

Midterm 3

This is a solo effort midterm. Any discussion with people, other than the course instructors and teaching assistants, about the content of the midterm is a violation of academic integrity. We take this very seriously.

The midterm is open-book, open-internet, open-compiler, open-static analysis, open AI, and open VM. It's not open-people. You may not post about the midterm to any website. You may not ask questions of anything human driven (quora, stack overflow, etc.). You can of course search for anything you want, but if something helps you, you must cite it in the README.md.

This is a hands-on exam. It is not an assignment. Clarification questions for the questions should be in the eclass discussion forum. But only clarifications will be given. You will not receive "help" or hints from TAs or instructors. You will receive clarifications about formatting and expectations. If you need clarification please post in the discussion forum with a subject that starts with `MIDTERM3` : .

Format

Midterm 3 is 4 questions. Each question is defined in:

- `question1.c`
- `question2.c`
- `question3.c`
- `question4.c`

Each question has a header at the start that needs to be read and filled out. It tells you the intention of each program. The programs fail because they fail asserts or crash.

You are to modify these programs to fix the problems. This will be about 1 to 5 lines of code per question. You can make more radical changes but these programs have small fixes. You might have to change variables, types, loops, structs, unions, definitions, declarations, includes, libraries, code, functions, etc.

You are to make the program pass the asserts. The programs mostly work just fine they just need a bug or two fixed to be great. Your purpose is to debug and fix, not to implement missing behaviour.

The Makefile describes how to build, run, test and lint each question:

```
make questionX # builds questionX executable
make run-questionX # where X is 1 to 4 runs a question executable
make lint-questionX.c # lints questionX.c
make clean eval # cleans, builds, and evals each question
make clean score # runs and scores each question
```

Please modify the questionX.c files. Please do not modify any line that has comment that says DNC. E.g.,

```
// DNC
```

DNC means DO NOT CHANGE. We will be checking if DNC lines have changed.

If you are worried you modified one of these lines try:

```
make dncs
```

Question Marking

We will run `make score` and look at the points.

0 Marks for a question are possible if you:

- A question gets 0 marks if asserts meant to mark the question are disabled. You may add asserts but you may not remove asserts.
- If the expected results are hardcoded into the source file.
- If the indentation is very inconsistent.
- If the head of each .c file was not modified to include your name, ccid, student number, and a reason why you made those changes.
- Any subversion of the marking: return 0 early to skip asserts, etc.
- If your question doesn't pass all of its asserts.

1 Mark is awarded:

- If a question passes its asserts and has an exit code of 0 but
 - that question doesn't lint properly or
 - your program produces warnings when it compiles.

2 Marks are awarded:

- If a question passes its asserts and has an exit code of 0
 - AND If that question passes linting properly.
 - AND that question produces no warnings when compiled

```
totalMarks = question1 * (question1 + question1Lint) +
              question2 * (question2 + question2Lint) +
              question3 * (question3 + question3Lint) +
              question4 * (question4 + question4Lint)
```

where questionX is if the questionX passes its asserts and has an exit code of 0

```
AND has consistent indentation
AND does attempt to subvert marking
```

AND has name, ccid, student number, and reasoning in header
where questionXLint is if questionX lints successfully and has no warnings

We will check if asserts are missing.

Marking will be using the VM.

Submission

Tar it up!

Make a tar ball of your midterm. It must not be compressed. The tar name is `__YOUR__CCID__-midterm3.tar`

the tar ball should contain:

- `__YOUR__CCID__-midterm3/` # the directory
- `__YOUR__CCID__-midterm3/README.md` # README filled out with your name, CCID, ID #, collaborators and sources.
- `__YOUR__CCID__-midterm3/question1.c` # C program
- `__YOUR__CCID__-midterm3/question1.c.dnc` # Asserts
- `__YOUR__CCID__-midterm3/question1` # executable
- `__YOUR__CCID__-midterm3/question2.c` # C program
- `__YOUR__CCID__-midterm3/question2.c.dnc` # Asserts
- `__YOUR__CCID__-midterm3/question2` # executable
- `__YOUR__CCID__-midterm3/question3.c` # C program
- `__YOUR__CCID__-midterm3/question3.c.dnc` # Asserts
- `__YOUR__CCID__-midterm3/question3` # executable
- `__YOUR__CCID__-midterm3/question4.c` # C program
- `__YOUR__CCID__-midterm3/question4.c.dnc` # Asserts
- `__YOUR__CCID__-midterm3/question4` # executable
- `__YOUR__CCID__-midterm3/Makefile` # Makefile
- `__YOUR__CCID__-midterm3/eval.sh` # shell script
- `__YOUR__CCID__-midterm3/floatest.h` # header
- `__YOUR__CCID__-midterm3/bitstring.h` #header
- `__YOUR__CCID__-midterm3/VERSION` #version of midterm

Extra files such as the test files are allowed to be in the tar file. Any file we provide you in the release tar is OK to be in your tar file.

Submit it!

Upload to eClass!

Marking

This is a 8-point midterm. It will be scaled to 9 marks. (9% of your final grade in the course: A 8/8 is 100% is 9 marks.) Partial marks may be given at the TA's discretion.

- You will lose all marks if not a tar (a `.tar` file that can be unpacked using `tar -xf`)
- You will lose all marks if files not named correctly and inside a correctly named directory (folder)
- You will lose all marks if your C code is not indented. Minor indentation errors will not cost you all your marks.
- You will lose all marks if your code does not compile on the VMs or the lab machines.
- You will lose all marks if `README.md` does not contain the correct information! Use our example README!
 - Markdown format (use `README.md` in the example as a template)
 - Name, CCID, ID #
 - Your sources
 - Who you consulted with
 - The license statement below

We reserve the right to modify the exam and provide errata as the exam commences.

License

This software is NOT free software. Any derivatives and relevant shared files are under the following license:

Copyright 2020 Abram Hindle, Hazel Campbell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, and submit for grading and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- You may not publish, distribute, sublicense, and/or sell copies of the Software.
- You may not share derivatives with anyone except UAlberta staff.
- You may not pay anyone to implement this assignment and relevant code.
- Paid tutors who work on this code owe the Department of Computing Science at the University of Alberta \$10000 CAD per derivative work.
- By publishing this code publicly said publisher owes the Department of Computing Science at the University of Alberta \$10000 CAD.
- You must not engage in plagiarism

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.