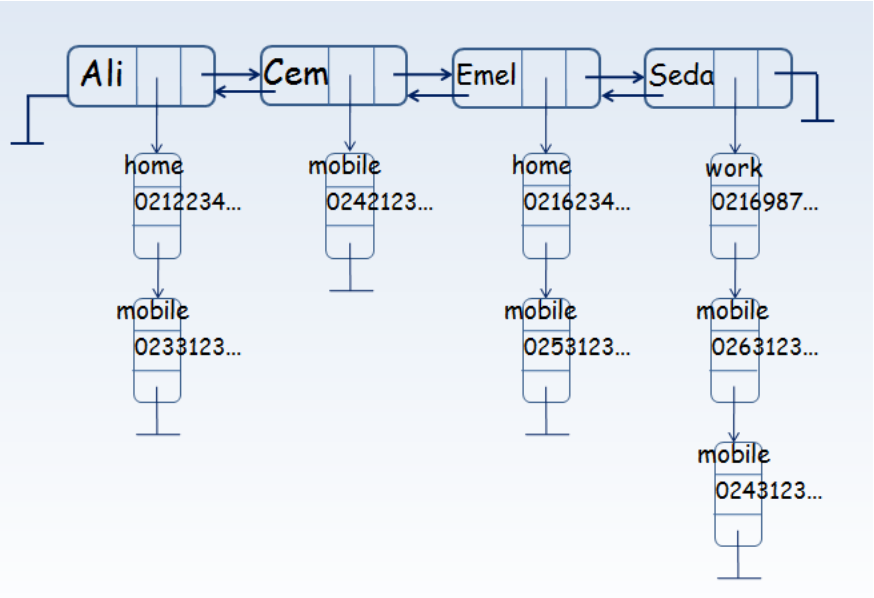
BLG 233E DATA STRUCTURES AND LABORATORY

EXPERIMENT 4 – LINKED LISTS & MULTI LINKED LISTS

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 using multi list structure. Your multi linked list should also be a double linked list. You can see a figure of the mentioned structure below.



- Each person can have more than one phone number on the list. So each person actually have a pointer to another linked list which holds the numbers of this person.
- A number may be a type of home, mobile or work.
- Write the necessary structures for this aim.
- You should implement the following methods for your structure. addPerson(), addNumber(), removePerson(), removeNumber(), updatePerson(), updateNumber(), search() and list().
- addPerson() should add records alphabetically and you don't need to implement any file operations. The list of numbers for the related contact should be an empty list.

- addNumber() should take the name of the person, number and type of the number. If the person does not exist on the list, then an error message should be given. Otherwise, number of the related person should be added to the end of the number list.
- removePerson() takes the name of the contact and deletes the related contact and its numbers from the memory.
- removeNumber() takes the name of the contact and lists the numbers of this contact. User selects a number to be deleted and this number is deleted from the memory.
- updatePerson() takes the name of the contact. After the related contact is found on the list, user enters the updated name (You should not destroy the alphabetic order too by updating a contact's name).
- updateNumber() takes the name of the contact and lists the numbers of this contact. User selects a number and this number is updated with the re-entered number.
- search() method should search a given person on the list and prints the numbers of the related contact to the screen. But since the records are added to the phonebook in alphabetical order, the search function should not perform unnecessary iterations. 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.
- list() method should print all records to the screen.
- You should deallocate memory before termination of your program.