "Auditorium");
    g.addEdge("Bus stop", "College");
    g.addEdge("Auditorium", "College");
    g.addEdge("College", "Canteen");

    string start = "Bus stop";

    // Traverse
    vector<string> dfs_order;
    vector<string> bfs_order;
    g.DFS(start, dfs_order);
    g.BFS(start, bfs_order);

    cout << "DFS: " << flush;
    for (const auto& it : dfs_order)
        cout << it << " " << flush;
    cout << endl;

    cout << "BFS: " << flush;
    for (const auto& it : bfs_order)
        cout << it << " " << flush;
    cout << endl;

    return 0;
}
```

OUTPUT :


DSA OBT.pdf
(103) WhatsApp
DSAL/Assignment C13.cpp at ma
Online C++ Compiler - online ed
https://www.onlinegdb.com/online_c++_compiler
Import favorites
WhatsApp
Gmail
Etihad Airways Abu...
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
Learn Python with KodeKloud
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2024 GDB Online
main.cpp
71
72 string start = "Bus stop";
73
74 // Traverse
75 vector<string> dfs_order;
76 vector<string> bfs_order;
77 g.DFS(start, dfs_order);
78 g.BFS(start, bfs_order);
79
80 cout << "DFS: " << flush;
81 for (const auto& it : dfs_order)
82 cout << it << " " << flush;
83 cout << endl;
84
85 cout << "BFS: " << flush;
86 for (const auto& it : bfs_order)
87 cout << it << " " << flush;
88 cout << endl;
89
90 return 0;
91 }
input
DFS: Bus stop College Canteen Auditorium
BFS: Bus stop Auditorium College Canteen
...Program finished with exit code 0
Press ENTER to exit console.
Search
ENG 07:39