Implementation

Before implementing a new function or sub function, I would first think about what purpose it serves, and during implementation make sure that it only serves that one purpose. When implementing a full solution, I would first think about what sub functions I would need in order for it to work. I would first work on making my solution work for ideal inputs, then start implementing rules in order to limit and control how the inputs are handled. For example in challenge 3, I first made a function that prints out whatever expression is inputted, after I made sure this worked, I then implemented recognition of sub expressions being macros, then their definition order.

Testing

After implementing a new function or piece of code into my work, I would first check that it would compile. Next I would open the new function on its own, or with any necessary sub functions which are also tested in this way, in ghci, I would then proceed to test it with every type of input I could think of, making sure it returns errors when it should and returns correct outputs when needed.

Once I had considered my function complete, I would collaborate with fellow computer science students to share various unusual inputs, we did not share our code or ask for help with debugging. Using these inputs I would first work out myself what the expected output would be, then ran it on my function. If the output was unusual, I would read through the code and step by step enact what it would do at each step, this would end with me finding the issue with my code most of the time.