

### **IMPORTANT REMINDERS**

1. It is not allowed to use USB sticks during the lab sessions.
2. You should unplug your ethernet cables during the lab sessions.
3. Any reference book or help material (C++) is allowed.

In this experiment, you are required to write a phonebook application by yourself again. However, this time you are supposed to use linked lists to store the records in the phone book.

- Write the necessary structures for this aim.
- Variables for names and surnames of records be dynamic and their sizes should be decided according to the length of user's input during runtime.
- You should implement the following methods for your structure. `add()`, `remove()`, `update()`, `search()` and `list()`.
- `add()` should add records alphabetically and you don't need to implement any file operations.
- `list()` method should print records to the screen.
- You should deallocate memory before termination of your program.
- The update function as is (in the course presentation of the 4th week) could actually ruin the order by name in the list. If the user updates a record that has a name Can with the name Mehmet, the order is destroyed. Write your update function that, in such a situation, takes the node out of its current position and places it in the appropriate position in the list.
- Since the records are added to the phonebook in alphabetical order, the search function is currently performing unnecessary iteration. Example: if we are searching for records starting with "ah", after the records meeting this criteria have ended, we should stop. Write the search function so that it produces output in a shorter time.
- We don't want the "search" function to make user interaction (input - output operations) anymore. The search function will not print the records it finds to the screen, instead, it will create a new list and return a pointer that points to the head of the new list. This list will be a pointer list that holds the references of the records found. The list will be empty if there is not any related record. The operation of printing the list to the screen will be performed by the calling function. After the list usage is over, it has to be returned to the system.