CS4390/5390 Applied Numerical Computing for Multimedia Applications Spring 2025. Homework Assignment 1 Due: 02/27/2025 11:59PM MST

In class, we have discussed several algorithms to add, subtract, multiply and convert integer numbers, as well as to divide them by a constant, typically 10. We have not yet implemented all these algorithms and we will not implement them together. We will rather discover an existing implementation in this homework assignment. In addition, we have not yet discussed the details of implementing floating-point formats based on integers. As you will see in the code, floating-point addition, subtraction and multiplication are very straightforward to implement based on existing integer operations. The only thing missing will be floating-point division and square root. We shall discuss these topics on Tuesday 02/25/25, in class.

For this first homework assignment, your task is to:

- Set up the library code environment, as it is available in a git repository. This step is detailed in Section 1.
- Add testing for the functions we have written for arithmetic with integers. This step is detailed in Section 2. If you find a bug in the code during testing, you must correct it and commit the corrected code back.
- Work with your classmates to distribute the task of presenting every integer and floating-point function in the code on Tueday 02/25/25 and/or Thursday 02/27/25 in class. This step is detailed in Section 3.
- Write a short report¹ on your testing strategy, the bugs you have found etc.

Your deliverables are:

- A modified environment that includes more testing than what is currently available in the git. You can submit this work in the form of a code archive by email to utep-numerical@christoph-lauter.org or by committing it to a *branch with your name* in the git and pushing back to origin. This latter way of submission is preferred.
- A PDF report with your testing strategy, description of bugs etc. You can submit this report by email to utep-numerical@christoph-lauter.org.
- A reasonable presentation of "your" functions in class on 02/25 and/or 02/27.

1 Code Setup

Clone the git available at https://gitlab.com/cquirin/utep-numerical-computing-2025, ideally on a Linux system². Refer to the internet for HowTos on git cloning. Document the steps in your report. Compile the code with make. Run the existing tests with make test. You can delete the compiled object files, libraries and executables with make clean. If you encounter issues, come to your instructor's office hours and try to find solutions. Document the issues and solutions you have found.

Once everything compiles and works as it should, proceed to the next step, the testing part.

¹In the format of a PDF document. No Microsoft Word, resp. convert to PDF in the end.

²If you need access to a Linux system but you cannot find any, ask your instructor. He has a system available with access over the internet.

2 Testing

Start by creating a branch of yours in the git. Refer to the internet for HowTos on git branches. You should name your branch with your name.

Modify the Makefile and the tests/test_integers.c files to add more tests. At the end, you need to cover all the functions declared in include/integer_ops.h, leaving out the integer division. For each function, you need to have at least 8 test points (inputs). More is better.

Document the tests, your reasoning behind the tests etc. in your report.

Do not forget to commit your work to the git and to push your branch back to origin³.

3 Presenting the algorithms

32 additional There are functions in integers/integer_ops.c and 16 functions in widefloat_widefloat_ops.c. Not all functions are equally important and equally complicated. Some functions go in groups; when one function is presented in detail, the other functions in that group can just be summed up. Work with your classmates on Thursday 02/20/25 to decide who presents which function and in which order and how to present the functions. Prepare your own speech, addressing the most important points in the code of each function. Do not paraphrase the code! If possible, get feedback from your peers on your first presentation run, while your instructor is absent.

³If, really, you cannot figure out git well enough, you can submit an archive by email.