

# CSE 341: COMPUTER ORGANIZATION

## Project Report (FALL 2020)

Project Due: December 10, 2020

1. What problems did you encounter during development? How did you solve them?

The first problem I encountered was how to know how much space to allocate. I achieved this by allocating enough space for a single int to get num\_points then allocate more space base on how many points were required.

Another problem I had was when sorting. I was trying to bubble sort, but I was struggling to keep track of where I was within the loops. I fixed this by using recursion and adjusting the base address and num\_points after each call.

2. If the input data is given in a text file instead, what would you do while loading points?

I would need to figure out a way to store the hex character as 8 character groups while removing spaces. I assume that the hex character groups would only be character and not be int, so I would need to convert those strings into numbers.

3. Do you have any good practice during developing the project? Summarize them.

I stored \$ra on the stack when jal was called within another function label.

I used recursion to simplify the code.

I used comments to make the code more readable.

I used call labels.

4. Do you have any suggestions regarding this project?

One suggestion that isn't about this project specifically is that I think it would be good to have a much smaller 5 point assignment a few weeks into the semester. I struggled a lot with the assembly questions before the midterm and this project help clear those problem up a lot.

Other than that I think this project had an appropriate difficulty for someone that had no prior experience of using MIPS other than in this class.

5. (Bonus) Compare your codes with the codes generated by the compiler, describe your steps including environment setup. What did you find? Which version do you think would provide better performance?

First I had to set up the gcc environment. Then I programmed some of the functions in C and used Disassembly Explorer in VScode. From what I found it is clear that the compiler doesn't care about readability. There are things that the compiler is doing that is hard to understand. That being said, I have to assume that the compiler is significantly more efficient because there has been 50 years of making the compiler more efficient.