While both the object-oriented and functional programs accomplish the same task, I had to go about making them in entirely different ways. In the object-oriented version, I utilized a lot of classes and for loops. For the functional program, I had to really change my design process to make it follow the proper functional programming paradigm. I changed all of the for-loops to recursive functions and converted the classes to pure functions that directly return a value instead of iterating through the code and appending things to a list. It was easier to figure out problems in my functional program, because everything was separate, so I could easily tell where a problem was occurring. It was a little difficult to figure out how to combine my classes with a lot of different methods into one concise pure function. For the OOP version, my thought process centered around figuring out what classes I needed to make and what methods to give them. For the functional version, I had to focus more on small steps and deciding what functions I needed to create, and what data those functions needed to return. OOP felt more like an interconnection of a bunch of different classes analyzing the file, while the functional version feels more like a step-by-step analysis walking through the script. Overall, I think both OOP and functional programming can be very useful, depending on what the goal of the program is. I personally prefer the object-oriented version of this program, since I preferred using for-loops to iterate through the script rather than recursive functions. However, the functional program is easier to read and it’s a lot clearer to see what each function does.