CS 211: C and Systems Programming, Spring 2019
Programming Assignment 1: Getting Your Feet Wet with C

1 Introduction

In this assignment, you will practice programming with the C language. Much of your code will come in handy for a sport diver who keeps a log that includes the duration of each dive.

Your task is to write a program that reads in dive times in the form of hh:mm (hours and minutes separated by a colon) and calculates the total dive time (i.e. the sum of all the log times), the average dive time and selects the longest dive time. The calculated outputs are all output in the form of hhh:mm or hh:mm (i.e. hours and minutes).

2 Implementation

Your implementation needs to be in a single C program called `logtime.c`.

Keep in mind that *coding style will affect your grade.* Your code should be well-organized, well-commented, and designed in a modular fashion. In particular, you should design reusable functions and structures, and minimize code duplication. *You should always check for errors.* For example, your program should always check to see if the input is in the proper format.

Your code should compile correctly (no warnings and errors) with the `-Wall` and either the `-g` or `-O` flags. For example

```
$ gcc -Wall -g -o logtime logtime.c
```

should compile your code to a debug-able executable named logtime without producing any warnings or error messages. (Note that -O and -o are different flags.)

Your code should also be efficient in both space and time. When there are tradeoffs to be made, you need to explain what you chose to do and why.

IMPORTANT NOTE: You may write your code on any machine and operating system you desire, but the code you turn in MUST (un)tar (see below), compile and execute on the clamshell.rutgers.edu machine or a zero grade will be given. Be sure to compile and execute your code on the clamshell machine before handing it in. This has been clearly stated here and NO EXCEPTIONS will be given.

3 What to turn in

You have to e-submit the assignment using Sakai. Your submission should be a tar file named `pa1.tar`.

Your pa1.tar file must contain:

- `readme.txt`: this file should describe your design and implementation of the `reschdule` program. In particular, it should detail your design, any design/implementation challenges that you ran into, and an analysis (e.g., big-O analysis) of the space and time performance of your program.
- All C source code including both implementation (.c) and header (.h) files.
- A test plan documented in testplan.txt and code to exercise the test casers in your code.

You can build your pa1.tar file in the following steps:

1. Put all the files you want to hand in in a subdirectory called pa1.
2. In the parent directory that contains pa1, invoke tar:

```
tar cvf pa1.tar pa1
```

The arguments to tar are cfv. The c tells tar to create a new archive file. The f tells tar that the next command line argument is the name of the output file. The v just makes tar list the files it's putting into the archive.

We will compile and test your program on the clamshell machine so you should make sure that your tar file can be extracted on clamshell, your program compiles on clamshell and your exceutable program(s) run correctly on clamshell.

Your grade will be based on:

- Correctness (how well your code works).
- Quality of your design (did you use reasonable algorithms).
- Quality of your code (how well written your code is, including modularity and comments).
- Efficiency (of your implementation).
- Testing thoroughness (quality of your test cases).