Home Work #3

1. Prepare a 1-page cheatsheet on everything we have covered on Regular Expressions. It should be readable when printed on a standard letter-sized paper. Feel free to look at the myriad such cheatsheets on the Internet but the one you submit must be your own work. Organize it in your own way. On the second page, cite the cheatsheets you found useful.

2. Given a text file named data-emails.txt residing in the current directory containing emails (example supplied), you are asked to write a program in Python 3 importing (only) the re library to extract for each email message: (Notation: ␣ represents a space or blank character.)

   - SenderName: the sender's name (if present, it will be available in the line beginning with *From:*␣ preceding the email address);
   - SenderEmail: the sender's email address (available in the line beginning with *From:*␣) (See the examples.);
   - Date: the date (available in the line beginning with *Date:*␣);
   - Subject: the subject (available in the line beginning with *Subject:*␣);
   - Recipients: the recipients (available in the line beginning with *To:*␣). This line may be followed immediately by a sequence of lines containing additional recipients, all starting with a tab e.g., see file msg-multiple-recipients.txt); in this case, all the recipients need to be concatenated into one recipient list;
   - NumConf: the number of occurrences of any of the words *confidential, confidentially, confidentiality* in the body of the message; and
   - NumDollar: the number of occurrences of $ (the dollar symbol) in the body.

   Each message starts with a header separated from the body by one blank line. The body of the message can contain blank lines as well as lines starting with *From:*␣ / *To:*␣ / *Date:*␣ but it will never contain a line starting with *Return-Path:*␣. For simplicity, you may assume that in every header, the *Return-Path:*␣ line will precede the *From*␣, *To*␣, and *Date*␣ lines.

   It is possible for a message to have an empty body.

   Output a CSV file containing and complying with the following header line NumConf,NumDollar,SenderName,SenderEmail,Date,Subject,Recipients The entries should be sorted by the word counts NumConf (primary) and NumDollar.

   - It is necessary that you import and use the re library to extract the data; but you may embed the calls to those functions / methods within regular Python 3 code. For example, the logic for detecting the header versus body need not be a regular expression. The regexps must be **adequately commented**. Use named patterns. Multi-line doc strings are recommended.
   - As before, submit a .py file containing a boilierplate redirection to main( ).

Appendix
Not for submission.

Here's a quick recap of the builtin anchors ^ , $, \A, \Z, and their differences in the multiline mode.

- ^ (Caret) matches the start of the string,
  and in MULTILINE mode **also** matches immediately after each newline.
- \A only matches the start of the string.
- $ matches the end of the string or just before the newline at the end of the string,
  and in MULTILINE mode **also** matches before a newline.
- \Z matches only at the end of the string.

https:
//docs.python.org/3/library/re.html#regular-expression-syntax
Example:

```
import re

text = '''I scream.
You scream.
You know
the rest.
I don't.'''

print(text)
```

What are the following queries designed to match?

```
# 1
print(re.findall(r'\A.*\.$',text,re.M))

# 2
print(re.findall(r'^.*\.\Z',text,re.M))

# 3
print(re.findall(r'^.*\.$',text,re.M))

# 4
print(re.findall(r'^.*\.$',text))
```