CST8234 - C Programming

LAB 2

LAB OBJECTIVE

By completing this lab, you will learn to:

- Printing to console and asking the user for input.
- Reading input from the user.
- Formatting printf string
- Bonus: loop control flow.

LAB INSTRUCTIONS:

STATEMENT OF THE PROBLEM:

North American telephone numbers structured as follow:

3-digit area code + 3-digit central office code + 4-digit subscriber

For example, Algonquin College number is “(613) 727-4723”, where the area code is “613”, the central office code is “727”, and the subscriber is “4723”. Neither the area code or the central office code

will start with the digit “0” or “1”. See https://en.wikipedia.org/wiki/North_American_Numbering_Plan#Modern_plan for details.

However, despite the consistent structure, people write the phone numbers in a number of different ways. For example, “(613) 727-4723”, “613-727-4723”, or “613.727.4723”. Yet, the presentation always follows the standard structure for phone numbers, so you don’t see a phone number written as “61-37-274-273”.

It is also common to ignore the area code. For example, Algonquin college number would be given as “727-4723”.

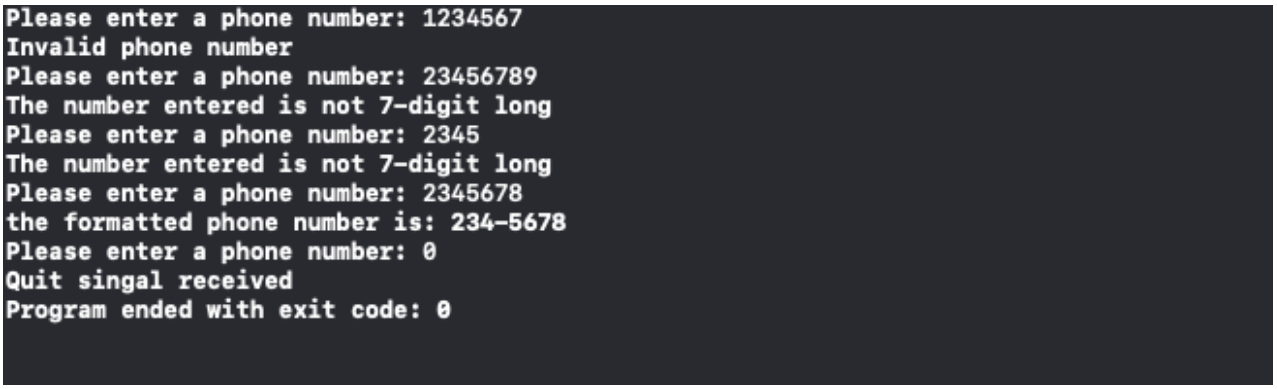
Yet, because the presentation is usually an application or a user preference, and for greater consistency, applications that store/show phone numbers usually saves them in a canonical representation with all formatting removed (e.g., “6137274723”) and then re-apply the formatting when the number has to be presented to the user.

REQUIREMENTS:

In this lab, you will write a program that will read a canonical phone number form the user, then prints it back to the properly formatted in the for “xxx-xxxx”. You program must achieve the following requirements:

1. If the user enters the number “0”, the program should exit immediately.
2. If the user enters any phone number that is grater than or less than 7-digits, the program should report an error back. In other words, only valid 7-digit numbers are accepted, so entering an 8-digits number or 5-digits number for example should trigger an error message.
3. If the users enters an invalid phone number, then the program should report the error to the user. Hint: central office code doesn’t start with “0” or “1”.
4. If the user enters a valid 7-digit numbers, then the program should format the phone number in a hyphenated 7-digit representation, i.e, “7274723” will become “727-4723”.
5. Bonus: You get 2 bonus marks if the program continue looping infinitely until the user explicitly enters “0”.

The following shows figure shows a sample solution output



Sample solution output

There are many ways to solve this problem, one of which involves simple math to separate the central office code from the subscriber code. Yet, any solution that achieves the above requirements will be accepted.

SUBMISSION INSTRUCTIONS

- You can work in a group of maximum 2 students to complete this lab. Individual work is also accepted.
- You are required to show your work during the lab time to the instructor.
- You must submit the source code for the program you wrote. Add all your files under a folder call “lastName-firstName-Lab2”, then Zip the file and submit the zip file only. Make sure to submit all files required to compile and run the program on the instructor machine with any errors.
- DON’T submit any extra file please. For example, the binary file (AKA object or output) files like .exe or .o.
- If working in a group, add a Readme.txt file that contain each student name and student number.
- Brightspace is configured to keep the last submission only. Please complete all steps required to finish the lab before your first submission.

- All submission must be done on the main Brightspace shell, 19F_CST8234_010_ALL or 19F_CST8234_020_ALL, not the lab section one.