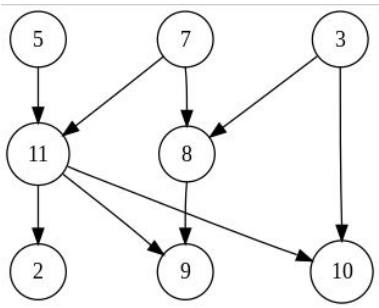
Assignment 13

Perumalla Dharan
AP21110010201

1. **Implement topological sorting in the following graph.**




```
#include <iostream>
#include <vector>
#include <unordered_map>
#include <stack>

using namespace std;

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

void topologicalSortUtil(int v, unordered_map<int,
vector<int>>& adjList, unordered_map<int, bool>& visited,
stack<int>& Stack) {
    visited[v] = true;

    for (int i : adjList[v]) {
        if (!visited[i]) {
            topologicalSortUtil(i, adjList, visited,
Stack);
        }
    }
```

```
    Stack.push(v);
}

vector<int> topologicalSort(unordered_map<int,
vector<int>>& adjList) {
    unordered_map<int, bool> visited;
    stack<int> Stack;
    vector<int> result;

    for (auto& iter : adjList) {
        visited[iter.first] = false;
    }

    for (auto& iter : adjList) {
        if (!visited[iter.first]) {
            topologicalSortUtil(iter.first, adjList,
visited, Stack);
        }
    }

    while (!Stack.empty()) {
        result.push_back(Stack.top());
        Stack.pop();
    }

    return result;
}

int main() {
    unordered_map<int, vector<int>> adjList;

    int numEdges;
    cout << "Enter the number of edges: ";
```

```
    cin >> numEdges;

    cout << "Enter the edges (format: source destination):"
<< endl;
    for (int i = 0; i < numEdges; ++i) {
        int source, destination;
        cin >> source >> destination;
        addEdge(adjList, source, destination);
    }

    vector<int> sortedNodes = topologicalSort(adjList);

    cout << "Topological Sort: ";
    for (int node : sortedNodes) {
        cout << node << " ";
    }
    cout << endl;

    return 0;
}
```

Output

```
Enter the number of edges: 9
Enter the edges (source destination):
5 11
11 2
11 9
11 10
7 11
7 8
8 9
3 8
3 10
Topological Sort: 7 5 11 2 3 10 8 9
```