Assignment 3 - Reflection

For this assignment, I sought to openly communicate with ChatGPT's open-source model for assistance. Through this process and after thoroughly analyzing my final code, I gained a better understanding of Python's lingo. Since it had been a while since I last engaged with coding before this class and I had never played around with Python before this assignment served as quite the refresher on computer language, since it all tends to generally be the same. Out of the three assignments so far, this one required the most problem-solving.

My code begins by implementing a Python-based calculator that my custom Mathematician Bot can later refer to for self correction. The calculator is programmed to repeatedly multiply the base result by itself based on the number of iterations provided by user input, as required by the assignment. I then prompted the AI to act as a mathematician with a humorous personality, performing the same calculations as the Python program but by itself. The temperature setting was adjusted to be open to a variety of math-based jokes.

I encountered several issues while setting this up. The main problem that I ran into was getting the AI to recognize whether its answers were correct or incorrect. Because the AI kept generating a bunch of text in its responses, python's result would compare the results differently. It took me a few hours to work around this, but I found a rough solution: extracting only the highest numerical value from the AI's output while filtering out all other words and punctuation. Since the largest number in the response was bound to be the right one, this method allowed the AI to spew math jokes that still referenced smaller numbers.

Another challenge was introducing a level of error. If the AI was always correct, it wouldn't be able to make mistakes. Fortunately, due to my faulty programming skills, this happened naturally. For some reason perhaps due to the temperature or top-p settings my bot struggles with large numbers. Typically, any result over a million exceeded its capacity to process accurate results. Finally, I added a self-deprecating mechanism where my bot eventually rage-quits and self-destructs after making three consecutive mistakes.