Objective: Your goal is to get practice writing a class to represent quadratic equations. A quadratic equation has the form $ax^2 + bx + c$. You will write a class that computes the discriminant, finds the roots, and has the ability to format the equation as a string.

Tasks:

- Write source code in a module called **quadratic.py**.
- Define a class **QuadraticEquation**.
- Write the constructor for the class. It should take in values *a*, *b*, and *c* as arguments. You should convert each one into a float inside the constructor before assigning it to the *private* member variable with the same name. You may assume that the user will always enter arguments that are of the correct datatype. For instance, the argument *c* will never have the value 'hello'.
- You must ensure the constructor has a means of handling argument *a* being 0, which means you actually have a linear equation, not a quadratic one. If that happens, raise a **ValueError** with the message, "`Coefficient 'a' cannot be 0 in a quadratic equation.`"
- Write **properties** for *a*, *b*, and *c*. You do not need to write setters.
- Write a method called **discriminant** which computes $b^2 - 4ac$.
- Write two methods called **root1** and **root2** which compute the roots of the quadratic equation. If the discriminant is < 0, the methods should return **None**. (You may call the **discriminant** method from inside another method.) Otherwise, the roots are computed as follows:
  - Root 1 = $\dfrac{-b + \sqrt{b^2 - 4ac}}{2a}$

  - Root 2 = $\dfrac{-b - \sqrt{b^2 - 4ac}}{2a}$

- Write the **__str__** method to return a string representation of the quadratic equation. This may be the hardest part of the lab, though it's really not that bad. Quadratic equations should be displayed as `ax^2 + bx + c = 0`. There are some corner cases, though, which are as follows:
  - If *a* is negative, put a minus sign in front.
  - If *b* or *c* is negative, the plus sign should be converted into a minus sign.
  - If *b* is 0, do not display that term.
  - If *c* is 0, do not display that term.
  - If *a* or *b* is 1, do not display it.
  - Examples:
    - (a = -1, b = 1, c = 1):   `-x^2 + x + 1.0 = 0`
    - (a = 2, b = -3, c = -1):   `2.0x^2 - 3.0x - 1.0 = 0`
    - (a = 1, b = 0, c = 25):   `x^2 - 25.0 = 0`
    - (a = 1.2, b = 2.3, c = 5.6):   `1.2x^2 + 2.3x + 5.6 = 0`
- Use the supplied PyUnit test script to check your work.