CISC 3130 Ari Mermelstein

Homework #2: Due February 22

In this homework assignment, you will implement a Contact List class. This assignment will use the Contact class from homework 1.

Your ContactList should use a vector (from the STL) as the underlying structure, and should have the following functionality:

Constructors:

- A constructor that creates an empty ContactList.
- A constructor that uses an array of Contacts and the size of that array in order to initialize the ContactList.

  (i.e. prototype ContactList(Contact* contacts, **int** numContacts);)

Operations:

- A function that searches for a particular contact by last name, and returns a reference to the Contact. If there is no such Contact, a sentinel value should be returned. (like the default Contact.) (You will return a reference to the Contact, so that the user may call update() in order to change some information.)
- A function that searches for a Contact by phone number, and returns a reference to the Contact. If there is no such Contact, a sentinel value should be returned. (like the default Contact.) (You will return a reference to the Contact, so that the user may call update() in order to change some information.)
- A function that searches for and returns a vector containing all Contacts with a last name starting with a particular letter. If there are no such Contacts, you should return the empty vector.

- A function that searches for and returns a vector containing all Contacts that live in a particular city. If there are no such Contacts, you should return the empty vector.
- A function that allows you to add a Contact to the ContactList. You should only add a Contact to the list if it is not there already.
- A function that returns the size of the ContactList.
- A function that erases a contact from the contact list with a particular last name.

Operators:

- An overloaded [ ] operator that returns a Contact by reference at a certain index in the ContactList. You should throw an exception if the index is out of bounds.
- An overloaded << operator that prints out a ContactList. You should print out each Contact using the << operator that you already defined on your Contact class. Please do not call all of the Contact class's accessor functions when you write this operator.

Your private data should be a vector of Contacts. As usual, you may write as many private functions as you'd like as helper functions to implement your class.