Introduction to C Programming II

Summary: In this homework, you will be implement a simple program that manages a course schedule.

1 Background

In this assignment you will write a simple program that manages a set of courses a student is taking. This program will introduce new keywords (struct, enum), and the handling of heap memory allocations. Similar to the last assignment, we ask that you do your program development on Linux.

The program will store a collection of courses, giving the user options to add to the collection, or remove from that collection. The collection of courses will be stored as a linked list (using structs as nodes). The problem must support any number of courses in the collection. Courses will be composed of a subject (enum), number, teacher, and number of credits. When a user adds a course, a new node must be created, and when a course is removed, its corresponding node is removed.

This document is separated into four sections: Background, Requirements, Compiling a C Program from Terminal, and Submission. You have almost finished reading the Background section already. In Requirements, we will discuss what is expected of you in this homework. In Compiling and Running a C file on Linux, we discuss how to compile and run your program on Xubuntu. Lastly, Submission discusses how your source code should be submitted on BlackBoard.

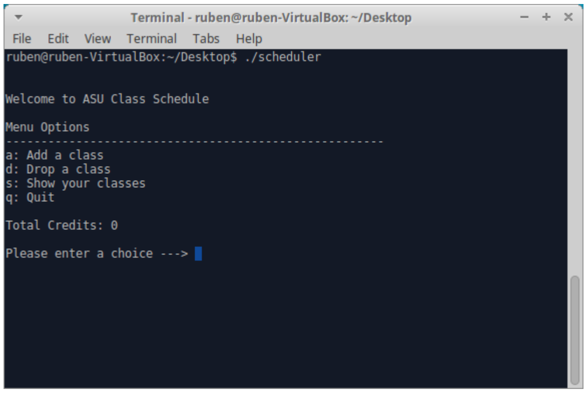
2 Requirements [35 points]

For this assignment you will write a course scheduler program in C. Your program **must**:

- **Compile on GCC 5.3 or greater under Xubuntu (or another variant of Ubuntu) 18.04.**
- Have an enumeration type (called Subject) to represent subjects ("SER=0, EGR=1, CSE=2, EEE=3").
- Have a struct type (called CourseNode) to hold information about a course. Include a subject (see enum type above), a number (int), a teacher (a string of length 1024), and credit hours (int). It should also include the variables necessary to represent a linked list.
- Use a linked list of structs (called course_collection) to store course information. Your program will allow the user to enter any number of courses - this means you must use dynamic memory allocation to store the courses. When a user adds a course, a new node must be created, and when a course is removed, its corresponding node is removed.

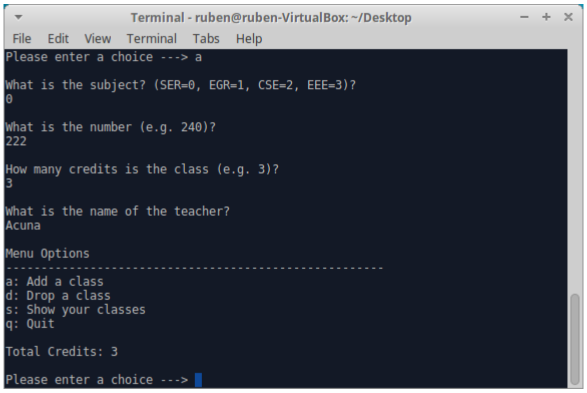
Note: your program may not exactly match the outputs below - screen shots are only provided as samples. Points allocation is:

- Update the main menu to display the total number of credits each time it is shown. [4 points]

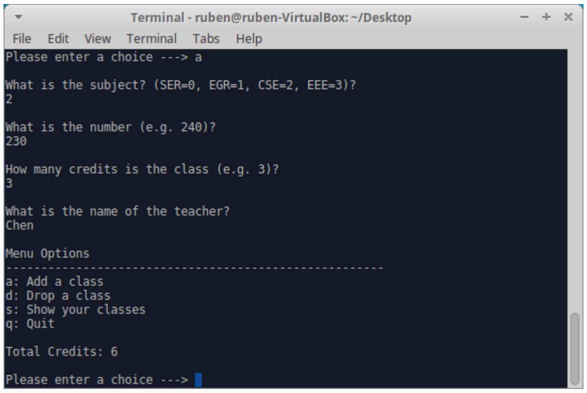

Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
ruben@ruben-VirtualBox:~/Desktop$ ./scheduler


Welcome to ASU Class Schedule

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 0

Please enter a choice --->
```

- The program must not leak memory on exit. [4 points]
- Create a void course_insert() function that adds courses to the collection and keeps it sorted by the course number (e.g., 240). Sorting by subject is not required. Populate the elements of the CourseNode as shown below. Two example inserts are shown. [10 points]

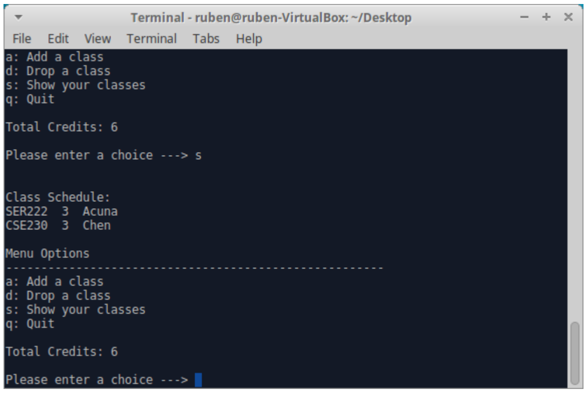

Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
Please enter a choice ---> a

What is the subject? (SER=0, EGR=1, CSE=2, EEE=3)?
0

What is the number (e.g. 240)?
222

How many credits is the class (e.g. 3)?
3

What is the name of the teacher?
Acuna

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 3

Please enter a choice --->
```


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

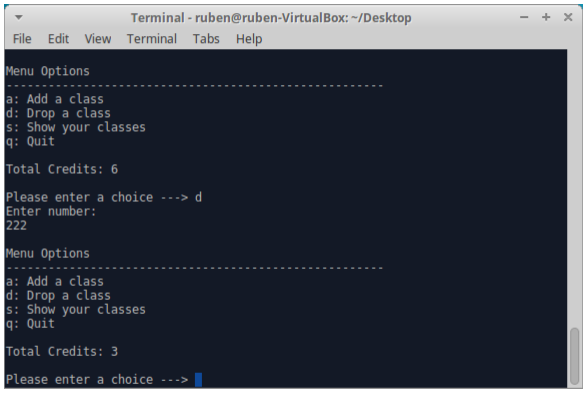
```
Please enter a choice ---> a

What is the subject? (SER=0, EGR=1, CSE=2, EEE=3)?
2

What is the number (e.g. 240)?
230

How many credits is the class (e.g. 3)?
3

What is the name of the teacher?
Chen

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 6

Please enter a choice --->
```

- Create a void schedule_print() function. This function will display the contents of the course_collection list with format Subject Number Credits Teacher as shown. (Hint: Use a switch statement to map from a enumeration to a printf.) [6 points]

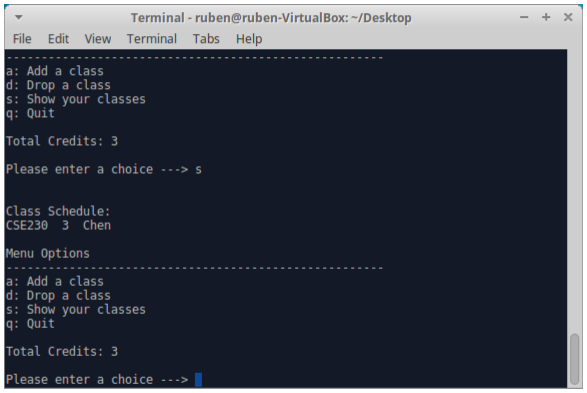

Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 6

Please enter a choice ---> s


Class Schedule:
SER222  3  Acuna
CSE230  3  Chen

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 6

Please enter a choice --->
```

- Create a void course_drop() function. This function will prompt the user for a course by its number. It then searches for the course in course_collection and removes it. If more than one course matches, then drop only the first occurrence. Dropping a course is only allowed if the course exists in the collection, otherwise an error message will be displayed. [8 points]


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 6

Please enter a choice ---> d
Enter number:
222

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 3

Please enter a choice --->
```


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

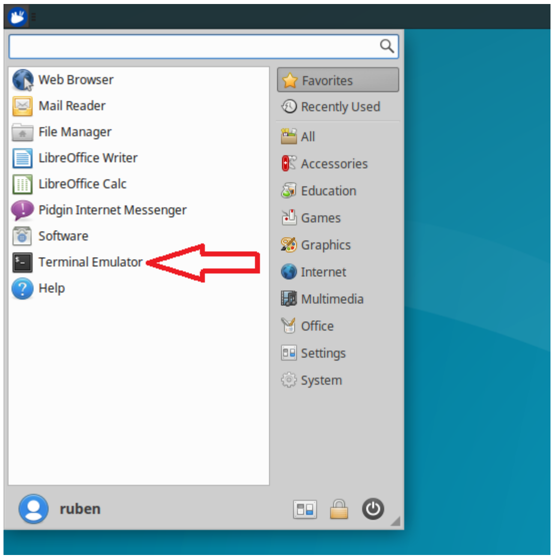
```
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 3

Please enter a choice ---> s


Class Schedule:
CSE230  3  Chen

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 3

Please enter a choice --->
```

- Properly deallocate memory when removing nodes. No memory leaks. [4 points]

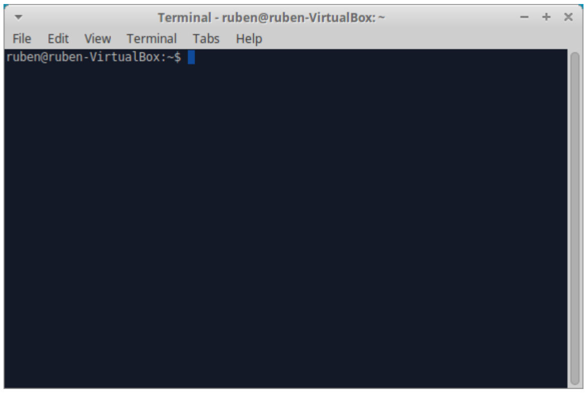
3 Compiling a C file from Terminal

For this assignment, you may use NetBeans to do your development work. However, you can also develop from the command line prompt. The following **optional** instructions will show you how to compile and run a C program from the terminal. In future assignments, you may be asked to develop in this manner so we suggest that you try to get some practice when doing this assignment.

1. Once you've logged onto the desktop, open a Terminal. Terminal is a command-line interface for Linux. On Xubuntu you can do this by clicking the mouse icon in the upper left and clicking Terminal Emulator. Alternatively, pressing Ctrl-Alt-T will open a Terminal.

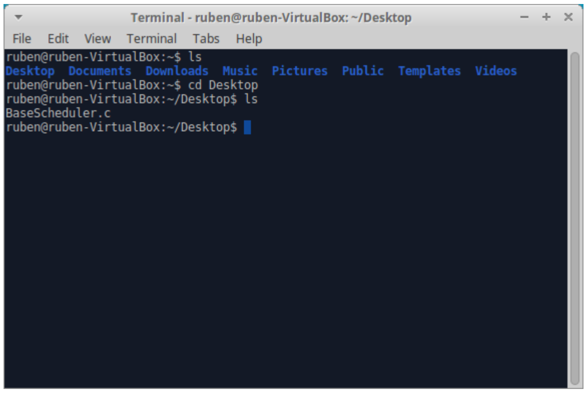


2. After doing this, you should see the following window:


Terminal - ruben@ruben-VirtualBox: ~
File Edit View Terminal Tabs Help

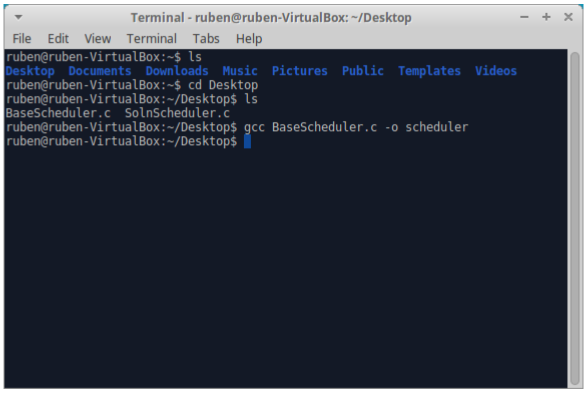
```
ruben@ruben-VirtualBox:~$
```

3. Next you need to navigate to the folder containing your source code. Initially, the terminal will be in your user's home directory. Useful terminal commands are ".." to move to the previous folder, "ls" to view the contents of the current folder, and "cd NAME" to enter a folder called name.


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

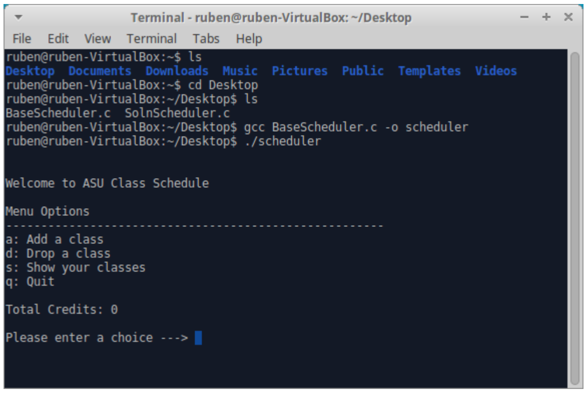
```
ruben@ruben-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
ruben@ruben-VirtualBox:~$ cd Desktop
ruben@ruben-VirtualBox:~/Desktop$ ls
BaseScheduler.c
ruben@ruben-VirtualBox:~/Desktop$
```

4. Here the file we want to compile is called BaseScheduler.c. To do this, we will execute the GCC compiler suite on the source file. The simplest command is “gcc BaseScheduler.c -o scheduler”, which will compile the file BaseScheduler.c and save the execute output as scheduler.


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
ruben@ruben-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
ruben@ruben-VirtualBox:~$ cd Desktop
ruben@ruben-VirtualBox:~/Desktop$ ls
BaseScheduler.c  SolnScheduler.c
ruben@ruben-VirtualBox:~/Desktop$ gcc BaseScheduler.c -o scheduler
ruben@ruben-VirtualBox:~/Desktop$ 
```

5. To run the file we have built, we need to enter “./scheduler”. The dot slash means look for a file in the current folder.


Terminal - ruben@ruben-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help

```
ruben@ruben-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
ruben@ruben-VirtualBox:~$ cd Desktop
ruben@ruben-VirtualBox:~/Desktop$ ls
BaseScheduler.c  SolnScheduler.c
ruben@ruben-VirtualBox:~/Desktop$ gcc BaseScheduler.c -o scheduler
ruben@ruben-VirtualBox:~/Desktop$ ./scheduler


Welcome to ASU Class Schedule

Menu Options
------------------------------------------------------
a: Add a class
d: Drop a class
s: Show your classes
q: Quit

Total Credits: 0

Please enter a choice ---> 
```

4 Submission

The submission for this assignment has one part: a source code submission. The file should be attached to the homework submission link on BlackBoard.

Writeup: For this assignment, no write up is required.

Source Code: Please name your source file as "LastNameScheduler.c" (e.g. "AcunaScheduler.c").