

*Your submission for this assignment must be commented and must include both your name and your student number as a comment at the top of every source file you submit. Each of your submitted files must use a file name beginning 'comp3007\_w23\_#####\_assignment\_03' (replacing the number signs with your own student number) and any submissions that crash (i.e., terminate with an error) on execution will automatically receive a mark of 0.*

Officially, the Due Date for this Assignment is:

**Friday, February 10<sup>th</sup>, 2021, at 11:59pm EST.**

Late Submissions are **Accepted Without Penalty Until Sunday, February 12<sup>th</sup>, by 11:59pm EST.**

Submissions received after that will not be accepted and will receive a mark of 0.

The objective of this assignment is to allow you to practice with guards and tracing in Haskell by designing and implementing a simple program for determining the real-valued roots of a second-order polynomial using the quadratic formula.

A second-order polynomial is a polynomial of the form...

$$ax^2 + bx + c$$

...and the quadratic formula for determining real-valued roots is...

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

If you do not recall how to use this formula, you are directed to first take a few minutes and read the article at <https://mathworld.wolfram.com/QuadraticEquation.html>. Essentially, the use of this formula will result in either 0, 1, or 2 real-valued roots, depending upon whether the "discriminant" (i.e., the  $b^2 - 4ac$  subexpression under the radical) is negative, zero, or positive (respectively).

For this assignment:

- you must write a function that takes three Double arguments and produces one [Double] (i.e., a list of Doubles) return. If the three arguments correspond to the coefficients of a second-order polynomial (i.e., a, b, and c, respectively), then the return value list should contain all the real-valued roots of that polynomial. A valid return value would thus be either an empty list (i.e., []), a list containing one element (e.g., [1.23]), or a list containing two elements (e.g., [1.23, 4.56]).
- your solutions must use guards and is not permitted to use the if-then-else syntax.
- your solution must (in every use case) use the "trace" function to provide an explanation of how your answer was determined. To clarify, if the coefficients you passed as arguments resulted in a negative discriminant, then that is your explanation for why the return value would be the empty list [] and that explanation must be displayed. Please recall that the trace function must be imported from Debug.Trace and takes two arguments (a String and something else) before displaying the first argument string and then returning the second argument.
- you may safely assume that your function will always receive three arguments of type Double, **and that the first argument (i.e., co-efficient 'a') will always have a non-zero value.**
- you must test all branches of your function thoroughly, using test cases you have designed yourself and have included as comments at the top of the program.