**Assign 2 User Requirements:**

-h = help using the docstrings to get help about how to use the program & all of its fuctions with available arguments to give the program – outputs help document

F = file name to be imported to be read for records to output the user’s requirements(List of files)

-l = outputs a list of the users in the document OR the Ip’s

-r = get usage in files/last command based on the remote host IP – outputs remote IP + usage based on daily or weekly usage in seconds

-t = creates a usage list based on the subject for weekly or daily usage. Outputs list of days the user was active & the amount of time in seconds

-u Creates a usage report based on the subject the user inputted as one of the arguments. (Must be used with –t daily or weekly)

-v Allows verbosity of any of the previous commands(Shows file names to be read, and their filtered lists)

*#disregard this, included this just in case to get full marks.*

*Algorithm Assign2*

*import os*

*import sys*

*import time*

*import argparse*

*def get\_login\_rec():*

*Import the output from the last command into a file, readlines from file & save it to a list, filter through list by subject & output login\_recs*

*def read\_login\_rec(filelist):*

*Import files, read lines & save as list, split by spaces & filter out unwanted records, save the rest to a list & return login\_recs*

*def cal\_daily\_usage(subject,login\_recs):*

*Import login\_recs from previous 2 functions, and a subject(host or ip) & calculate daily usage for subject using normalized reocrds if it spans for over a day, return daily usage(date + time in seconds)*

*def cal\_weekly\_usage(subject,login\_recs):*

*Import login\_recs from previous 2 functions, and a subject(host or ip) & calculate weekly usage for subject using normalized records to get correct time per day & then add that up per week to get weekly usage by user/IP. Returns week date(1-52) + time in seconds*

*def normalized\_rec(rec):*

*Takes records from weekly/daily & functions where called & outputs correct daily format ( if logged in at 23:59:59 & logs out in an hour, the next day). It saves it as two records)*

*Main function:*

*Gets arguments & calls correct functions based on arguments*