CPSC 1160 SECTION 001 INSTRUCTOR: GLADYS MONAGAN

ASSIGNMENT #2: FUNCTIONS AND PARAMETER PASSING

DUE DATE WITH BRIGHTSPACE: WEDNESDAY, JANUARY 18, 2023 AT 11:50 PM

THE LEARNING OUTCOMES

- to write a complete C++ program and to learn to compile, link, and run the program
- to call library functions (from the math library)
- to test user input and report errors
- to use (and thus understand?) parameter passing, specifically, to use pass-by-reference
- to modularize by writing (short) functions
- to unit test with the lightweight unit testing framework doctest

READINGS

Read chapters 1-6 of the textbook (for now you can skip §2.8.2, §3.9, §4.4, §4.6, §4.7, §5.8, §5.9, §6.14).

PROGRAM AT HAND

The standard form of a univariate polynomial of second degree is

$$f(x) = a\,x^2 + b\,x + c$$

The coefficient *a* must not be 0 but *b* and *c* could be 0.
We can solve f(x) = 0 by finding the zeros of the quadratic equation

$$a\,x^2 + b\,x + c = 0$$

where *a*, *b* and *c* are real constants. In other words, we want to find the roots or the values of *x* that will make the left side of the equation zero. Note that the roots may be equal to each other and they may be real numbers or complex numbers.

For example, the quadratic equation with *a=1*, *b=-5*, and *c=6* $x^2 - 5\,x + 6 = 0$ has two real roots at *x=2 and x = 3*.

The rules for finding the real roots of a quadratic equation:

1. If $b^2 < 4ac$, then there are no real roots.
2. If $b^2 = 4ac$, then there is one real root *x = -b / 2a* .
3. If $b^2 > 4ac$, then there are two real roots:

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

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

IMPLEMENTATION DETAILS

In the file solver.cpp

- Write a function to read the values *a*, *b*, and *c* from standard input (with cin)
- Write a function to output the quadratic equation to standard output (with cout):
  For instance, if the coefficient is 0, do not print 0 in 0*x^3 ;
  however, make sure that the zero polynomial is printed.
  You could write a separate function to output the coefficients but you don't have to.
- Write a function to display the results to standard output (with cout)
- In the main function, call the various functions and ask the user whether to continue by reading new a, b, and c and then solving the quadratic equation or whether to stop.

In the file equation.cpp

- Write a function getRoots to return the number of real roots in the quadratic equation and the (possible) calculated roots. See the documentation given in the file equation.cpp.

  BONUS: It is up to you as to how to return a meaningful value when a, b, and c are all zero. If you implement this, show how it works in solver.cpp

  The data types in getRoots right now are missing: the code does not compile.

  There should be a single function getRoots to both return the number of roots and to computer the real roots (pass the arguments for the roots by reference as described in §6.12).
  Do not do any input nor output in getRoots.

In the file unittest_equation.cpp

- write unit tests to test the function getRoots

Since the user may choose to enter 0 for the value of *a* (in code in the file solver.cop),
deal with this error appropriately (e.g. print an error message after reading the input value and do not call the function to determine the number of roots).

BONUS: deal with the case when a, b, and c are all zero

You will probably want to use the `sqrt` function using `#include <cmath>` which you can look up following the link http://www.cplusplus.com/reference/cmath.

TO SUBMIT WITH BRIGHTSPACE AS A SINGLE ZIP FILE:

1. The file **solver.cpp** that implements the program that interacts with the user.
   This file has the main.
   Keep the #include equation.cpp as given in the file.
   This file also has the functions to read the values, output the equation, and display the results.
   Document the functions.
2. The file **equation.cpp** that implements the function getRoots as documented in the file
3. Some "screen captures" or "print screens" in a pdf file that show that your program in solver.cpp works.
   Assume that all the input is numeric, i.e. a, b, and c are given numbers.
4. The file **unittest_equation.cpp** that implements the unit test cases using doctest.
   Keep the #include equation.cpp as given in the file.
   The documentation in this file will come from the names of the TEST cases.
   The test cases have been started for you.
5. The file doctest.h
6. Possibly a README.txt that indicates that you did the BONUS.

Do not submit any executables nor project files.