INF 510 – Principles of Programming for Informatics

Homework 3

Spring 2017

Due Date: Feb. 03, 2017 at 4:30 PM in Class

Instructions:

- Send your work to the instructor at: pedro@isi.edu
- For all the programs in this assignment provide a separate file following the naming scheme: lastname_INF510_HW3_Qn.py for question n.
- You are encouraged to use the python interpreter to validate your answers.

Question 1. [20 points]
Using the sample 'mbox.txt' file available at the class web site develop a python procedure that takes as argument a file name and creates a file named 'email_list.txt' with all the 'sender' emails found in mbox.txt. This list of e-mails should contain no duplicates. In the end, your procedure should also print out to the output the number of e-mails found.

Question 2. [30 points]
Develop a python program that traverses a file tree hierarchy and collects in a file all the file names (including the path from the current working directory) of all the files whose name end in a given string given as parameter. In the end of the processing, your procedure should also print out to the output the number of files found.

Question 3. [30 points]
Write a python program that takes a comma-separated-value (csv) file and print the fields of the line of that file that has the highest numeric value for a column given as parameter to the program. If there are not enough columns in a given line of the program, or the value is not numeric, that specific line should be ignored. Develop a couple of sample files to validate your work.

Question 4. [20 points]
Write a program that reads the e-mail 'mbox.txt' file and finds the sender of the e-mail messages within. To accomplish this, your program should find the lines that start with the string "From", and split those lines using the split function to isolate the second word which is the sender's e-mail. For example, in the line '`From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008`' your program should identify the sender as '`stephen.marquard@uct.ac.za`' . Your program should then collect all the addresses that sent emails, and then print out the number of times such sender e-mails occurs. To make matter simple, the output of your program should look something like this:

```
gopal.ramasammycook@gmail.com : 1 time
louis@media.berkeley.edu : 3 times
cwen@iupui.edu : 5 times
```