Introduction to LabView: Problem Set 1

Problem 1:

Write a VI with a single numerical control, labeled “Total Time in Seconds” and 3 indicators labeled “Hours”, “Minutes”, and “Seconds”. The user be able to input an integer number of seconds into the control, and the VI should convert that number of seconds into the correct number of hours, minutes, and seconds. For example, 3665 seconds should give a result of 1 hour, 1 minute and 5 seconds.

Problem 2:

Write VI’s which evaluate some simple equations:

1. The area of a triangle, given the base width (*b*) and the height (*H*). Let *b* and *H* be double precision floats. Use a decoration to show a triangle on the front panel, with the height and width parameters indicated on the triangle.
2. The area and perimeter of a circle, given the radius *r*. Make *r* a double precision float. Add a circle decoration to the front panel with the radius parameter indicated.
3. Takes a temperature in Kelvin as an input, and converts to both degrees Celsius and degrees Fahrenheit. Use thermometers as the indicators.
4. The roots of a quadratic equation $y = ax^2+b^2+c$. Take a, b and c as integer inputs and evaluate the quadratic formula to obtain the roots. Make sure to return both roots of the quadratic equation. Can LabView deal with imaginary roots?

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

5. The range (*R*) of a projectile, given the velocity *v*, height *H* and angle θ to the horizontal. The user should be able to input θ in radians. Note, we will later see how we can use the formula node to make constructing this formula much simpler.

$$R = \frac{v^2}{2g}\left[sin(2\theta) + \sqrt{sin^2(2\theta) + \frac{8gH}{v^2}cos^2(\theta)}\right]$$

Problem 3:

An XOR gate is a logic gate with the following truth table:

A	B	A XOR B
TRUE	TRUE	**FALSE**
TRUE	FALSE	**TRUE**
FALSE	TRUE	**TRUE**
FALSE	FALSE	**FALSE**

LabView, of course, already contains an XOR gate. However, for this exercise we will make our own XOR gate using AND, OR and NOT functions. Write a VI with two buttons (labeled A and B for inputs) and an LED indicator for the output. Both the left and right hand sides of this equation give XOR logic. Write VI's which have 2 inputs (A and B), and one output (Q) that implement XOR gates based on both sides of this expression. Test that your XOR gate gives returns the above truth table for all possible arrangements of the two inputs A and B.

$$\overline{\left[\left(\bar{A} \vee (A \wedge B)\right) \wedge \left(\bar{B} \vee (A \wedge B)\right)\right]} = (\bar{A} \wedge B) \vee (A \wedge \bar{B})$$

Use LabView's provided XOR gate and verify that it produces the same outputs.

Problem 4:

Write a VI that simulates betting on the toss of a coin. The user should be able to bet on either "Heads" or "Tails", the VI should generate a random number and then determine if the user won or lost, and display the result of the toss in a string indicator, and the words "WIN" or "LOSE" in a second string indicator. Use a Menu Ring for the control so it displays the word "head" or "tail".

Problem 5:

Write a VI that simulates a simple calculator. The program should have two numeric inputs (A and B) and either a Menu or Text Ring control with the options "Add", "Subtract", "Multiply" and "Divide". The VI should use the inputs A and B, make the selected operation and display the result in and indicator.