COSC 251 – Programming Languages
Project 2
Spring 2010

Objective: Use Python to solve a bevy of problems.

Your Task: The SMCM Programming Competition had a wide variety of problems all of which could be solved via Python without too many issues. You've seen (or will see) a couple of them as part of a Python lab, but those were part of the "below 201" subset of problems. For this project, you will provide solutions to 4 of the problems in the more difficult "above 201" subset of problems. You will be required to answer all four questions and each question is worth 25 points.

Q1: An anagram is a word or phrase formed by rearranging the letters of another word or phrase. For example, carthorse is an anagram of orchestra. Blanks and other whitespace are ignored for the purpose of creating anagrams. For this problem, the user should be able to input a pair of phrases, one at a time, and the program should output a message regarding depending on whether the two phrases are anagrams or not.

Sample Run:
```
Enter the first phrase: carthorse
Enter the second phrase: horse cart
Yes, these phrases are anagrams.

Enter the first phrase: horse car
Enter the second phrase: orchestra
No, these are not anagrams.
```

Q2: Create a program that will take in a Roman numeral and output the correct decimal integer for that Roman numeral. The following table indicates the letters and values you will need to handle:

I = 1	V = 5	X = 10	L = 50	C = 100
D = 500	M = 1000	v = 5000	x = 10000	l = 50000
c = 100000	d = 500000	m = 1000000		

Note that uppercase and lowercase does matter. The user will continue to input numbers until the user inputs the word 'quit'.

Sample Run:
```
Enter a number to convert: XVII
XVII = 17
Enter a number to convert: IX
IX = 9
Enter a number to convert: xXI
xXI = 10011
```

Q3: You are lost in a city that is organized as a rectangular grid of 10201 total locations. You have a device that you can input coordinates and a relative direction and it will tell you how many possible locations for your position. The input is formatted as X Y D, where X is an integer coordinate on the X axis (0 to 100, inclusive, west to east), Y is an integer coordinate on the Y axis (0 to 100, inclusive, south to north) and D is a cardinal direction as a single letter (N, S, E, W). For instance, if you enter the input 50 30 E, then this means that your x coordinate is 51 or higher.

Your program should ask for the number of inputs, get the inputs, then output the number of possible locations.

Sample Run:

```
How many inputs? 3
Input 1: 30 40 S
Input 2: 60 15 E
Input 3: 10 20 N
The number of possible locations is 760

How many inputs? 1
Input 1: 100 99 N
The number of possible locations is 101

How many inputs? 2
Input 1: 50 60 N
Input 2: 70 40 S
The number of possible locations is 0
```

Q4: Your final program will calculate the possible blood types for a child or a parent, based on the input given. Every person's blood has two ABO alleles, each being one of three possible letters: A, B, or O. Every person also has two alleles for the Rh factor, each being the characters + or -. Depending on the combinations of both sets of alleles, a child can have any number of different blood types:

Allele Combination	ABO Blood Type	Allele Combination	Rh Factor
AA	A	++	+
AO	A	+-	+
BB	B	--	-
BO	B		
AB	AB		
OO	O		

Your program will take in an input of three sets of blood type values, two for the parents and one for the child, where one of the sets will be a question mark. If the question mark appears for the child blood type, the program will output the possible blood types for the child based on the parents' blood types. If the question mark appears for one of the parents, then the program will determine what possible blood types the parent has depending on the blood type of the other parent and the blood type of the child. Keep in mind that this may not be possible. The input will be formatted as Parent Parent Child.

Sample Run:

```
Enter blood type input: O+ O- ?
Child's possible blood types: O+, O-

Enter blood type input: O+ ? O-
Parent's possible blood types: A+, A-, B+, B-, O+, O-

Enter blood type input: AB+ ? O+
Parent's possible blood types: IMPOSSIBLE
```

Deliverables: your Python sources, including test programs for each of the four problems.

Expectations: The code should be clean, concise, well-commented and correct. If you use an outside source, be sure to document that source. Significant use of outside sources will result in a deduction. Grading rubric will be provided a week ahead of the due date. You are allowed to work in pairs for this project.

DUE: March 26th, 5:00pm via Digital Dropbox