```cpp
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
bool canFinish(int numCourses, vector<vector<int>>& prerequisites) {
    vector<int> inDegree(numCourses, 0);
    vector<vector<int>> adjList(numCourses);

    for (const auto& edge : prerequisites) {
        int from = edge[1];
        int to = edge[0];
        inDegree[to]++;
        adjList[from].push_back(to);
    }

    queue<int> zeroInDegreeQueue;
    for (int i = 0; i < numCourses; ++i) {
        if (inDegree[i] == 0) {
            zeroInDegreeQueue.push(i);
        }
    }

    while (!zeroInDegreeQueue.empty()) {
        int course = zeroInDegreeQueue.front();
        zeroInDegreeQueue.pop();

        for (int neighbor : adjList[course]) {
            inDegree[neighbor]--;
            if (inDegree[neighbor] == 0) {
                zeroInDegreeQueue.push(neighbor);
            }
        }
    }

    for (int degree : inDegree) {
        if (degree > 0) {
            return false;
        }
    }
    return true;
}

int main() {
    int numCourses = 3;
    vector<vector<int>> prerequisites = {{1, 2}, {2, 3}};

    if (canFinish(numCourses, prerequisites)) {
        cout << "It is possible to finish all courses." << endl;
    } else {
        cout << "It is impossible to finish all courses." << endl;
    }
    return 0;
}
```