Assessment Task Rules

1. You do not *talk* about ~~Fight Club~~ the assessment task
2. You *do not* talk about the assessment task
3. You will be given 2 tasks. Read them *both*.
4. Tasks are **NOT** of equal value. They are awarded different maximum marks.
5. Complete *at least ONE* task
6. You are permitted to complete the easy task, get it marked, then attempt the harder one *if you have time*

Assessment Task Grading

7. A marking guide is included. Read it. It contains the correct output.
8. You are graded on your ability to convert mathematics into a C program which “works”
9. Code style, comments, use of advanced languages features, etc, **are not graded** (yet)
10. Optimisation, however, does carry some weight
11. If your code solves the given problem and produces the correct output you get the marks

Assessment Task Grading

12. Every lab time slot gets a different task
13. Some differences are intentionally subtle
14. If you ask to be marked and your program *exactly matches* a solution from a previous lab you may be asked to justify how you got those results
15. I will be analysing results to measure how much students talk

Assessment Task Timing

16. The task will begin 30 mins after the start of the lab
    - This is to let people find a place in case their lab is full
    - Priority is given to students enrolled in the lab. Demonstrators will have lists.
17. You will have **2 hours**. This allows 30 mins dedicated marking time at the end.
18. You may be marked (and leave) at any time
19. If you leave before being marked you will be awarded zero (unless you feel ill, email me)

Access to Information

20. Live programming support is forbidden:
    - No use of mobile phones
    - No use of Instant messaging
    - No talking to peers
    - No Help from demonstrators
    - No reading from other's screens
21. You may access pre-existing Internet sources (eg: StackOverflow, YouTube, etc) but not actively participate
22. Access to any other source of information is unrestricted

Computer Hardware Access

23. You may use lab computers or a personal laptop to complete the assessment task
24. Code must be compiled with gcc
    - OnlineGDB, Che, and Codeboard.io all use gcc
    - MinGW uses gcc. Native *NIX gcc is also fine.
25. You must make sure your laptop's development environment works
26. Demonstrators are not required to provide technical support for personal machines
    - If yours breaks, use the lab computer
27. Demonstrators should provide technical support for lab computers

Define “Technical Support”…

28. Demonstrators may help you get a C development environment working on a lab computer
29. Demonstrators may **not** help you solve programming problems
    - Your ability to fix compiler errors is being assessed
    - Question interpretation is your problem
        - You need to know how to interpret the mathematical notation in Lab 3