CECS 275 – Project 3 – Rolodex

Create a class that represents a Contact (.h and .cpp). A Contact has a first name, last name, phone number, address, city, zip, and state, all stored as private strings. Make a constructor and get and set methods to change the Contact's information. Overload the < (compare by last name, then first name), == (first and last name), <<, and >> operators.

Use the Linked List class (.h and .cpp) given in lecture (normal or recursive). Modify the Node to store a Contact. Methods of the Linked List class should include: get, set, size, isEmpty, add (to end), remove (by last name, first name), remove (by index), sort, search (by last name, returns list of matches), search (by zip code, returns list of matches), search (by first and last name, returns a Contact). and an overloaded << operator to display an enumerated list of Contacts in sorted order (by last name then first name).

In your main, create a Linked List and initially populate it with Contacts read in from the file 'addresses.txt'. Allow the user to add, remove, search, modify, and display the list. When the user quits, write the updated list back to the file in the same format.

Create functions and menu options to:

1. Add
   a. add a Contact to the end of the list. Use the overloaded >> to input.
   b. ask the user enter the first and last name, then allow the user to input any other information they have. If they do not enter a value for a particular field, store an empty string (check that a first and last name was entered).
2. Remove
   a. by first and last name – if found, then remove the Contact from the list.
   b. by index in the list – display the enumerated list (1 based), get the user's choice, then remove the Contact at that location.
3. Search
   a. last name – display all information for each Contact with that last name.
   b. first and last name – display all of information for the matching Contact.
   c. zip-code – display all information for each Contact's with that zip code.
4. Update
   a. prompt the user to enter the Contact's full name. Then prompt the user to enter which part of the Contact they'd like up update. Take in the updated information, and update the Contact's data.
5. Display
   a. display an enumerated list of all contacts alphabetically by last name, and then first name.
6. Quit – write back to the file when the user is finished.

Notes:

1. Document all classes, methods, and functions (@params and @returns).
2. You should have at least 5 files (.h's and .cpp's)
3. Check all user input.
4. Read and write to the same file. Write back in the same format.

Example Output:

```
Rolodex Menu:
1. Add Contact
2. Remove Contact
3. Search for Contact
4. Modify Contact
5. Display Contacts
6. Quit
1
Add Menu:
Enter First Name: Bart
Enter Last Name: Simpson
Enter Phone #: 555-1212
Enter Address:
742 Evergreen Terrace
Enter City: Springfield
Enter Zip-Code: 12345
Enter State: IL
Rolodex Menu:
1. Add Contact
2. Remove Contact
3. Search for Contact
4. Modify Contact
5. Display Contacts
6. Quit
3
Search Menu:
A. Search by Full Name
B. Search by Last Name
C. Search by Zip-Code
B
Enter Last Name: Simpson
1. Bart Simpson
   555-1212
   742 Evergreen Terrace
   Springfield 12345 IL
2. Homer Simpson
   555-1212
Rolodex Menu:
1. Add Contact
2. Remove Contact
3. Search for Contact
4. Modify Contact
5. Display Contacts
6. Quit
4
Enter First Name: Herman
Enter Last Name: Munster
Herman Munster
131-1313
Update Menu:
A. Enter First Name
B. Enter Last Name
C. Enter Phone Number
D. Enter Address
E. Enter City
F. Enter Zip-Code
G. Enter State
Q. Save and Quit
D
Enter Address:
1313 Mockingbird Lane
Modify Menu:
A. Enter First Name
B. Enter Last Name
C. Enter Phone Number
D. Enter Address
E. Enter City
F. Enter Zip-Code
G. Enter State
Q. Save and Quit
Q
Saving...
Rolodex Menu:
1. Add Contact
2. Remove Contact
3. Search for Contact
4. Modify Contact
5. Display Contacts
6. Quit
6
Quitting...
```