CSCI-15 Assignment #1 — Functions and files review (40 points), due 9/6/18.

Write a program, broken into functions as specified below, to read the coefficients of a series of quadratic equations from a text file and print the associated roots (or appropriate error messages if there are no real roots) to another text file. The coefficients of a quadratic are the a, b and c of an expression of the form $ax^2 + bx + c$, and the roots are the values of x that make the value of the expression 0. The roots are given by this formula:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If a == 0, the formula describes a line, and you may ignore any roots (just say it isn't a quadratic, don't try to calculate the root), and if (b^2-4ac) < 0 it has only complex roots.

Write a function to read **one** set of values from the file, using reference parameters to get the values out of the function and the function's return value to indicate whether or not the function was able to correctly read three values. Return the 'stop' code on end-of-file or a data error. The only data error you must deal with here is too few values on the line, e.g., the line has a and b values only and no c value. You may assume that the last line in the file is the only one with an error (if any error exists), and your function should return an error code to make the program stop processing after this error. This restriction allows you to use stream extraction to read the file, rather than reading lines and parsing them yourself. (If you really want to try using getline() and parsing the line yourself, you may, but get it working the easy way first.) This function MAY NOT try to read all the lines in the file, just read one line per call to the function.

Write a second function to calculate the roots, taking the coefficients through value parameters and giving back the roots (if they exist) via reference parameters. Use the return value to indicate success at calculating roots (0), no solution (-1) or complex roots (-2). Do not call this function if you have a data error on input.

Write a third function to print a reasonably formatted table (one row per call from the main loop) of the coefficients, and either the roots or (different) messages indicating the various error conditions. It must print appropriate error messages in the cases of a data error on input and either no solution or complex roots from the calculation function. **Your table must have a reasonable title line (or lines) with legends describing what things are below it in the columns, and this table's title line must be printed inside the print function. The function must know if it is being called for the first time (or not) to print the title line. You MAY NOT pass this information into the print function from main().** The print function MUST do this by itself. Always print the coefficients unless you have a read error. You must align the values in columns in a reasonable way (if you can't align the decimal points in the columns, that's OK). Don't worry about page breaks and new title lines if the output runs beyond one page.

Your main() function will prompt for the file names, open the input and output files, check for file open errors appropriately, print the input and output file names to the output file (but NOT the title line for the table! This MUST be printed in the third function!), then loop over the input file reading coefficients, calculating roots (if possible), and printing results to the output file until either end-of-file or error on input, and then close all the files and exit.

You may make no assumptions about how many coefficients are in the input file (e.g., you may not hold the values in arrays and process them after reading them all).

You may not use any global variables in this program. Your variables must be declared appropriately within the functions where needed; and passed to other functions as either reference or value parameters as appropriate. Your functions will indicate any problem they encounter by returning a value to main(), where the error must be handled appropriately. Your functions outside main() may do only the task assigned to them, and must do all of that task. For example, you may not check for `a` == 0 within main and only call the calculation function if `a` is not zero. Think carefully about what data you need inside each function, and what must be passed around between functions.

Each correct input line will comprise three real values of the form
[optional sign][digits][decimal point][digits], or
[optional sign][digits] if an integer.
The last input line might have fewer values. For example, your data file might look like this:

```
1 1 1
1.2 -2.3 0.4
-2 -3 -4
+0 -2 8.85
2.345                    (error — only one data point)
```

These data are available as the file quadratic1.txt on my Web page.

Test your program with several files containing numerous lines of data, including a few with data errors and several data lines for quadratics having no real roots. Use related names for the input and output files, and attach the files with the program source to your e-mail to turn in. You don't need to capture the DOS screens for each run to turn in, just make sure the data and output file names clearly show the file names and the relationship between the files.

Remember; do not send me object or executable files (or project files)! Also, **I do not grade programs submitted without data and output files – they get a zero score.** Also, programs whose output does not match what is produced by the program itself get a zero score. Thus, if you make a change to the program, re-run your tests! Take the time to get this correct BEFORE you submit it: I grade the first assignment I get and ignore all others.