

Programming Project #4

This assignment is worth 40 points (4.0% of the course grade) and must be completed and turned in **before 11:59 on Monday, October 10, 2011.**

Assignment Overview

In this assignment you will practice with:

1. strings
2. if statement
3. loops (for, while)



Background

In 1937 Claude Shannon of Petoskey, MI wrote the seminal work for computing that showed how Boolean logic could be expressed in electronic circuits leading to computers as we know them. For fun he created Throbac (THrifty ROman-numerical BACKward-looking Computer), a calculator that did arithmetic with Roman numerals (pictured above)—a useless, but whimsical device. In this project, you will write a program that performs addition with Roman numerals, just like Throbac.

Project Specification

Roman numerals have values, but are not positional like our number system, and there is no zero (zero arrived in the west circa 1000 AD). To keep things simpler, we will use five of the seven symbols: I = 1, V = 5, X = 10, L = 50, and C = 100. See http://en.wikipedia.org/wiki/Roman_numerals. Numbers are formed by combining symbols together and adding their values—only whole numbers. For example, CLXVII is 100+50+10+5+1+1 = 167. Generally, symbols are placed in order of value, starting with the largest values. When smaller values precede larger values, the smaller values are subtracted from the larger values, and the result is added to the total. For example, in XLIV the smaller X before L means subtract X from L as 50-10 to get 40, and the I before V means subtract I from V as 5-1 to get 4, so the final number is 44. Some call that the “subtraction rule.” A useful restriction of the subtraction rule is that there can never be more than one smaller value. That is, XL is valid, but XXL is not: thirty is represented by XXX.

The table below shows the base values up to three hundred—to keep things simpler we will assume that our values will stay less than 380 (but we will not check).

	1	2	3	4	5	6	7	8	9
Ones	I	II	III	IV	V	VI	VII	VIII	IX
Tens	X	XX	XXX	XL	L	LX	LXX	LXXX	XC
Hundreds	C	CC	CCC						

Your program will:

1. Prompt for two Roman Numerals.
2. Convert both numbers to integers.
3. Find the sum of the two integers.
4. Print the sum as a Roman Numeral.

Deliverables

You must use Handin to turn in the file `proj04.py` – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file name, and save a copy of your `proj04.py` file to your H: drive as a backup.

Notes and Hints:

You should start with this program, as with all programs, by breaking the program down into parts. Here is some of that breakdown to help you

1. First, play with Roman numerals with pencil and paper to get familiar with them and their rules, especially conversion to and from integers. Notice how the “subtraction rule” requires you to look at pairs of symbols. The restriction that only one smaller value can precede a larger value will be quite helpful in developing your algorithm.
2. Simplify! Inside of every problem is a simpler one.
 - a. The most obvious simplification is to handle the four steps individually—I did steps 2 and 4 as separate programs while developing my solution.
 - b. There is another simplification. Can you see it? (Check this footnote¹ for it.)
3. I used a large number of *if-elif-else* statements. For example, in my solution I counted 19 occurrences of *if* and 21 occurrences of *elif*. Depending on your reasoning you may have more or fewer, but I mention the numbers so you get a sense that there will be many.
4. Your final result should be in proper form, e.g. IV instead of IIII.

Example Output:

```
Enter First Roman Number (no spaces): XLIV
Value of XLIV : 44
Enter Second Roman Number (no spaces): CCLXV
Value of CCLXV : 265
Digital sum is: 309
Roman sum is: C C C I X
```

¹ Solve the problem for Roman Numerals that don't use the subtraction rule, e.g. assume input is something like LXXII and not numbers like XLIV. After you get that working, introduce the subtraction rule.