CSCI 2170 Spring 2006
OLA 6 (Due Tuesday, April 11th)

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

- whether a sequence of EastWest flights from the origin city to the destination city exists,
- if an itinerary exists, the actual flight itinerary, and the price of the entire flight itinerary.

This program will build on the code written for OLA5.
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:

19
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*

…

Copy the files into your own account by : frank% cp ~cli/data/filename filename

The program should produce output of the following form:

Request is to fly from Albuquerque to San-Diego.
EastWest 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, EastWest airline does not serve Paris.

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

You are required to:

- Implement the pointer based StackClass
- In the flight map ADT (FlightMapClass) created in OLA 5, add the following functions and data:
  - the non-recursive IsPath algorithm discussed in class to find the itinerary between two cities. Modify the code to display the full itinerary if one is found.
  - Additional functions used by the IsPath algorithm need to be added to the FlightMapClass (e.g., MarkVisited, IsVisited, UnvisitAll, GetNextCity…)
  - Additional data for the class: visited array to record whether a city has been visited during the itinerary planning process.

To turn in the program, follow these steps:

```
frank% script log6
frank% pr –n –e4 listClass.h
frank%pr –n –e4 listClass.cpp
frank%pr –n –e4 stackClass.h
frank%pr –n –e4 stackClass.cpp
frank%pr –n –e4 FlightMapClass.h
frank%pr –n –e4 FlightMapClass.cpp
frank%pr –n –e4 ola6.cpp
frank%aCC listClass.cpp stackClass.cpp FlightMap.cpp ola6.cpp –o run
frank% run
frank%exit
frank% lph log6
```