CSCI 2170 Summer 2016 OLA 7 Part 2 (due midnight, Thursday Aug 4th. No late program is accepted)

The FlyWithUS airline company wants you to help them develop a program that generates flight itinerary for customer requests to fly from an origin city to a destination city. For each customer request, indicate

- Whether a sequence of FlyWithUs flights from the origin city to the destination city exists. Give appropriate message if it does not exist.
- If an itinerary exists, the actual flight itinerary, and the price of the entire flight itinerary is printed in well formatted table format (as shown below).

This program will build on the code written for OLA 7 Part 1. The three data files used in the program are:

(1) **cities.dat** : the names of cities that airline serves, one name per line, for example:

30
Albuquerque
Chicago
San-Diego
.

(2) **flights.dat**: pairs of city names (each pair represents the origin and destination of one of the flights) plus a price indicating the airfare between these two cities, for example:

178 *Albuquerque* *Chicago* *250*
703 *Chicago* *San-Diego* *325*
550 *Nashville* *San-Diego* *180*
...

(3) **requests.dat** : pairs of city names, each pair represents a request to fly from some origin to some destination, for example:

Albuquerque *Chicago*
Chicago *San-Diego*
Nashville *Seattle*
San-Diego *New-York-City*
...

You have already copied the first two data files in OLA7 part 1. In this OLA, copy the flight request file with this command:

cp ~cen/data/requests.dat **codelite-directory/project-directory/.**

The program should produce output of the following form:

Request is to fly from Albuquerque to San-Diego.
FlyWithUS *airline serves between these two cities.*
The flight itinerary is:

Flight #	*From*	*To*	*Cost*
178	*Albuquerque*	*Chicago*	*$250*
703	*Chicago*	*San-Diego*	*$325*
		Total:	*$575*

Request is to fly from Albuquerque to Paris.
Sorry, FlyWithUS *airline does not serve Paris.*

Request is to fly from San-Diego to Chicago
Sorry, FlyWithUS *airline does not fly from San-Diego to Chicago.*

You are required to:

- Use the C++ STL container "stack"
- Add the following data and methods to the FlightMap class:
- For FlightMap class,
  - The **IsPath** algorithm discussed in class used to find the itinerary between two cities. ***Modify the code to display the full itinerary if one is found***.
    - ***Data***
      - ***visited*** array to record whether a city has been visited during the itinerary planning process. This array should also be allocated dynamically. (Hint: do this in ReadCities method)

- ***Methods:***
  - **MarkVisited**: record that a city has been visited; void MarkVisited(string city)
  - **IsVisited**: returns true if a city has been visited; bool IsVisited(string city)
  - **UnvisitAll**: mark all the cities as not visited; void UnvisitAll()
  - **GetNextCity**: pass back the first adjacent city that has not been visited. If such a city is found, return true, otherwise, return false; bool GetNextCity(string city, string &nextCity)

Electronic submission for OLA 7 Part two:
Submit the following files:

- type.h and type.cpp
- slist.h and slist.cpp
- flightmap.h and flightmap.cpp
- main.cpp