COMP 2160 Programming Practices
Lab 4

Oct. 24th - 26th

Objectives

- Testing and (basic) debugging.
- To do this lab you will work in pairs. If you can't find anyone to work with, the lab demonstrator will work with you.

Lab Requirements

Column 4 and **Column 5** of *Programming Pearls* must be read prior to your lab. The key to good testing is to be thorough. The key to good debugging is to think like a detective.

The file `search.c` is test scaffolding written by the author of Programming Pearls. It has been modified somewhat for this lab. The function `binarysearch2` is the 'classic' binary search you learned in first year. The function `test` performs rigorous testing of this function. Note that the author does not use the `assert.h` module, but instead writes his own assert function (as a preprocessor macro).

You need to demonstrate that you can use lldb for debugging in this lab. In other words, you should not try to find the bug by just examining the source code.

Exercise 1

One partner will introduce a 'bug' into `binarysearch2`, and the other will then attempt to discover and fix the bug. Then the partners will switch roles, with the second partner introducing a (different) bug, and the first one finding it.

Exercise 2

Partners may work together on this exercise. In the main function, un-comment the call to the function `testb`, which tests the 'buggy' function `binarysearch2b`. Compile and run the program. You will find that it enters an infinite loop. Use Ctrl+C to quit the program, and try using the `assert` macro to attempt to diagnose the problem. You will probably find that the program still enters an infinite loop, since unlike the `assert.h` module, the program is not stopped by the assert macro. Add the `assert.h` module, and 'comment out' the macro, and see if this helps diagnose the problem.