COMP 201 - Computer Systems & Programming: Assignment 0

Due: 11.59 pm on October 16, 2020
Corresponding TA: Mandana Bagheri ***mmarzijarani20@ku.edu.tr***

Note: You may discuss the assignment with your peers but the submitted work must be your own original work. Assignments submitted after the deadline are subject to COMP 201 late policy explained in Lecture 1 slides. This is an individual assignment (no partners).

Assignment goals:

This assignment is aimed at making sure you familiarize yourself with basic C fundamentals and building simple C programs as well as basic Git commands and working through Github classroom.

You are asked to accept an assignment on Github Classroom using a link provided in this document and completing a simple C program that involves the very basics of C programming.

As we use Github Classroom to invite you to this assignment, you'd need a Github account prior to starting.

Assignment Description:

After accepting the assignment on Github Classroom using the link, a private Github repository will be created for you to work on (your repository will only be visible to you and classroom admins). In this assignment this repository will include several files:

- main.c: the entry point to the C program
- mylibrary.c: a large open source mock library
- mylibrary.h: the corresponding header file of mylibrary.c
- **myprogram.c:** a simple library with 7 functions
- myprogram.h: the corresponding header file of myprogram.c
- Makefile: the make script
- test/*: automated tests
- README.md: the repository readme file

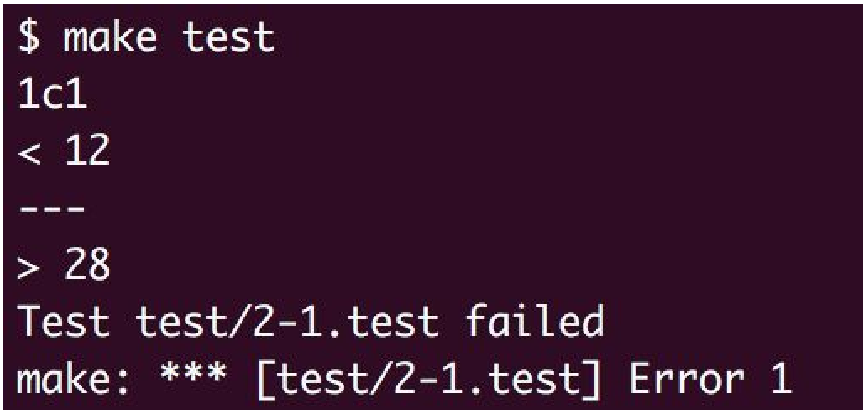
You only need to modify the **myprogram.c** file. In particular, any section that needs modification and writing of code is marked with a //TODO comment. You need to write a short block of code that satisfies the functionality that TODO comment asks for:

```
char alphabet_index(int index)
 {
        //TODO: for index 0, return A. index 1, B, etc. until 25 for Z.
        //if index is out of range, return space ' '.
        return 'A';
}
```

To run the program, you must first build it. The provided Makefile streamlines this process, all you need to do is type *make* and hit enter, and the program will be compiled and the binary file *assignment0* will be created (assuming there are no errors).

You can run this binary file by typing *./assignment0* which will start the program. The program start point is the function main() inside *main.c*. Feel free to read that file but please do not modify anything except *myprogram.c*.

You can manually test your code by running the program and providing it with manual inputs, or use *make test* to run the automated tests that are included in the repository. If any of the automated tests included in the package fail, you will receive an error in the form below:



```
$ make test
1c1
< 12
---
> 28
Test test/2-1.test failed
make: *** [test/2-1.test] Error 1
```

The error specifies which test has failed (which you can look inside test/ directory to see expected input/output) and also tells you what output was expected and what was generated by the program. For example, in the figure above, the expected output was 12 but the generated output was 28.

It goes without saying that most of the tests should fail when you begin the assignment, as the corresponding code is not written yet.

Steps you need to take:

- Accepting the assignment using the invitation link which will result in a private Github repository created for you:

  https://classroom.github.com/a/ixjqiVFR

- Choosing one of the options below to modify *myprogram.c* file in a way that satisfies the requirements of //TODO comments and the tests:
  - You may clone the assignment repository to your local machine and work on it locally.
  - Or, you may use Github Classroom integration with REPL.it which is a cloud based IDE and lets you work on your repository from your browser.
- Committing and pushing the modifications to your private repository frequently using Git commands accompanying appropriate comments (whether you choose to work on the assignment locally or on REPL.it).
- Testing correctness of your implementation using *make test*.

You are required to push your changes to the repository frequently. If you only push the final version, even if it is implemented 100% correctly, you will lose a fraction of the grade because you are expected to learn to use Version Control Systems effectively. You do not have to push every small piece of change to Github but every meaningful change should be pushed. For example, each of the functions coded and tested can be one commit.

It is highly recommended that you do not solely rely on Repl.it as this assignment is a great opportunity for you to get started with the command line environment through interacting with the Github repository.

Grading:

Your assignment will be graded based on correctness, effective use of version control and coding style. On top of the automated grading, assignments are randomly reviewed for originality and plagiarism.

Useful links:

- If you are a Windows user, you can study this short tutorial on how to use Git on Windows: https://www.computerhope.com/issues/ch001927.htm#install
- 35 basic Linux commands that every user should know
- You can also take a look at Unix shell commands on page 15 and 16: http://cslibrary.stanford.edu/107/UnixProgrammingTools.pdf

Good Luck!