**Assignment 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 Assignment 2

import os

import sys

import time

import argparse

def get\_login\_rec():

get\_login\_rec function will read from the command if it has the word "last" then get the data from the system, using last command and return the valid data to an unformatted list

def read\_login\_rec(filelist):

read\_login\_rec accepts one argument which is the log file name that will be given, it reads the file and return an unformatted list.

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

cal\_daily\_usage accepts 2 arguments. First argument can be username or ip address. Second argument is the list of formatted records which get from function normalized\_rec. It collects all the records in the same day then sum all the total time. It will return a dictionary with the key is the date and the value is total time for each date, and the Total time of all the dates.

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

cal\_daily\_usage accepts 2 arguments. First argument can be username or ip address. Second argument is the list of formatted records which get from function normalized\_rec. It collects all the records in the same weekday of the year then sum all the total time. It will return a dictionary with the key is the date and the value is total time for each date, and the Total time of all the dates

def normalized\_rec(rec):

Takes records from weekly/daily & arguments 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