Lab 4

Due: Tuesday, October 27

1 Vectors

Repeat the "Grades" problem (homework 1) using a vector to store the grades. All the basic requirements are the same:

- User enters grades continuously (until they enter 999)
- Grades must be in the range [0,100]
- Print both the average *and the standard deviation* (Refer to the instructions in homework 1 for the standard deviation calculation)

2 Unit Testing

Turn your quadratic formula code from lab 1, part 1 into a function that accepts the three quadratic coefficients a, b, and c (all doubles). Do not prompt the user for these values, just write a function that accepts them as inputs. There should be no "cin" statements anywhere in your program.

Note that in general a quadratic equation will have two solutions. In order to "return" these two solutions from our function, we pass two additional `double` variables by reference and set their values within the function. To handle cases with no solution, define a global variable: `const double NO_SOLUTION = -9999;`

Your function should handle the cases described in the lab.

- If the solution is complex, $x_1 = x_2 =$ `NO_SOLUTION`
- If $a \neq 0$, x_1 and x_2 are the two solutions of the quadratic formula
- If $a = 0$, $x_1 = -\frac{c}{b}$, $x_2 =$ `NO_SOLUTION`

Now write a unit test to make sure your function works as expected. Make sure you test it for all the different possibilities:

- At least one test where $a = 0$
- At least one test where $\sqrt{b^2 - 4ac} < 0$
- At least one test where $\sqrt{b^2 - 4ac} > 0$
- At least one test where $\sqrt{b^2 - 4ac} = 0$

Run the unit test function from `main`

Useful information: Equality comparison with doubles

The following information will be helpful when completing part 2 of the lab.

Keep in mind that exact equality comparison with doubles is difficult, due to the limited precision of a computer representation of a double. Therefore, the better way to determine if two doubles are equal is to test whether their absolute difference is sufficiently small:

$$|x_1 - x_2| \leq \epsilon$$

Where ϵ is small compared to both x_1 and x_2.

You can implement this in your code in any way you like. One way is to write a new function: `bool are_equal(double num1, double num2)` which performs the above calculation (you can declare a global variable `const double EPSILON = 1e-5`, or some other very small number, to use for the equality test) and then use this function for your unit tests.