CSCI 1133, Spring 2015
Programming Assignment 5

Assigned: Friday February 27, 2015
Due: 11:55pm Thursday March 5, 2015

This is ***not*** a collaborative assignment; you must design, implement and test the solution(s) on your own. You may not consult or work with anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class. Obtaining or *providing* solutions to any homework problems for this class is considered academic misconduct. If you are not sure what this means, consult the class syllabus or discuss it with the course instructor.

This assignment requires writing a single Python script that must be submitted online *prior* to the due date/time. Late submissions will not be accepted. Name your source code: `hw5.py` Submit your source code file using the appropriate homework submission link on the Moodle website.

The total point value for programming assignments will be awarded for solutions that are *complete*, *correct*, and *well constructed*. A *"well constructed"* program entails good design, appropriate comments and general readability (descriptive names for variables and procedures, appropriate use of blank space, etc.). The following will result in a score reduction equal to a percentage of the total possible points:

- Incorrectly named/submitted source file (10%)
- Constraints not followed (40%)
- Failure to execute due to syntax errors (30%)

Note that your work will be graded using, and must function correctly with, the current version of Python 3 on CSE Labs UNIX machines. If you complete your programming assignment using a different system, it is your responsibility to ensure your programs work on CSELabs machines *prior* to submitting them.

A. (40 *points*) **Roman Numerals**

Write a Python program that will read Roman Numerals and convert them to decimal values. This assignment involves constructing a single source file, however you should approach it in three distinct steps as follows:

Part 1: Write a function definition for a function that will take a single Roman Numeral digit (type string) and return its integer value according to the following:

I = 1
V = 5
X = 10
L = 50
C = 100
D = 500
M = 1000

If the input value is not a valid roman 'digit', return 0 (zero). Be sure to thoroughly test your function.

Part 2: Write a function definition for a second function that will take a single string argument containing a Roman Numeral (e.g., 'IX', 'VIII', etc.) and return its equivalent decimal value (integer). To convert a Roman Numeral to a decimal value, you need to consider the "rules" of Roman Numeral representation:

1. Reading left-to-right, higher valued symbols generally appear before lower valued symbols. The value is obtained by summing the individual symbol values (obtained by calling the function created in part 1)
2. However, if a *single* lower valued symbol *immediately* precedes a higher value, it is subtracted from the total rather than added.
3. The symbols I, X, C, and M cannot be repeated more than 3 times in succession.

4. ‘I’ can only be subtracted from ‘V’ and ‘X’. ‘X’ can only be subtracted from ‘L’ and ‘C’. ‘C’ can only be subtracted from ‘D’ and ‘M’. ‘V’, ‘L’, and ‘D’ can never be subtracted.
5. Only a single lower valued symbol may be subtracted from a higher valued symbol.

This function must include a call to the function you created in Part1 to convert the individual Roman 'digits' to their equivalent decimal value.

Hint: Your function should sum up the symbol values from left-to-right, but *defer* adding/subtracting each value to the total until the subsequent symbol is determined to be larger/smaller.

Part 3: Write a short Python program that will input Roman Numerals, call the function you created in Part 2 and then output the equivalent decimal values. Use an indefinite, sentinel-controlled loop that will continue converting numbers until a null (empty) string is entered.

Constraints:

- Use only simple indexing and/or iteration on the input string. Do not use any string methods or alter the input string in any way
- If lists are employed, use only simple indexing and/or iteration operations, do not use any list methods.
- You must create and use the function definitions as described

Examples:

```
Enter a Roman Numeral: X
Decimal value: 10
Enter a Roman Numeral: XIX
Decimal value: 19
Enter a Roman Numeral: MCCL
Decimal value: 1250
Enter a Roman Numeral: XL
Decimal value: 40
Enter a Roman Numeral: MCDLVIII
Decimal value: 1458
Enter a Roman Numeral:
>>>
```