

# CSE 341: Computer Organization

## Report (Spring 2022)

Name (Last, First): Peng, Winston

UB 8-digit number: 50364686

### Project A Report: MIPS Assembly Programming

1. Describe your implementation in detail for the Minimum-Maximum problem. (less than 200 words)

First, it creates a bunch of global variables that will store important data, including IN which stores user input, MAX which stores max value, and MIN which stores min value. The program then asks for user input with a syscall, using a loop. It iterates through the value stored into IN, by incrementing by three, and grabbing the next two values each time. The two values are then subtracted by 48 in order to translate them from the ascii value into the decimal value, and then converted into decimal form in order to be compared to both the MIN and MAX values. If it is larger than the MAX or smaller than the MIN, then that value will be written to that global variable in place of what the original value is. The exception to the previous rule is the first iteration, where values are automatically written to the MIN and MAX values, since the loop counter is equal to 10 (for the 10 input numbers). Finally, after getting the min and max values, they are written to the screen with syscall, with a space between them.

2. Describe your implementation in detail for the Sorting problem. (less than 200 words)

For this question, it first creates an array to store the user input into. This is because unlike the previous problem, we will need to access all values of the user input multiple times, so it prevents needing to parse the input data each time. The parse code is very similar to the previous problem, though in this problem, the value is then stored into the global array variable. After being stored into the array variable, the array is then sorted. The array has two counters to keep track of the fact that there is not only a loop, but also a nested loop. For each iteration, the entire array is traversed, and for each index of the array, we compare it to the one next to it, and swap them if the one on the right is larger. By repeating that iteration 10 times, we guarantee that all the values in the array end up in their properly sorted positions.

3. Describe your implementation in detail for the Bonus problem (If attempted). (less than 200 words)

4. Highlight major challenges that you faced while working on Part A of the project (if any). (less than 200 words)

Parsing the data was quite difficult. Most other languages make it easy to convert from the user input string to an integer, but in mips we had to deal with ascii, and alignment in order to properly translate the input.

5. Please mention any comments or suggestions for Part A of the project (If any). (less than 200 words)

Zybooks didn't cover enough info on MIPS (or at least about writing a MIPS program from scratch). Another recitation about MIPS could've helped.