

# COMP 303 Operating System

## Homework 1

Getting familiar with C  
Phone book with linked list

Note:

Install **64bit Ubuntu 18.04.5 LTS (Bionic Beaver)** on **VirtualBox**. Use **Eclipse** as your IDE. Codes written in Windows might not work on Ubuntu (different system calls).

HW1 **version 1** (Check if this version number is changed. Then you would know if I had modified the HW. Version changes will look like this. If you are sure that something is wrong, canvas me to correct this document.)

Last edited: 10 Oct 2020, 20:20

### Question

Write a phonebook software in C and on Ubuntu with the following requirements and rules.

1. Each phonebook entry must at least have
  - a. a name (max 50 characters),
  - b. A phone number,
  - c. Notes section (The note could be in **any length**)
2. When you run the program, it should ask user to pick one of the following options:
  - a. Create a new entry
  - b. List all
  - c. Edit (User should be able to edit name, phone number and the note.)
  - d. Delete (user should be able delete any entry)
  - e. Case insensitive search (find the entries whose names or notes having the search string. 'KAS' and 'kas' should give the same result.)
  - f. Save and exit (save it into a file in human readable text format. You are free to choose your format, e.g, XML, your own format. Ask filename to the user.)
  - g. Load (Get filename from the user)
3. Number of entries in the phonebook **should not be limited**.
4. **Do not preallocate space**. You need to create space when a new entry arrives (dynamic allocation, malloc).
5. Do not leave any empty space. When an entry is deleted, you need to remove it from your list.
6. The list should be a **linked list**.
7. The list should be **always sorted** by name.

8. For each functionality, you need to have a **separate function**. That is, your code will call delete function when delete is selected. Pass your list by reference (see call by reference)
9. Most importantly, your program must not have any **memory leak**. Check with **valgrind**. Attach your valgrind output as an image to the HW.
10. **Provide your Makefile as well.** I will use it to compile your code from the console.
11. Submit your HW code, **a sample phone list file** and **a screenshot showing that your code is working**. Submit them to **Canvas**. Put all of your work (answers) in one folder with your full name and compress it. Submit only one compressed file (.zip or .rar)

First three people who submit a fully working code will be **announced** and get **extra 10 points**.

See the course syllabus for more details about grading and consequences of **plagiarism**.

## Sample output:

Welcome to myPhoneBook! Please select an option:

- 1) New entry
- 2) List all entries
- 3) Edit
- 4) Delete
- 5) Search
- 6) Save and exit
- 7) Load

>> 1

Enter a name: Kasim

Enter a phone: 05324444444

Enter a Note: We met in AGU

Entry has been successfully created. Total number of entries: 1.

Please select an option:

- 1) New entry
- 2) List all entries
- 3) Edit
- 4) Delete
- 5) Search
- 6) Save and exit
- 7) Load

>> 4

1- Kasim

05324444444

We met in AGU

Please enter the number of the entry to be deleted:

>> 1

Entry "Kasim" will be deleted. Are you sure? (y/n)

>> y

Entry is successfully deleted. Total number of entries: 0.