

CST520 Project 1-A: Sorting Competition

Due: Sept 10th 9 am

The goal of this project is to test your ability to write and optimize PLP programs. Your team will be given skeleton code, which has calls to functions you will need to write. These functions will be implementations of different sorting algorithms that will perform an in place sort on arrays in memory. This project will end in an in class competition. To the victors go the spoils, extra credit points (each competition is worth 2 points). Each algorithm implementation will compete for two separate prizes. One for the implementation with the fewest number of instruction (comments, labels, and white space don't count as instructions) and one for the implementation that takes the fewest cycles to execute.

Skeleton code has been provided for you to fill in the three required functions: bubble sort, insertion sort, and a wildcard sorting algorithm (this is an implementation of any sorting algorithm of your choosing). Each function should be implemented in its corresponding ASM file where stubs have already been added (functions should use the `return` instruction that is already located in the stubs). The source files, *main.asm*, *input_and_verification.asm*, and *output.asm* should not be modified.

When the program is run it waits for a list input from the UART. The format for input is a list of positive decimal values separated by a single space and ending with a semicolon (;). For example, the following is a valid input:

```
250 10 2015 4 1;
```

Each of your functions will be passed an array starting address and the number of elements in the array using the registers `$a0` and `$a1`, respectively. It should sort the array in-place (wherever it is located in memory) from smallest to largest. Arrays will then be validated to ensure they are sorted and then an output will appear on the UART. The output for each algorithm will either be a running sum of the cycles used in an algorithm since the program started, or the number of times a sorting algorithm has failed validation.