|  |  |  |  |
| --- | --- | --- | --- |
| Name | Input Parameters | Returns | Purpose |
| modifyImage | 1. Choice of user in a string 2. numRows of the imageArray as an int 3. numCols of the imageArray as an int 4. the imageArray as a list for the image 5. highIntensity is the result of finding the max intensity as an int 6. average is the average intensity as an int | Nothing | For modifying the image based on the user’s choice. Placed into an if statement to prevent each effect from manipulating directly onto each other’s array. |
| readInFile | 1. filename reading in as a string 2. imageArray to store data | Nothing | Open the filename and read in the data line by line into the imageArray |
| writeOutFile | 1. filename reading in as a string 2. imageArray to store data | Nothing | Writes the data into the output file from the imageArray |

I chose these lines to convert into functions because I wanted the main focus of the main file to be on the 4 statistics computed. The rest of the outside code was just there to support the main analytics that we wanted to compute. I took the suggestions of the instructions, but I was also just planning on making these lines into my functions anyway because of my view of having most of the main be dedicated to the analytics. I was originally going to have the modifyImage function be 3 separate functions, but decided to combine them into one function as it was just going to be 3 separate functions called after each other in a row.

The functions are general enough for the scope of this project, but they would likely have very niche use cases outside of this project because of differences in how people have stored their data and the modifyImage requires prerequisites that you’ve calculated maximum intensity and average intensity to use the function beforehand. The read in and write out functions I think are pretty standard for people to have as reading and writing files are pretty common.

Python is great because it has a very large and useful built-in function library. The python functions are probably way more general as they were developed for most every use case under the sun, while my functions were only specifically designed for within project use. I do not image myself using .pgm formatting in the future as I had never heard of it before this.

I am happy with how the functions have cleaned up the readability of the main file as the main focus was on the 4 statistics calculated. The outside helping aspects of the file are all contained within functions minus the simple entrance and exit print statements. The modularity of the file is straightforward with just the opening and exiting being in their own box that can be broken off, and the modify image is also self-contained and can be skipped through giving the prompt bad input. As for maintainability, there already wasn’t much repeating aspects of the code, but the functions are definitely easier to navigate and find, rather than scrolling down on one big main file. Also it is easier for testing the main file now, as one can simply comment out one line of code if they want to skip over it, rather than commenting out 10 lines of code and making sure to keep them together.