This project was a step forward in terms of our collaboration with assembly coding. Project 1 had us creating a MIPS program that implemented the Fibonacci sequence with any input including overflow exceptions. This time project two has us implementing Booth's Algorithm for 32-bit multiplicand/multiplier, and printing the product as a binary, 64-bit integer with the only difference this time paired programming is a requirement.

To start our project, we created a public repository that all three of us can work on and make edits to our program as we go. For further communication we created a group me before spring break so we can tell each other different tasks that needed to be done.

As far as programming the algorithm itself we were able to see each other's edits or commits we have made. We "switched roles" by this method of collaboration. Some of the bugs we encountered were mostly minor issues such as, adding a new line to the end of our code or initialization errors that were effortless to fix. Following the Algorithm was extremely easy because the flowchart gives a clear path on where to start and take when symbols like A, Q, or Q-1, are certain binary combinations. Each iteration requires a shift right action so implementing the action was fairly simple.

Compared to project 1 this project mainly tested us on our communication skills seeing that this project was a lot harder to implement than the Fibonacci sequence from before. Overall, we handled this project surprisingly well.