```cpp
// Raul Martinez
// raul.martinez01@utrgv.edu

#include <iostream>
#include <fstream>
#include <ctime>
#include "graph.h"

using namespace std;

int main()
{
    directedGraph graph;
    int vs , v, adj, edg;
    double w;
    fstream inFile;

    // Graph builder
    cout << "Building graph..." << endl;
    clock_t startTime = clock();
    inFile.open("NYC.txt");
    inFile >> vs;
    inFile >> edg;
    for (int i = 0; i < vs; i++)
        graph.addVertex(i);
    while (!inFile.eof())
    {
        inFile >> v >> adj >> w;
        graph.addDirectedEdge(v, adj, w);
    }
    inFile.close();
    clock_t endTime = clock();
    cout << "Elapsed time:" << (endTime - startTime)
    / (double) CLOCKS_PER_SEC << " sec." << endl;

    cout << endl;

    // First test
    cout << "Shortest path from 42155 to 234370: ";
    startTime = clock();
    graph.shortestPath(42155, 234370);
    endTime = clock();
    cout << "Elapsed time:" << (endTime - startTime)
    / (double) CLOCKS_PER_SEC << " sec." << endl;

    cout << endl;

    // Second test
    cout << "Shortest path from 34967 to 983: ";
    startTime = clock();
    graph.shortestPath(34967, 983);
    endTime = clock();
    cout << "Elapsed time:" << (endTime - startTime)
    / (double) CLOCKS_PER_SEC << " sec." << endl;

    cout << endl;
```

```
        // Third test
        cout << "Shortest path from 90210 to 211111: ";
        startTime = clock();
        graph.shortestPath(90210, 211111);
        endTime = clock();
        cout << "Elapsed time:" << (endTime - startTime)
        / (double) CLOCKS_PER_SEC << " sec." << endl;

        cout << endl;

        return 0;
    }
```