Project 2

CS 1344 Fall 98: Programming in C

Due October 26 at 11:59 PM

Roots of Quadratic Equations

Project 2 must be entirely your own work. Assistance should only be sought or accepted from the course instructor or the TA during their office hours. This assignment will give you the opportunity to learn C++ functions.

Problem Statement:

Write a program to find the roots of a quadratic equation, using the following design which involves the use of six functions. Recall that for a quadratic equation

$$Ax^2 + Bx + C = 0$$

the roots are given by

$$\frac{-B+\sqrt{(B^2-4AC)}}{2A} \text{ and } \frac{-B-\sqrt{(B^2-4AC)}}{2A}$$

Note that the quantity (B^2 - 4AC) is called the **discriminant** (D). If A != 0, then there are three distinct possibilities for D.

(1) If D > 0, then there are two distinct real roots. For example, if A = B = 1 and D = 4, then we have the two roots

$$\frac{-1+\sqrt{4}}{2} = \frac{-1+2}{2} = 0.5 \text{ and}$$

$$\frac{-1-\sqrt{4}}{2} = \frac{-1-2}{2} = -1.5$$

(2) If D = 0, then both roots are real and have the same value. For example, if A = 1, B = 2, and D = 0, then the roots are both

$$\frac{-2+\sqrt{0}}{2} = -1.0$$

(3) If D < 0, then there are two complex number solutions of the form M + Ni and M - Ni. For example, if A = B = 1 and D = -4, then we have the two roots

$$\frac{-1+\sqrt{-4}}{2} = \frac{-1+2i}{2} = \text{-0.5 + 1.0i and}$$

$$\frac{-1-\sqrt{-4}}{2} = \frac{-1-2i}{2} = \text{-0.5 - 1.0i}$$

Note: If A = 0 but B! = 0, then the equation becomes Bx + C = 0 which has the single real solution -C / B. If A = B = 0, then the coefficients do not form a valid equation, and an error message should be written. In your program, all the floating-point numbers and coefficients in your equations should be stored as type double to avoid loss of precision.

Input

The input for your program is to be read from the file, **Project2.in**. Each line of this file has three real numbers that represent the equation coefficients A, B, and C, respectively. It is unknown how many equations are to be solved. A typical input file might have the following data:

```
1.0     -2.0     -3.0
1.0     -2.0     10.0
1.0     -2.0      1.0
0.0     -0.0      1.0
0.0      2.0      4.0
```

Functions

This problem is to be solved using the following Six functions. This is a requirement for this project. If you are following a different functional decomposition, you should at least have one function which returns a value, and another function which uses reference parameters. Your solution should use at least six functions.

1. **PrintHeading (output_stream)** does nothing but print the 7 lines of header information for the tabular output.

2. **GetCoefficients (input_stream, A, B, C)** reads an input data line containing the three coefficients of the quadratic equation.

3. **IsValid (A, B, C):** This function should return the value **true** if the coefficients form a valid equation. Otherwise return **false**.

4. **Solve (A, B, C, num1, num2, NumRealSolns)** actually solves the quadratic equation with coefficients A, B, and C. NumRealSolns is an integer which is either 0, 1, or 2 to represent the number of real solutions to the equation. If the data is invalid (when A = B = 0) then Solve should not be called at all.

    a) If NumRealSolns = 2 (D >= 0) then num1 and num2 are the two roots which may be identical if D=0.

b) If NumRealSolns = 1 (A = 0) then num1 is the single real root and num2 is meaningless.

c) If NumRealSolns = 0 (D < 0) then num1 and num2 are the real and imaginary parts of the complex solutions (M and N). You may assume that num2 is positive. Then the answers are num1 + num2 i and num1 - num2 i.

5. **WriteResultLine (output_stream, A, B, C, num1, num2, NumRealSolns):** This function prints the output line according to the results provided by Solve.

6. **WriteInvalidLine (output_stream, A, B, C):** When A, B, and C do not form a valid equation this function will print the invalid coefficients message shown below.

Output

The output must be written to the file, **Project2.out**. The output generated by the input data above is

```
                        Roots of Quadratic Equations
                      Name of programmer: Jayan K. Nair
------------------------------------------------------------------------
                                    Number
                                      of
       A        B        C        Solutions        Root 1        Root2
------------------------------------------------------------------------
     1.00    -2.00    -3.00         2 Real            3.00        -1.00
     1.00    -2.00    10.00        2 Complex      1.00 + 3.00i  1.00 - 3.00i
     1.00    -2.00     1.00         2 Real            1.00         1.00
     0.00     0.00     1.00     ***** Error: Invalid coefficients ! *****
     0.00     2.00     4.00         1 Real           -2.00
```

Documentation:

Your program should include all the documentation called for in the "Elements of Programming Style," including a header for the main program and readable in-line documentation that describes the purpose of each logical segment of code. White space and indentation for increased readability, a documentation header for each function you write, are also required. All function parameters must also be documented so that a user will understand how they are used. Please look at the Project 2 page, of the class web pages for specific style pointers applicable to this project.

Submission:

You should submit your source code (the *.cpp file) electronically to the automated grader. Do not submit the input or the output file. Before you submit your program, make sure your program runs perfectly and provides right output with the sample test data. Be aware that your program will be tested against a randomly generated input data. You will be allowed a maximum of four submits. Your submissions are archived by the grader and will be hand graded for the entire class for conformance to departmental documentation and coding style standards. This project should be submitted as Project Number 2.

Distribution of Points:

Function breakdown : 10%

(Prototypes of functions you are planning to use, **Due on October 13 in class**)

Correct output : 65%

Documentation : 25%

Total :100%

Bonus points and late penalty:

Bonus points : 10 (2 per day up to a maximum of 10 points)

Late penalty : 20% per each day late

Hints:

1) First of all you should divide your project into simple functions (Functional decomposition).

2) Each function should be designed in such a way that, it will do only a single job.

Incorrect design	ReadValuesAndCalculateOutput( )
Good Design	ReadValue( )
	CalculateResult( )
	OutputResult( )

Even after dividing this way, if a function is still complicated, divide it again into simpler functions, until you reach a stage, where each function can be implemented by a few lines of code.

3) Solve your program incrementally. Do not attempt to code all of it at once. Instead, start with the Read-Input function (For Example: GetCoefficients). Code it first and using a simple main() program test the function. Only when the Read-Input function is working that you should proceed to the next function.

4) Whenever a single value is returned, use a value-returning functions. Use reference parameters, only when you have to pass back information from the functions.