COMP-211 Data Structures

Assignment 2

Aim

To familiarize students with the use of linked lists.

Submission

Late submissions are not accepted.

Attention: Exhaustive checks will be made for copies (special software will be used). Copy-cases will immediately receive a grade **zero**.

Grading

A program to get full marks its necessary to:

1. Have your personal details, your program details (e.g., name, surname, program title, date, program goal, etc.) are included at the beginning of the code.
2. Use appropriate comments.
3. Have well indentation code.
4. Have meaningful identifiers.
5. Solve the problem.

Programs that do not compile or run (crash) automatically get grade **zero**.

Program Description – Telephone directory

The purpose of this project is to implement a C++ program to maintain and manage a telephone directory. The telephone directory will be used to store subscriber details (surname, name and telephone(s)). The implementation should be done using SORTED linked lists. To manage the directory your program should offer the following methods through a menu:

1. **Add a new subscriber**: Ask the user for the new subscriber's name, surname and telephone number(s), organize them appropriately in a structure (class or struct) called ***User***, and store this structure in a linked list (sorted with surname+name) using dynamic memories.
2. **Removing a phone number from a subscriber**: Ask for the surname and name of the subscriber, and the phone number to be removed and perform the removal.
3. **Add a phone number to a subscriber**: Ask for the subscriber's surname and name, the phone number to be added, and add it.
4. **Display a subscriber's phone numbers**: Ask for the subscriber's surname and name, and display all the subscriber's phones numbers.
5. **Delete a subscriber**: Ask for the subscriber's surname and name and delete the record associated with that name.
6. **Sorted display of directory items by Surname-Name**: Print all directory items sorted in alphabetical order by subscriber **surname** and then **name**.
7. **Exit**: Your program should save all the entries in the table to the file specified at the start of the program (as described in the next paragraph) and terminate it.

Your program should read from the file "**teldir.txt**", the subscriber data given in previous executions of the program (during the first execution of the program the file should be empty, or if it does not exist to be created) and at termination it should save in this file the data of the subscribers as they have been configured during the execution.

Therefore, to process the telephone-directory information, you should use a sorted linked-list with dynamic memory entry. Therefore, there is no need to set a maximum number of users. Each structure will be stored in a separate file so that it acts as a file-library (header-file) for future use. It should be implemented as much as possible independent of the specific application so that it is useful for future applications that require the use of linked-lists. Specifically, you need to implement the methods to:

- insert an item in a sorted list,
- export an item from a sorted list,
- search an item in a sorted list, and
- print information of a sorted list.

Caution:

- Your implementation should be based on Object-Oriented-Programming (data abstraction – information hiding).
- Do not waste memory unnecessarily.
- Carefully organize your program by dividing the code into different files, each covering a different logical entity, and making appropriate use of header files.
- Use class-templates in your implementation.

You must hand in:

(a) A ZIP-file containing your ***Project folder*** with all files include (+ the *text-file* ***teldir.txt***).
(b) Another ZIP file containing the ***executable file (.exe)*** and the *text-file* ***teldir.txt*** so I can run it. Make sure that your executable file runs on **Windows**.
(c) Presentation of your *algorithm* and *test-data* in a ***PowerPoint*** file.
(d) The "*Declaration of Ownership*".

Good Luck!