

Computer Systems Organization
CSCI-UA.0201 Spring 2024
Programming Assignment 2

Due Friday, March 8 at 11:55pm

This assignment helps prepare you for the midterm exam by making you (1) manipulate bits of variables, (2) understand how integer multiplication and floating point addition is performed, and (3) write some simple assembly code.

I have provided some of the code and step-by-step instructions in the comments. I have also provided a few hints in a document called Hints.pdf.

Download one of the following compressed files appropriate for your computer:

- assignment2_cygwin.tgz for Windows/Cygwin.
- assignment2_macos.tgz for macOS.
- assignment2_linux.tgz for Linux.

Save the downloaded file in the directory where you want to write your Assignment 2 code. To uncompress the file, in a shell (e.g. Mac Terminal or Cygwin), type

```
tar -xzf filename
```

where *filename* is the name of the file that you downloaded.

The five files that are extracted from the compressed file are:

- assignment2.c, a C file that you will be adding code to.
- sum_squares.s, an assembly language file that you will be adding code to.
- sum_squares.h, a header file that you should not change.
- makefile, a file that makes compiling quick and easy (do not change this file).
- The Hints.pdf file containing helpful hints.

You should fill in the missing code in assignment2.c, where you will need to write functions for reversing the bits of a 32-bit integer, multiplying two integers by using shifts and adds (as shown in class), and adding two floating point numbers without using the built-in floating point addition. You simply need to follow my step-by-step instructions in the comments.

You will also need to write assembly code in the file sum_squares.s, within the function sum_squares. Again, you just need to follow my instructions in the comments.

Because of the makefile file that I provided, to compile your code, you simply need to type

```
make
```

into the shell (e.g. Mac terminal or Cygwin). This will invoke gcc for you.

Then, to run the code, type

```
./assign2
```

However, if you just want to compile the assignment2.c file but not the sum_square.s file because you haven't inserted the assembly code into it yet, then compile as usual:

```
gcc assignment2.c
```

and run it by typing **./a.out** (on macOS or Linux) or **./a** (on Cygwin). Once you're ready to compile with sum_squares.s, then follow the instructions for using make, above.

When you are finished, upload your versions of assignment2.c and sum_squares.s to Brightspace.

As always, you should write your own code. You may work with other students to figure out how to approach the problem, you can even ask other students for help. However, if you don't write your own code then you will not be able to do well enough on the exams to get a decent grade in this course.