Programming Assignment 2: Quadratic Equation Solver

D1271450 Eileen 李宇恩

The process of writing the code for the Quadratic Equation Solver

Firstly, I analyze the assignment, and think about how to write the code in the neatest way. I came down to the conclusion that to print the first two lines then dismantle the last line into many parts, using if else to determine what should be printed out.

Secondly, I printed out the first two lines, and wrote a code to print error when equals zero as commanded in the assignment, then, to determine whether it is the multiple real root of equation, the real roots of equation or the complex roots of equation through the equation b to the power of two minus four times c, then is to print the formula using pretty printing which would be explained below.

Lastly is to calculate the answer to the equation which differs between having the multiple real root of equation the real roots of equation or the complex roots of equation, by (-b+sqrt(pow(b,2)-4*a*c))/(2*a), (-b-sqrt(pow(b,2)-4*a*c))/(2*a), sqrt((-1)*(pow(b,2)-4*a*c))/(2*a), -b/(2*a) these equations.

The problems I faced when writing the code, and the solutions

It is hard to come up with a solution that would make printing more efficient, without having to consider every kind of possibilities, which would be more complicated, after thinking and looking at the assignment, I finally figured out I can just dismantle the third line, and if you don't change to next line, you can just keep on printing that line.

Another problem that I faced is that after finishing the code, when facing 0.0000 when the real roots of equation, it would print out -0.0000, so I had to add an if else in the code to make sure the 0.0000 is printed out as expected.

A trouble I had when running the numbers in the assignment examples is when inputting 1 5 7 the results printed out is -2.0000, not -2.5000 as in the example, after thinking for a while, I found my mistake of setting a, b and c as integer, as soon as I change them into float, the result has been accurate.

Pretty printing

Using if else determining a, b and c's scope, if a equals one, print X**2, don't print 1X**2, if a does not equal one, but equals minus one, print -X**2 instead of printing -1X**2, and if not all above including a does not equal one and minus one, but is bigger than one or smaller than minus one, print %.0fX**2, plugging a into %.0f.

If b is bigger than one, print +%.0fX, plugging b into %.0f, if b isn't bigger than

one, but is smaller than minus one, print %.0fX, plugging b into %.0f, and if b isn't bigger than one and not smaller than minus one, but equals one, print +X, if b isn't bigger than one and not smaller than minus one and doesn't equal one, but equals minus one, print -X, if not all above including b isn't bigger than one and not smaller than minus one and doesn't equal one and minus one, but equals zero, print nothing.

If c is bigger than zero, print +%.0f, plugging c into %.0f, if c is not bigger than zero, but is smaller than zero, print -%.0f, plugging c into %.0f and if not all above including c is not bigger than zero and is not smaller than zero, but it equals zero, print nothing.

Pretty printing is to let the formula printed on the third line be neater and easier to understand without any unnecessary numbers or symbols.