I used Python to create a MIPS assembler. First, I created 3 Python dictionaries that hold R-Type function codes, I-Type opcodes, and register numbers. This minimizes conditional statements for all of the different instructions by making it simpler to find the correct codes. It begins with parsing branch labels in the assembly file for their names and locations and stores that information in a Python dictionary. This is necessary because an instruction may branch to a label that has not been reached yet, so a preliminary label search solves that issue.

Next, I parse the file once again to read in instructions.