CSCI 2170 OLA 7 Part A Due midnight, Friday, July 29th

The FlyWithUs airline company would like for you to help them develop a program that generates flight itinerary for customer requests to fly from one city to another city. Your are given these two data files containing the list of cities and all the flights served by the company:

cities.dat : First line of the data file gives the total number of cities served by the company. Next, the names of cities the airline serves, one name per line, for example:

31 ← total number of cities served by the company
Albuquerque
Chicago
San-Diego
…

flights.dat : each flight record contains the flight number, a pair of city names (each pair represents the origin and destination city of the flight) plus a price indicating the airfare between these two cities, for example:

178	*Albuquerque*	*Chicago*	*450*
703	*Chicago*	*Atlanta*	*120*
550	*Nashville*	*San-Diego*	*580*
833	*Chicago*	*Washington-DC*	*500*
1180	*Atlanta*	*Chicago*	*120*

…

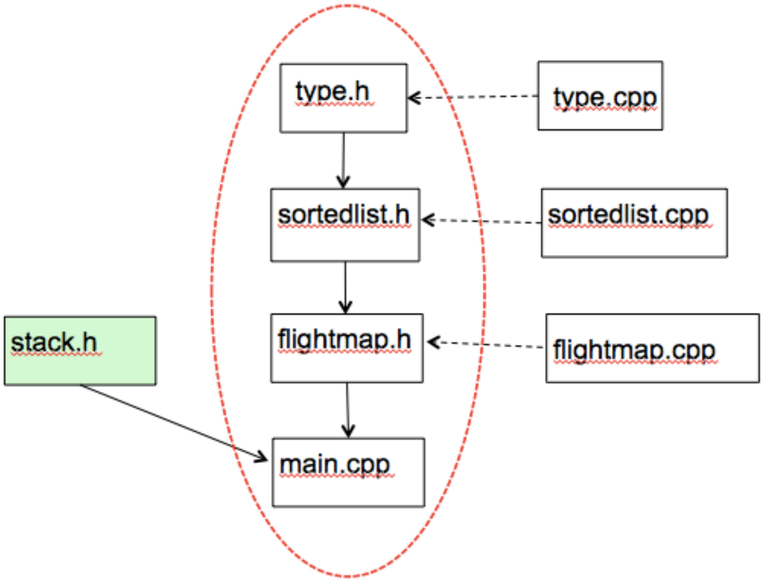
To support flight itinerary generation, it is necessary to build a database of all the available flights. For this assignment, you are to write a program that builds an adjacency list structure to store flight information. An adjacency list is an array of linked lists. Your program needs to have two classes, a sorted-list class (discussed in class) and a flightMap class. The flightMap class implements the adjacency list structure.

Your client program reads in a list of city names for which the company currently serves. The city names are in the data file named "cities.dat". Then, it reads in a list of flights currently served by the company. The flight information is in the data file "flights.dat". Copy the data files into your own account by:

```
ranger$ cp ~cen/data/cities.dat  cities.dat
ranger$ cp ~cen/data/flights.dat  flights.dat
```

Since we are to store flight records as data in the linked list, the listItemType for the linked list with be a flight record structure. Define the struct in a type.h and a type.cpp file as following:

- Create a type.h and type.cpp file to define the list item type:
  - type.h defines the FlightRec structure and the overloaded operators (==, >, and <, <<) for this structure
  - type.cpp implements the overloaded operators



The adjacency list will be implemented in a FlightMap class. Implement the **Flight Map** ADT(flightMap.h and flightMap.cpp) which has the following data and at least the following methods:

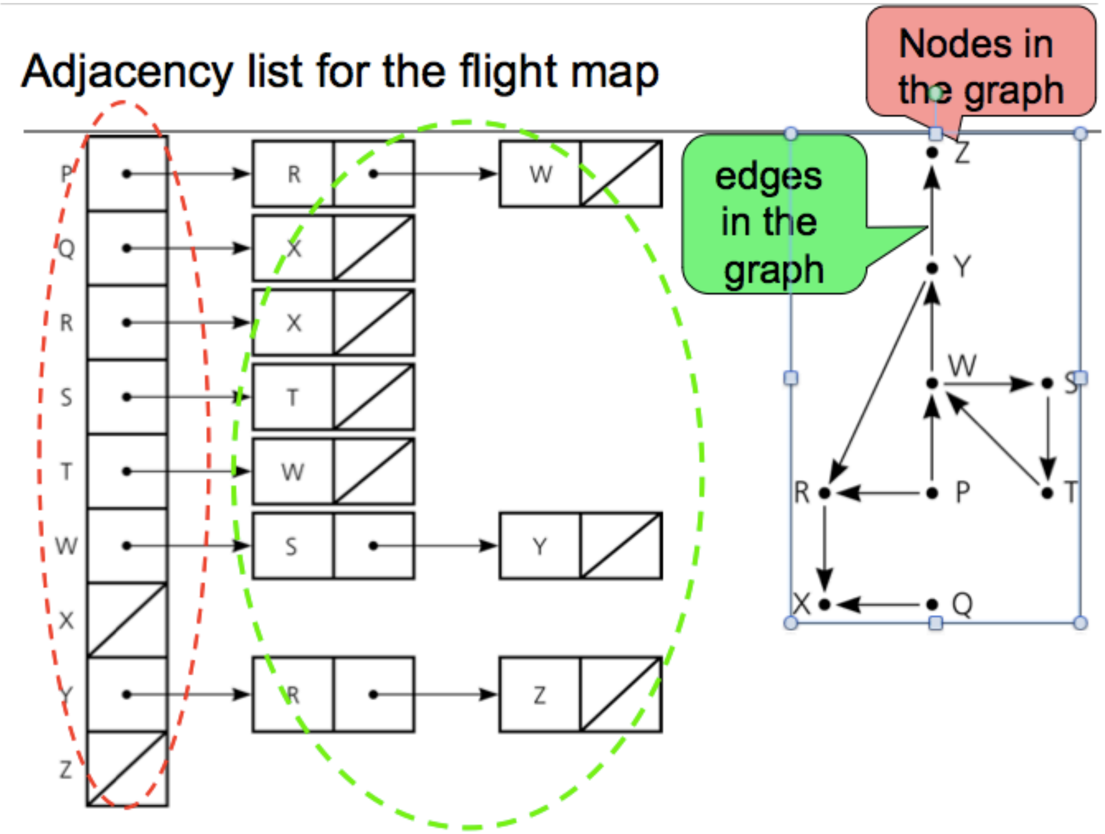
- **Data:**
  - number of cities served by the company
  - list of cities served by the company – (use a 1D array for this. you should create/allocate memory for this array dynamically)
  - flight map of the company stored in the form of an adjacency list, e.g., array of sortedListClass objects. (The array needs to be created dynamically)
- constructor(s) and destructor

- **Methods:**
  - reads cities (cities.dat)
  - reads flight information and build the adjacency list (flights.dat)
  - displays the flight information as shown above.

- Implement the client program that:
  - Creates a flight map object
  - Reads the list of cities
  - Reads flight info and builds the flight map, i.e., the adjacency list

Print the flight map in a formatted table as shown You program should print out the flight map/information in a well-formatted table as shown below:

```
        Origin             Destination        Flight    Price
==========================================
  From Atlanta to:      Chicago            1180      $120
                        New-York           320       $180
                        Seattle            1200      $210
  From Chicago to:      Atlanta            703       $120
                        Washington-DC      833       $500
  ……
```


Adjacency list for the flight map
Nodes in the graph
edges in the graph
P
Q
R
S
T
W
X
Y
Z
R
W
X
X
T
W
S
Y
R
Z
Z
Y
W
S
R
P
T
X
Q