# Python for Informatics

# Assignment 5

# "Message Frequency Count"

## Description:

1. ***As you complete this assignment, please do not use single character variable names. Instead, use meaningful names (names that tell the reader what is represented and why).*** Using the ***DSU pattern***, write a program that reads through the mail box data and when you find a line that starts with "From", extract the address information from the line. Count the number of messages from each person by using a ***dictionary***. Note that you might need to look at more than "From" because of duplicate instances of the address (hint: "From " vs. "From:", not both! If you search for "From", you will find both "From " and "From:", which will erroneously double your count—please note that "From " is not the same as "From"). Otherwise, embedded email "thread histories" may cause your count to be incorrect. In other words, when counting the message "sends", you definitely don't want to count any embedded messages that are included as part of the message thread history. The idea is to count "original" sends. If you send me a message, and I respond, you respond to my response, etc., etc.,... how many times did you send that original message? Only once, right?
2. After all of the data has been read, print (i.e., *print*) a message to the user stating that the person with the highest number of sent messages is <email_address>, along with the number of sent messages associated with that email address (e.g., "with *n* messages"). In other words, as with any informatics output, your output should be meaningful—e.g., "The person with the highest number of sent messages is neo@matrix.com with 7700 messages." To obtain your result, create a list of tuples (count, email) from the dictionary, sort the list in reverse order and print out the person who has the highest number of messages. The ***U*** in ***DSU*** does not mean that you print a list, and it doesn´t mean that you print a tuple. Rather, it means that you ***Undecorate***, i.e. extract the necessary information from the tuple and present it to the user in a meaningful form (something your boss will understand).

Note: To succeed with this assignment, ***know your data!*** When your program counts the messages, how do you know if you are counting the messages correctly? Could you be counting the same message more than once? If your program were to operate correctly, how would you know it? Is there a smaller file that you can use to test your program? (Hint: There's a file named *mbox-short.txt*.) Be careful though. When you submit, are you submitting an example using the required *mbox.txt* file?

## Deliverable:

Two files as attachments at our course shell assignment page. The first file should be a Python .py file with the specified functionality. Please ensure that your full name is specified as a Python comment at the top of the ***.py*** file. The second file should be a screenshot image file (.png or .jpg) *demonstrating the correct execution of your program with "mbox.txt"*.

## Submission Deadline:

Please see the course schedule in our syllabus for all assignment submission deadlines.

## Peerwise Reminder:

If you haven't created any multiple choice Peerwise questions yet, that probably means that you don't want the course points associated with them.