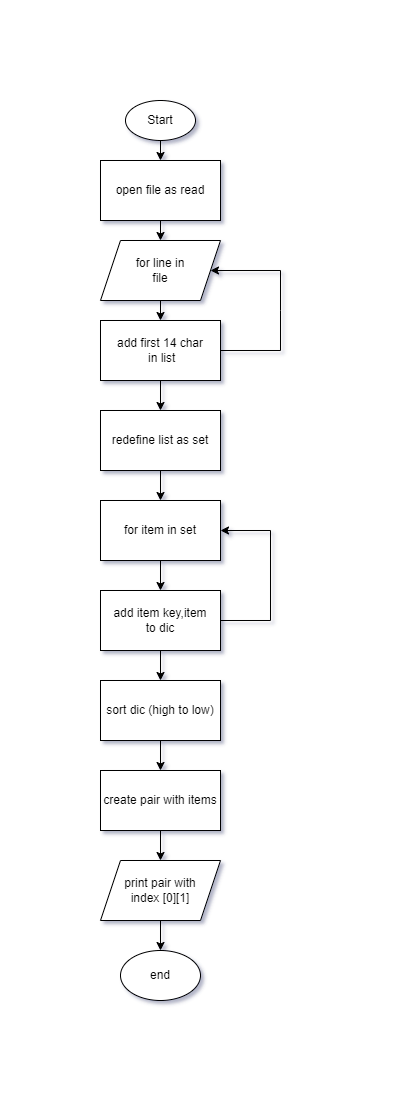
|  |
| --- |
|  |
| Report |
|  |

|  |
| --- |
| Patrick Acheson ec1948437  04-05-2022 |

# [Design](https://moodle.edinburghcollege.ac.uk/mod/assign/view.php?id=2035747) Stage

# Flowchart

I will be using flowchart diagram to represent a workflow of my python script. A flowchart can be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

# Evidence 1 & 2

**Code storing data in an appropriate structure**

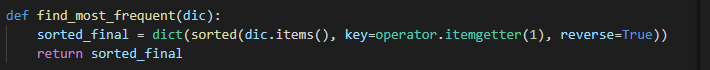
Using a list to store all the IP addresses found the in the log file each item being a string. A dictionary will be used to link the count of occurrence of the corresponding IP address, this will be my key, item pair.



I used a set function to remove all the duplicate addresses from the list of IP addresses and used this to build the dictionary.



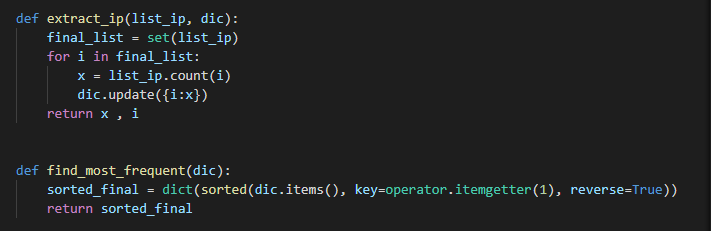
I also stored the sorted final dictionary as a dictionary, this was sorted from large to small based on the item value.



**Code retrieving highest occurrence of an IP address**

In this part I needed to find a way to count all the occurrences of each unique IP address. I did this but build a set of all the unique addresses, I then iterated through this and counted how many times (i) occurred in the original list of addresses, I then updated my dictionary with (i) being my key and (x) being my item, it did this for all the unique addresses list.

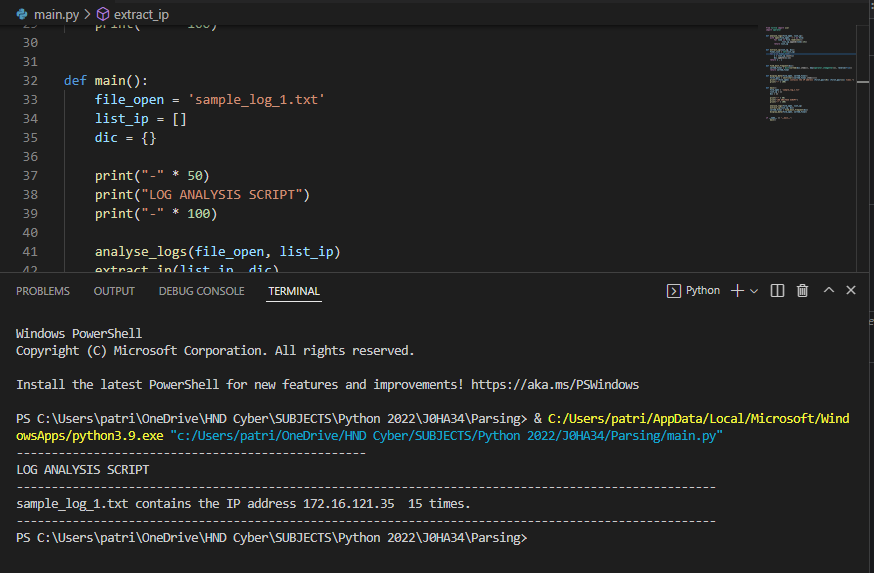
After this I had to sort this list by highest to lowest using the items(x) value. I had to import an operator function as well as the built-in reverse function. This worked although it was from small to large, so I reversed this by setting reverse to true.



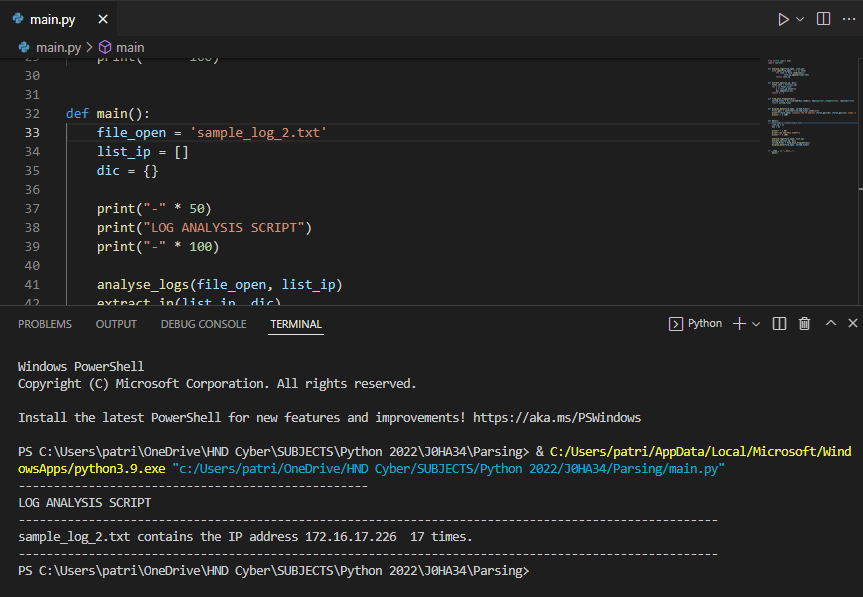
# Testing Evidence

**Code storing data in an appropriate structure**

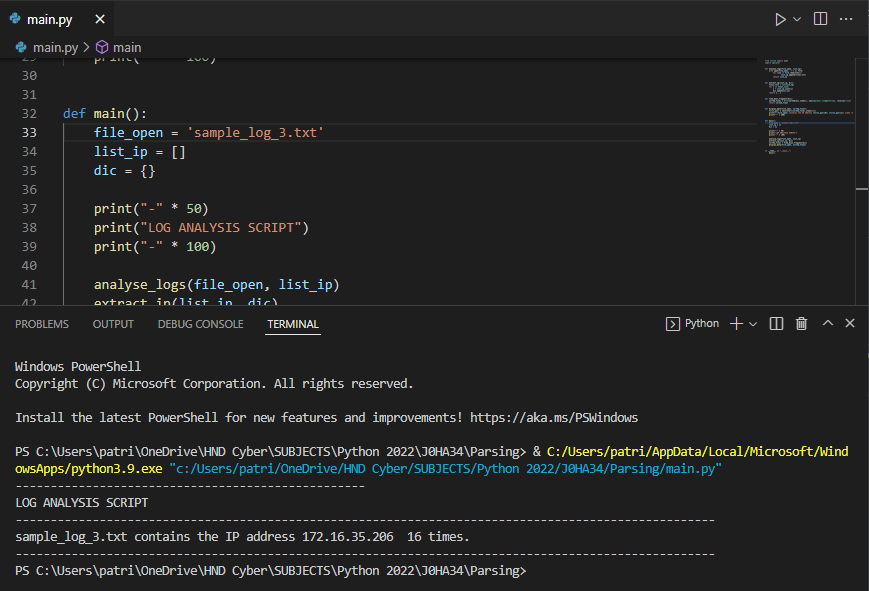
**Sample\_log\_1.txt**



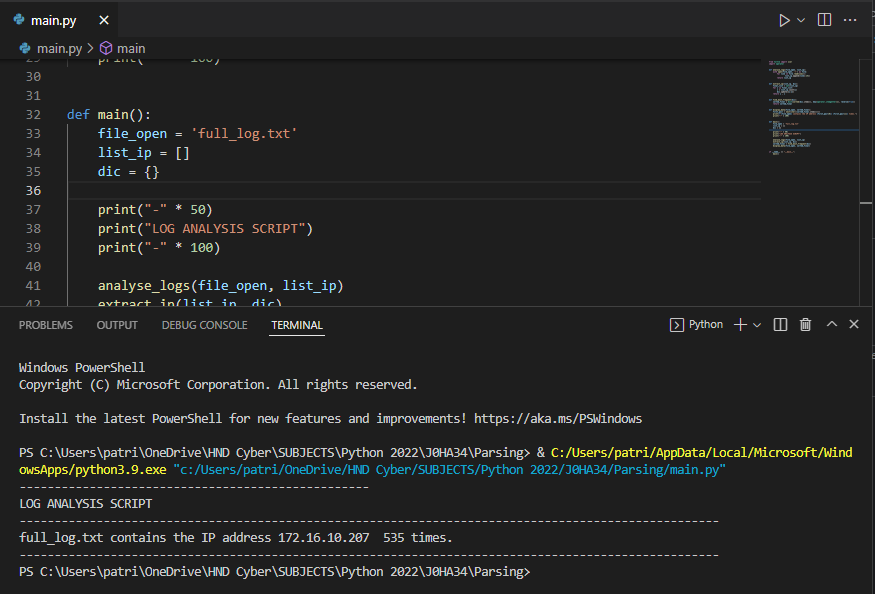
**Sample\_log\_2.txt**



**Sample\_log\_3.txt**

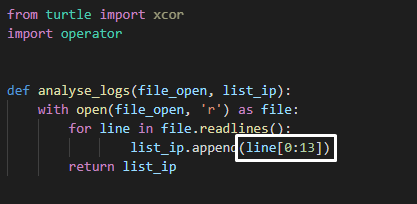


**Full\_log.txt**

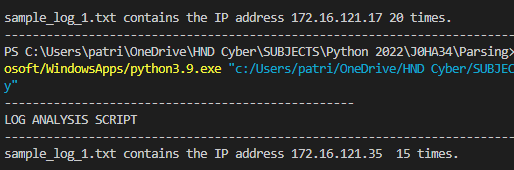


# Development Log

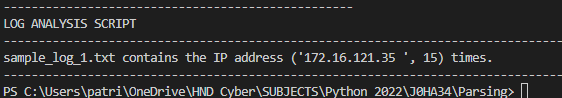
I started writing the script, while creating the first analyse\_logs function I made a mistake. In this part I was attempting to take the first 13 characters from the beginning of each line. The issue with this was it would cut off the last base 10 value from the end of the IP address. Instead of outputting the expect value it would give a higher count value.



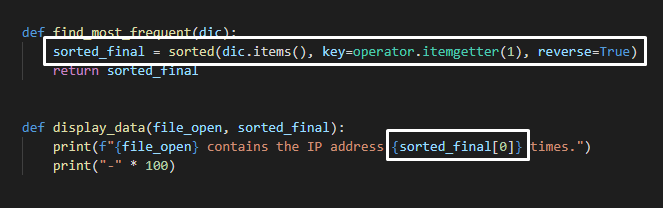
I fixed this issue by changing the value of 13 to 14 this meant it would take all the IP address and produce a correct count.

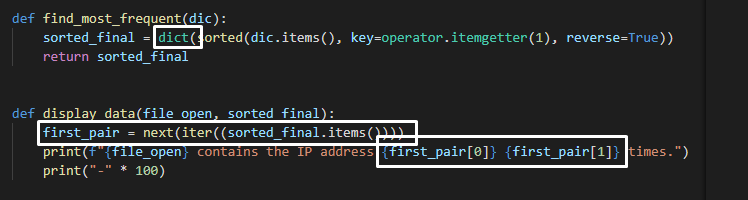


Down the line of the development, I had another issue. When I printed the IP and the occurrences it was print in a list type.



This issue was when I was sorting the dictionary, I was making it a list, this was not only inefficient but make the output loose its formatting.



I realised I could fix this but making it sort back into a dictionary and finding a way to display my key and items from the first value in this list (the highest count).

After I had made these changes and displayed the key and value by index, I had a formatted output and I had completed the development on this script.

