

## CSCI 3110 Project 4

The BlueSky 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 BlueSky 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 based on the code written for Project 3.

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
...
```

The program should produce output of the following form if an itinerary exists:

```
Request is to fly from Albuquerque to San-Diego.
BlueSky 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
```

The program should produce output of the following if

- (1) the origin or the destination city is not served by the company
- (2) there is no flight itinerary from the origin to the destination city:

```
Request is to fly from Albuquerque to Paris.
Sorry, BlueSky airline does not serve Paris.
```

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

### You are required to:

- Use STL stack class in the development of the algorithm for deriving the itinerary
- In the flight map ADT (FlightMapClass) created in Project 3. 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...) see the updated FlightMapClass.h.
  - Additional data for the class: visited array to record whether a city has been visited during the itinerary planning process.