SY201 Name(s): _____

Alpha(s): _____

Assignment Type:	Lab	Collaboration Policy:	**Discuss Only**
Assignment Title:	Lab 8	Submit Project Name:	lab8
Electronic submission due: _____ Paper submission due: NO PAPER SUBMISSION Submission instructions: http://courses.cyber.usna.edu/SY201/calendar.php?load=policy			

1. Assignment Overview

In this assignment you will build a command line program that processes log files. You will use the new topics of command line arguments, file I/O, and exceptions in this lab.

This lab assignment will be very different from previous ones in that the internal structure of your program will not be specified by the assignment. **You** will have to decide how to break up and solve the problem to achieve the desired functionality.

2. Background

a. We will be using three log files for this lab: access3.log, access5.log, and access7.log. They are all linked from the lab page. You will need to download them and have them in the same directory as your lab8.py file to test the functionality of your program. Each line of each log file represents a communication from some computer (various clients) to a single web server. The IP address of the client is the first entry on each line of the log files.

b. There will be a few times that we ask you to print out the key-value pairs from a dictionary in order, this means with the keys sorted. You can loop over the sorted set of keys from a dictionary with the following code:

```
#key is a variable name and dictionary is the name of your dictionary
for key in sorted(dictionary):
```

The built-in function sorted() returns a sorted copy of the sequence passed as an argument. The call of sorted(dictionary) will return a sorted list of keys from dictionary.

c. The exception raised when trying to open a file for reading (file I/O) and the file does not exist is a **`FileNotFoundError.`**

d. We have linked a working version of lab8.pyc so that you can test it to help you understand the behavior this assignment is looking for. This is a compiled Python file. You can run it the same

way you would run a non-compiled Python file from the command line (e.g. python3 lab8.pyc), but you will be unable to see the file's code. Reminder: code decompilers are NOT authorized.

3. Specification.

Don't forget:
- Write your alpha and section number in a comment at the top of your program
- Identify anyone you discuss the assignment with in a comment at the top of your program
- Include comments adjacent to each function you write that describe the function's purpose
- Your program should print out nothing additional to what is called out below

Your program is going to be a command line program that support the following functionality:

a. Processes log files and maintains a running tally of how many connections each host IP address has made to the server across multiple program runs. For example, I should be able to run the program to process access3.log and then have the program end. I should then be able to run the program again to process access5.log and the program should incorporate this new data with the previous data from processing access3.log. To achieve this functionality, you will need to save the work your program has done into a file before your program ends and read that data back into your program when your program starts again.

Your program will also need to have the functionality to:
(1) print the entire running tally of data that has been processed
(2) identify the host that has connected the most to the server in the logs that have been processed
(3) delete the current running tally of processed data (think of this like a reset)

More specifically...

b. When run with the command line argument '-p' (for print):

i. Your program should print out each unique IP address it has processed so far and the number of times that IP address has connected to the host and then terminate. When printing this data, the IP addresses should be sorted using Python's built-in sorted() function. The output should be formatted like (IP address on the left, then a space, then a colon, then a space, then the number of times connected):

1.1.1.1 : 2
2.2.2.2 : 5
3.3.3.3 : 10
… (for all IP addresses processed)

Execution syntax: python3 lab8.py -p OR python3 lab8.pyc -p

c. When run with the command line argument '-d' (for delete):

   i. Your program should delete the current running tally of data and then terminate. The next time your program runs and processes a log file, it should start from a blank slate in terms of counting IP addresses and the number of connections.

Execution syntax: python3 lab8.py -d OR python3 lab8.pyc -d

d. When run without the arguments '-p' or '-d':

   i. Your program should process the <u>one or more</u> arguments as filenames and attempt to process each of those log files. Processing a log file entails counting each connection from a host IP address and incorporating it to any previously processed and/or stored data. The same log file can be processed more than one time.
   ii. Before terminating, your program should output the IP address or IP addresses that have had the most connections to the server in the following format. Only print out multiple IP addresses if they are tied for the most connections.

The most common connection(s):
91.121.31.184 with 1490 connections

   iii. Note: your program should NOT crash even if the filenames provided are invalid. See Paragraph 2.c.

Example execution syntax: python3 lab8.py access3.log access5.log
OR
python3 lab8.pyc access3.log access5.log