Describe a mistake you made or error message you got while solving the problems, and how you will recognize in the future that you made the mistake again.

- I actually wrote an infinite loop accidently within the fizzbuzz_adv () function. I was calculating powers for both x and y, but when n was 0 the function just kept running as it called valuation(n,d). I changed the order of the fizzbuzz_adv () function so that it evaluates if n is <=0 and if n is a float rather than an int and assigned a result of "invalid". Through the console I learned how to monitor if the cell that I am running is still running because I kept moving forward in the file, but didn't realize for a while that after I hit shift+enter that the cell just kept running and never stopped. I will try to also incorporate this line of thinking into the pseudocode that I write, so I address these corner cases earlier in the code.

What makes a good test case for determining if a function works correctly? How do you know that your functions perform correctly?

- I think it is important to utilize the pytests, but when the assert statements failed then I would walk through each assert statement manually by running the function with those values. I could then see which assert was failing and start debugging the problem. I also think it is important to consider corner cases when writing and using test cases. The 0 for valuation was a good corner case for me to compensate for and forced me to rewrite the internal workings of the function I was working on.

Did you make any changes to your mortgage calculator other than package it into a function?

- I still used the same calculations for the most part. I did create a function to make it so the calculations inside amortization () were more readable in nature. I also changed the variable names to what they are versus one or two letter representations. This makes the main function more readable. I also added a return None, None statement for when interest is greater than the monthly payment as certain unit tests were not working.

Does the returning of None, None in the infinite loop sufficiently communicate to the caller that there was an issue with the inputs? What might be done differently?

- I still kept the statement "You have to increase your current monthly payment to be able to pay off your loan" in order to direct the person submitting inputs that something needs to be adjusted otherwise the amortization() function would not work properly.

Describe one thing that you learned while working on this lesson that stood out as useful or interesting.

- Obviously writing functions to make your code more clean and thus more understandable was a great learning for me. Also running your pytests as you go rather than at the end has made this more efficient towards finishing the assignment without so many surprises. I also kept writing pseudocode and working through several examples on paper as I wrote code to trace through what I would expect the code to do.