Team Members: John Hunter and Vamsee Gangaram
PA2Group 14

1. IMPLEMENTATION IDEAS
    a. We utilized multithreading in a client/server architecture to allow multiple clients to connect to one server and execute a number of tweet-related functions.
2. DIVISION OF LABOR
    a. Initially, John was in charge of the subscribe, unsubscribe, and timeline functionalities, and Vamsee was in charge of the remaining functions: tweet, getusers, gettweets, and exit. After the initial work was done for these functions, both members worked equally on whatever bugs needed fixing as they arose. Additionally, both members worked equally on the initial development of the multithreading architecture and overall program structure.
3. HOW TO USE THE CODE
    a. First, run the server program from the command line as such:
        i. `$ python3 ttweetser.py <PORT>`
    b. Then, run up to five separate clients from the command line as such:
        i. `$ python3 ttweetcli.py <SERVER_IP> <SERVER_PORT> <USERNAME>`
        ii. The available client commands are:
            1. **tweet** "<150 char max tweet>" <Hashtag>
            2. **subscribe** <Hashtag>
            3. **unsubscribe** <Hashtag>
            4. **timeline**
            5. **getusers**
            6. **gettweets** <Username>
            7. **exit**
    c. The only item needed to be installed is Python3
        i. The packages used in the files are socket, sys, shlex, threading, time, and os. However these packages should be preinstalled with Python3.