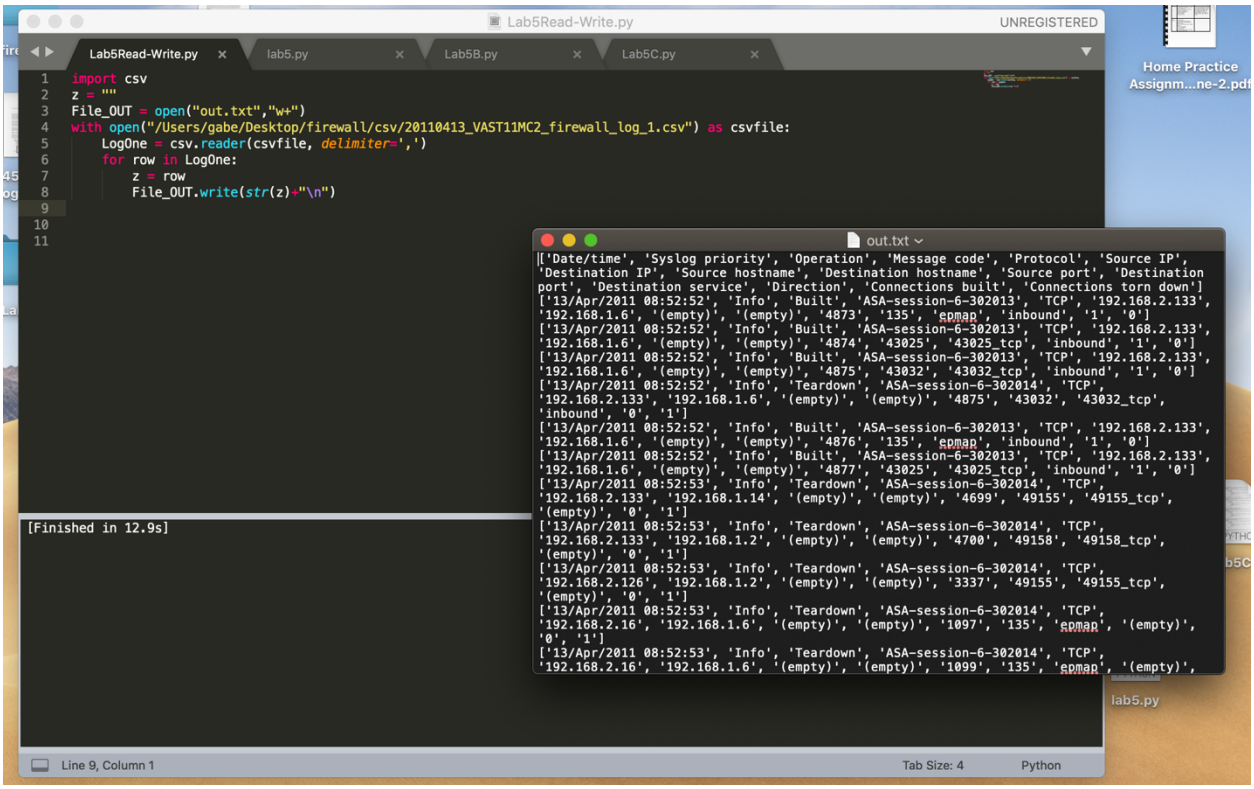
Gabriel Naples
4/3/2019

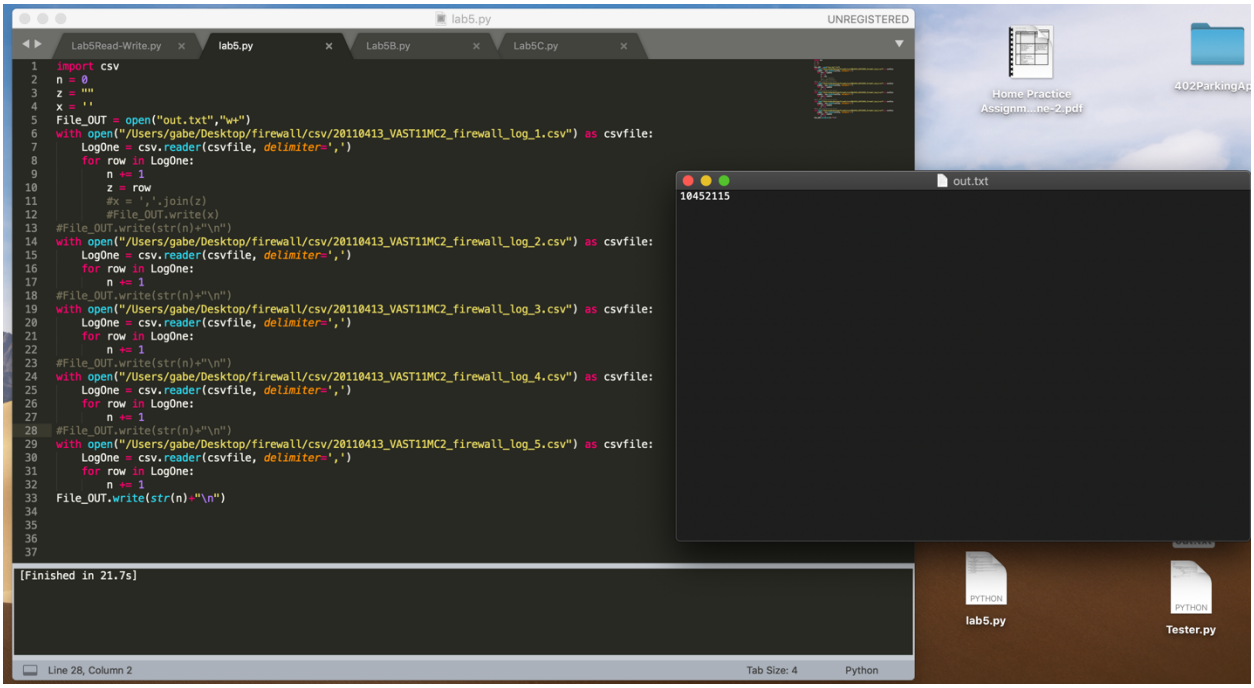
Option 2 – Programming

Part A:

Below is a screencap of my program going through the CSV file for the first log and reproducing the information, line for line, in the output.txt file. This basic program could easily run through every single log file and reproduce them line by line with some slight modification; however, I chose to only do it for the first log file since the file for all 5 logs would be massive and I would be unable to open it. I could also output each log into its own unique out.txt file following the same logic below.

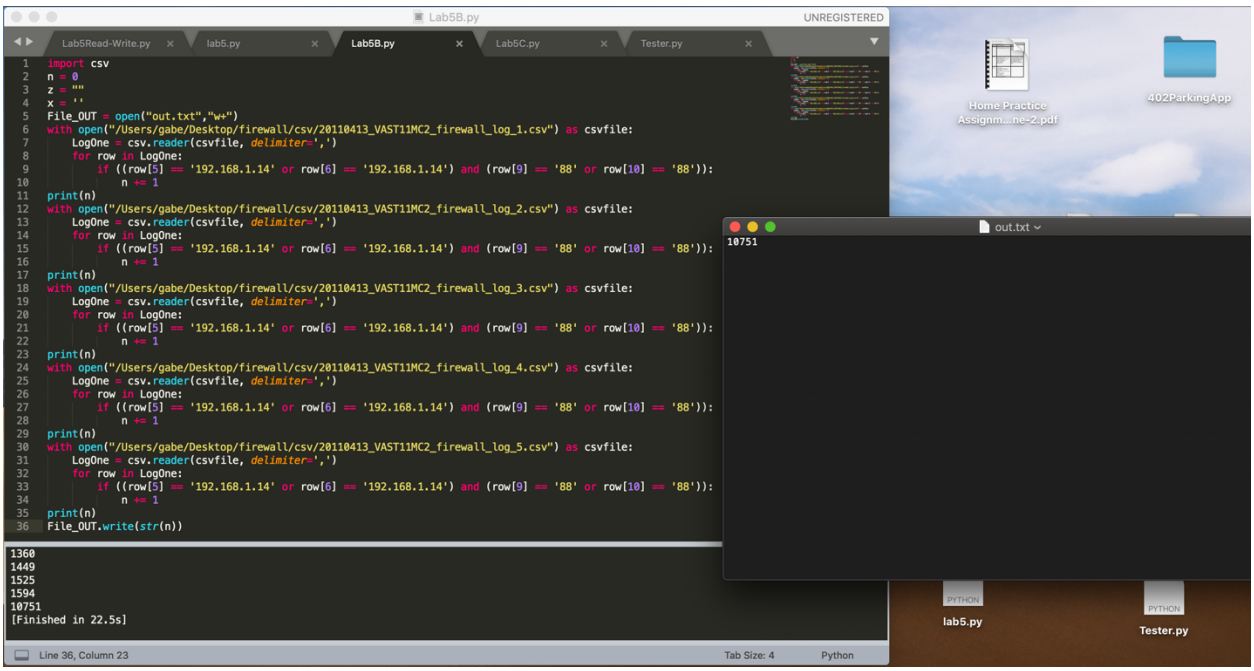


The below code goes through each file and tallies the total number of connections. In the output file it prints the total number of 10,452,115 connections. I could also vary this program to print the total amount of connections contained within each log file individually.



Part B:

Below is the code that goes through each document and counts the total number of connections that fit the specified criteria (containing IP 192.168.1.14 and port number 88 with both destination and source ip and destination and source port). The code then creates a running total of the amount of connections that fit this criteria and outputs only the total amount of connections to the out.txt file. Within the program itself it keeps a running count as it goes through each document which I used for testing purposes. Furthermore, I could add an additional line in each for loop iteration that prints out the whole row of the information desired.



Part C:

Below is a screencap of the program running and creating the table of the amount of times that IP 172.20.1.5 and port 80(for both source and destination IP as well as source and destination port) appear and how many times per hour. This program goes through and creates a count of the amount of times the criteria appears while constantly checking that the hour matches. Once the code sees that the hour no longer matches, or has changed, it outputs the total for the hour it was keeping track of and resets the counter back to zero. Then it goes through and starts counting the amount of times the criteria match for the new hour. It is important to note that the first line in my output is 0 because of the way I created my buffer variable for recording and iterating through different hours.