

## Project 2

Due January 30, 2015 at 11:55 PM

You will be working alone for this project. You should avoid using existing source code as a primer that is currently available on the Internet. You **must** specify in your comments of the file any sources of code that you have viewed to help you complete this project. All class projects will be submitted to MOSS to determine if students have excessively collaborated. Excessive collaboration, or failure to list external code sources will result in the matter being transferred to Student Judicial Affairs.

1. You will be writing a Python program solve the quadratic equation. Name your file **quadratic.py**.
  - a. Place a comment at the top of your file as follows:
 

```
#
# Name: Your Name
# ID   : Your Student ID
# Date: Today's Date
#
```
  - b. Prompt the user to input the coefficients of the quadratic equation on one line. Your prompt needs to be as follows:
 

```
Input A B C >
```
  - c. Validate the user input. Display the errors as specified below. The user must input three floating point numbers (do not worry if the user inputs non floats and your program crashes). The user must not enter zero for A. Prompt the user until the user provides valid input. Below are two invalid input examples, the user input is in bold:
 

```
Input A B C > 1 3
Input error "1 3" is not a valid A B C
Input A B C > 0 2 1
'A' cannot be zero!
```
  - d. Calculate the solutions for the quadratic equation. This must be able to determine if there is a single solution, two solutions, complex solutions, or purely imaginary solutions.
  - e. Print the solution(s) in a pretty format. The display should print differently if there is only a single solution vs. two solutions. The formatting of the quadratic equation must omit 1 and -1 when they are used for coefficients A and B. The formatting of the quadratic equation must omit the Bx and C when the coefficient values for B or C are zero. If the imaginary part is 1 then the value should be omitted as with 1 for coefficients. Below are examples of output given various inputs.

```
Input A B C > 1 2 1
Solution for x^2 + 2x + 1 is: -1
```

```
Input A B C > 1 5 -6
Solutions for x^2 + 5x - 6 are: 1 and -6
```

Input A B C > **1 2 2**

Solutions for  $x^2 + 2x + 2$  are:  $-1 + i$  and  $-1 - i$

Input A B C > **1 0 1**

Solutions for  $x^2 + 1$  are:  $i$  and  $-i$

Input A B C > **1 1 1**

Solutions for  $x^2 + x + 1$  are:  $-0.5 + 0.866025i$  and  $-0.5 - 0.866025i$

Input A B C > **4 2 0**

Solutions for  $4x^2 + 2x$  are:  $0$  and  $-0.5$

Input A B C > **2 0 0**

Solution for  $2x^2$  is:  $0$

2. You will be writing a Python program to determine if two phrases are anagrams of each other or not. Name your file **anagram.py**.
  - a. Place a comment at the top of your file as follows:
 

```
#
# Name: Your Name
# ID   : Your Student ID
# Date: Today's Date
#
```
  - b. Prompt the user to input the original string and an anagram string. You should keep prompting the user until they enter in a string that has at least one non-space character in it. Your prompt needs to be as follows:
 

```
Enter original string>
Enter anagram string >
```
  - c. Determine if the anagram string truly is an anagram of the original string. You should ignore spaces and punctuation, and must do your comparison of letters as case insensitive. Determine how many of which letters are missing (if any) and how many of which letters extra (if any) in the anagram string.
  - d. Print the result of your calculation. If the two are anagrams of one another. Show that they are, if they are not, specify they are not and which letters are missing/extra. The missing/extra letters must be listed in alphabetical order. Below is an example output of several runs, the user input is in bold:

Enter original string> **Anagram**

Enter anagram string > **Nag a Ram**

"Nag a Ram" is an anagram of "Anagram"!

```
Enter original string> Hello World
Enter anagram string > Goodbye
"Goodbye" is not an anagram of "Hello World"!
"Goodbye" is missing:
1 H
3 L's
1 R
1 W
"Goodbye" has additional:
1 B
1 G
1 Y
```

```
Enter original string> Dog
Enter anagram string > Goodbye
"Goodbye" is not an anagram of "Dog"!
"Goodbye" has additional:
1 B
1 E
1 O
1 Y
```

```
Enter original string> Good
Enter anagram string > Dog
"Dog" is not an anagram of "Good"!
"Dog" is missing:
1 O
```