Overview

Students will write the data structure to keep track of an airline’s route map (for simplicity we will only look at toy examples.) The map is stored as an array of hub cities, each hub city has a name and a list of connecting cities.

Details

Students will have an array of structs of type Hub. For simplicity the array of hubs will not be dynamic, it will be limited to 25 hubs. Each individual hub will contain a string, a dynamic array of City structs, and the size of the list of cities. Specifically,

```c
struct Hub {
    string name;
    City *cityList;
    int cityCount;
};
```

Each city will consists of a city name (a string) and the distance to the hub (a double).

```c
struct City {
    string name;
    double distance;
};
```

These structures are part of the class AirlineMap whose main data member is the array of Hubs. (It is a private data member.) There is a private data member freeLocation which keeps track of the next available slot in the array. There are two public methods to implement, inputHubFromUser, remove hub.
Hub AirlineMap::inputHubFromUser()

The input method will create a new hub. The user will input the hub name, the number of cities in the city list and the city list itself. The city list is a dynamic array, students will create a dynamic array using new, corresponding to the size of the city list input form the user.

void AirlineMap::removeHub(string name)

The remove hub will remove the hub from the list and delete its city list. This should be done by deleting the dynamic array, setting the city count to zero, and the hub name to the empty string. The array should not have holes in it, students will have to ensure that the array is contiguous. Think about this before coding!

In addition to insert and remove, we will have a search function, which takes a origin and a destination and checks if there is a way to travel between the two with at most one connection. Similarly there is a related distance function. These methods will be implemented at a later date.

We have provided the following methods for completeness. Students may use any of these methods.

//constructor
AirlineMap();

//insert method (user input)
void insertHub();

//internal insert method
void insertHub(Hub hubToAdd);

//returns index in array of hub named, name
int indexOf(string name);

//functions to think about
bool isConnected(string origin, string destination);
double distance(string origin, string destination);

The majority of the implementation is provided, in addition we will provide a short main function which students should expand for proper testing. This code is located on the course website.

Submissions should be done using provide.

provide comp15 hw1 AirlineMap.cpp