ENGR 102 – Programming Practice
Practice Session (Week 8)

In this practice session, you will learn about regular expressions, and have a chance to practice what you have learnt on small problems. First, the session lead will cover the basics of regular expressions (the presentation is available on LMS under Week 8), and then you will test your knowledge on the following sample problem set. A separate tutorial on regular expressions is also made available for you on LMS as an additional resource for self-studying.

1. Find all numbers that include years, percentages, dollar amounts and other numeric figures in one regular expression.
2. Find all quoted statements (e.g., “statement”)
3. Count the number of paragraphs and number of words through regular expressions. Each word is only a set of alphabets (hint: see len(list) for counting)
4. Replace person’s name (i.e. Alan Spoon) with your name. You are not allowed to use “Alan Spoon” as a string to match. (hint: see re.sub() functionality to substitute string)

   e.g:
   ```
   RE_space = re.compile(‘\s’)
   sentence_without_space = RE_space.sub(‘’, sentence_with_space)
   ```
5. What strings are matched by the following regular expressions?

   (a) [a-zA-Z]+

   (b) [A-Z][a-z]+

   (c) \w+|[^\w\s]
6. Come up with regular expression to match the following date format:

   17/05/2009 8:15

   Test your regular expression using the re.match function on the string above.
7. Introduce groups in the regular expression from the script in the previous exercise, to extract the hour of the day, and the minutes. Use these values to calculate how many seconds have passed since midnight, and print out the answer.
8. Use the re.split function with an appropriately formed regular expression to extract the numbers from the following line of data:

   45, 3453 : 19, -1.e-10

   Your script should be able to handle the case that any of the numbers are in exponential form (such as the last number shown above).
9. Imagine you have a script that names variables with a leading underscore, like this: _someVar. You decide you want to remove the leading underscore, and use a trailing underscore instead, like this: someVar_. Write a short script that uses the re.sub function to achieve this transformation. Come up with a small amount of trial data, and test your script on it.