LAB #7 – Debugging

Remember, you will not receive lab credit if you do not get checked off before leaving each lab. Once you have a zero on a lab, then it cannot be changed because we have no way to know if you were there or not! If you need to get last week's lab graded, you need to show your program to the TAs within 10 minutes of getting to this lab.

(2 pts) Conditional Compilation: One of the useful features of it is the ability to conditionally include code, based on macro definitions. For instance, this is often referred to as a DEBUG macro:

```c
#ifdef DEBUG
    /* your debug code here */
#else
    /* your non-debug code here, if differences exist */
#endif
```

OR

```c
#ifndef DEBUG
    /* your non-debug code here, if differences exist */
#else
    /* your debug code here */
#endif
```

In some cases, you have different function calls for debugged versions, in other cases, debugging code is simple output statements. For this task, ensure that all of your print statements are wrapped in DEBUG macros. Now, you can define DEBUG to turn it on, and comment this statement to turn it off OR you can compile with a -D DEBUG to define it or leave it out.

```
g++ prog.cpp -D DEBUG
```

(2 pts) Bring it all together: For the next part create a makefile with a target for each program. If necessary keep the -D flag in the rule. Do you need a separate rule for a non-debug command for each program?

(6 pts) Debugging and Errors. Here are some example bugs, and you need to find the bug, correct the bug, match the bug to the error below, and explain how you found the bug and corrected the bug.

http://classes.engr.oregonstate.edu/eecs/winter2016/cs162-001/Labs/L7/bug1.tar
http://classes.engr.oregonstate.edu/eecs/winter2016/cs162-001/Labs/L7/bug2.tar
http://classes.engr.oregonstate.edu/eecs/winter2016/cs162-001/Labs/L7/bug3.tar
http://classes.engr.oregonstate.edu/eecs/winter2016/cs162-001/Labs/L7/bug4.tar

To untar the files, tar -xvf bug1.tar

Types of Errors to Find/Correct/Match

· Off by one error in an array
· Initialize a different object than the one you are using
· Forget “Big Three” with dynamic member variable
· Access memory that hasn’t been allocated

1. Trace code to say what it does (plain English)

2. How do you know what the code is doing? What is it supposed to do?

3. Fix the errors and show the fixes to your TA.