TODO:

- Fix some file and input issues with app, examine architecture again -- input issue still an issue, but file can be switched now
- Color picker - would be nice if it didn't reset the app back to the home screen
- ability to edit a file or save
- solve the issue of accepting inputs through the GUI
- The outputs from the test files are also incorrect, it appears to be outputting the WRITE command itself, like +1115 instead of writing the content of memory address 15 like it should.
- Design Document Updates: wireframe isn't quite accurate (you don't have the Load File button any more, and the Options Menu should be outlined in detail) -- also still need updates to use cases and requirements. all the docs need to be kept up to date from milestone to milestone
- README needs to be up to date with your current app (even if the app itself isn't up to date with milestone features).

Person A: Katie Hancock

- Research how best to fix the input issue (accepting inputs from the GUI, not the console, when needed by the running program)

- Change the execution engine to support 6 digit files instead of 4. (Loading 6-digit files, parsing them, getting them into your internal memory, then executing them correctly -- remember this involves math operations and overflows in addition to the three digit function code and address space)

- Research and write the function to convert a 4-digit file into 6 digit format. (write this separately first as a separate Python program that loads a file, parses it, converts it, and saves it to another file, then test it separately, without adding it to the GUI code yet)

- Update the GUI wireframe document with the current state of your GUI (may need to wait until later to get accurate details on what the GUI will look like with everything else added).

- If needed, update the class definition document as well with any changes to your classes and/or architecture to keep it up to date with milestone 5.

Person B: Margo

~~- Afterwards, (after testing) adding that 4 > 6 conversion function to the GUI (on a button or menu item handler). **I believe for your group, it would be better to take this approach: have a separate 4 > 6 conversion option rather than adjust your execution engine to support both types. That way Person A can focus on just 6-digit execution in the engine itself, and Person B can work on the conversion as a separate feature apart from the engine execution**)~~

– no longer needed since conversion is automatic

-validate

- Update the README to be more comprehensive and include details on how to use all GUI features (may have to do this last, so you know how the GUI ends up looking and how things work)

- Update the use cases in your design documents to make sure you have one use case for each new feature from milestones 4 and 5. (Probably 5-6 new ones total)

- Update the requirements doc in correlation with your use cases, to make sure that all new milestone 4+5 requirements are represented in your team's requirements doc when submitted.

Person C: Jarrett
- Add "Save" and "Save As..." functionality to the GUI -- figure out how to take the display content from your GUI and save it off again as a file under a new name.

Person D: Nick
- Research and implement the ability to edit (including cut/copy/paste) the content in your GUI display so the user can do manual edits before execution. (Will need to work with the person doing the Save/Save As functionality as the user will want to be able to save the edited contents to a file)