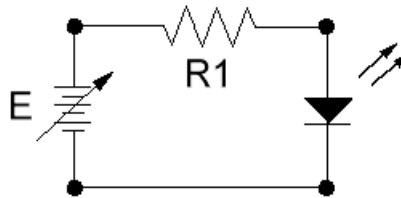


**Problem Description:**

Create a program that simulates the electrical circuit below.



Include a drawing, similar to the one above, in the main window.

The user inputs the resistance of R1, and using option buttons, selects one of three LED's for use in the circuit. See **Table 1** below for details on the 3 LED's. The user can then vary the voltage (E) between 0 and 12 volts using a vertical scroll bar. The following effects on the circuit should be continuously updated and displayed on the screen:

- ▶ power dissipated by R1
- ▶ current through the LED
- ▶ colour of a label (text) located next to the LED, according to the following conditions:
  - if the LED is off, text colour is BLACK;
  - if the LED is on and working at or below the maximum current and power levels, text colour is GREEN;
  - if the LED is on but working above the maximum current or power levels, the text colour is RED.

Characteristics of the 3 LED's are shown in the table below:

<b>Table 1: Electrical specifications for LED's</b>			
<b>Part #</b>	<b>Colour</b>	<b>min. forward voltage</b>	<b>max. current</b>
PR5534S	RED	2.5 V	100.0 mA
SEL-1320G	GREEN	3.0 V	30.0 mA
LNG992CFBW	BLUE	3.5 V	20.0 mA

## **CTEC1831/2009F**

### **Lab #1: Windows Programming with Visual Basic.Net and C#**

---

**Include the following in the program (for a maximum mark of 90%):**

- (a) Include an "About . . ." dialog box. In the dialog box, display your name and the date of the latest revision.
- (b) Include a "File" menu that, when selected, allows you to either exit the program or clear all input values and all calculated values displayed.

**For the additional 10%: include one additional (very) useful feature in the program!**

#### **Programming Notes:**

**You must do two versions: one in Visual Basic.Net and the other in Visual C#.Net.**

- (a) Use variables. They will reduce your typing and make the code much easier to follow.
- (b) Format the values displayed. Limit values to 3 decimal places to the right of the decimal point.
- (c) Use standard naming convention and make the names for controls and variables *meaningful*.
- (d) Ensure that your program will accept only valid inputs. When invalid values are entered, a message box should pop up to inform the user of the problem and prompt him or her to re-enter a valid value.
- (e) Program should be intuitive and easy to use (e.g., minimize keystrokes and the number of times a user has to switch between mouse and keyboard).
- (f) As you are creating the program code, include helpful comments explaining the purpose of your code.

**To be submitted: All files required to load, view, and execute your program.**

**A README file describing the additional feature (for the additional 10%).**

**Due Date: by midnight, Friday, October 16, 2009**

*Late submissions lose 10% per day and receive a zero after 5 days late (Oct. 30).*