Use the Direct Solution Format for all problems on this assignment. Use a word processor to include the source code for all problems involving programming. The first sheet of the source code listing should have your name, the assignment number and the date. The first page does not need to be a separate cover sheet. The problem number and name of the Arduino sketch should be clearly labeled for each problem. In other words, do not just print the source code on otherwise unlabeled sheets of paper. Print the word-processor file and include it with any other sheets of paper you submit as part of your assignment. Do not email the document to your instructor.

1. PWM is a handy technique for controlling the brightness of an LED because the brightness is programmable. But what do you do if you don't have a microcontroller? One approach is to use a potentiometer to adjust the resistance between the voltage supply and the LED. However, to be safe, you should still use a current-limiting resistor in series with the LED.

   a. Sketch a circuit that uses a potentiometer to control the brightness of an LED. Be sure to include a current-limiting resistor to protect the LED when the potentiometer is adjusted to zero resistance.

   b. Build the circuit to test it. Take a photo of your breadboard and include it with your solution.

   The solution to this problem is only a circuit on a breadboard. You will need power from your Arduino to test the circuit, but no sketch is involved. Include brief instructions (say one or two sentences) on how someone should use your circuit and how they should expect your circuit/system to respond.

2. Build a nightlight circuit with an adjustable darkness threshold. The final project should have these features.

   - A photoresistor to detect the ambient light level.
   - An LED to simulate the light of a nightlight.
   - A potentiometer to adjust the threshold light level at for turning on the nightlight.

   Include a photo of your circuit and your Arduino code. Have your code and circuit ready to display at the start of class.

3. Note: The instructions for this assignment are rather long, but the solution is a relatively short Arduino code that uses an if-else if – else structure. The instructions are long so that the intent of the assignment is clear.

   Write an Arduino sketch that converts an analog reading according to the function depicted in Figure 1 on the next page. The x-axis represents the input, i.e. a value returned by the **`analogRead`** function ($0 \le x \le 1023$). The y-axis represents an imaginary physical quantity that is measured by the sensor attached to the analog input pin. In other words, Figure 1 represents a calibration function (calibration curve) for the imaginary sensor. The point of this problem is to write the code that implements the calibration function for the imaginary sensor: You make a reading (x) and your code produces a value of y that would be used for some other purpose in the code. To test your code, use a potentiometer circuit to generate an analog input signal.

The specific steps in completing this assignment are as follows.

a. Build a potentiometer circuit that supplies a voltage to an analog input channel. Note that this same circuit could be used for problem 2, above.

b. Write an Arduino sketch that reads the potentiometer input and produces an output in the Serial Monitor that corresponds to Figure 1. For this specific exercise, use $y_1 = 10$, $y_2 = 20$, $x_1 = 400$, $x_2 = 800$ and $x_{\text{max}} = 1023$ (of course). The shell of a sketch `stepRampAnalogScaling.ino` in Listing 1 may be helpful as a starting point. The reading returned from `analogRead` is the x value in Figure 1.

c. When your code is working, you should be able to adjust the potentiometer to generate a constant analog reading. Your code and circuit should pass the following tests:
   - When the analog reading is any value between 0 and 400, your code should print the reading and a value of 10 (for y) in the Serial Monitor.
   - When the analog reading is any value between 400 and 800, your code should print the reading a value that falls on the sloping line in Figure 1.
   - When the analog reading is any value between 800 and 1023, your code should print the reading and a value of 20 a value of y

For your solution, include your Arduino code listing and a screenshot of the Serial Monitor when the analog input value is between 400 and 800. Have your code ready to run at the start of class. (You can switch codes between problem 2 and 3, but the circuit board should not change for problem 2 and 3.)

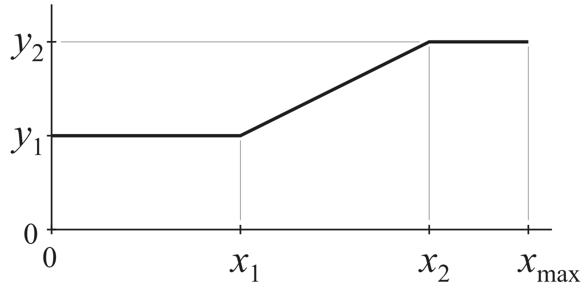

Figure 1. Step-ramp analog scaling function.

Skeleton of solution to Problem 3

```
//  File:  stepRampAnalogScaling.ino
//
//  Convert an analog input signal with a step ramp function

void setup() {
  Serial.begin(9600);
}

void loop() {
  int   reading, x1=400, x2=800, xmax=1023;
  float y, y1=10.0, y2=20.0;

  // -- Maybe insert additional code here?  Or not.

  reading = analogRead(A3);   //  Any analog input pin will do

  // -- INSERT code to calculate y from reading

  Serial.print(reading);
  Serial.print("   ");
  Serial.println(y);

  delay(100);   //  Not required; Helps in viewing Serial Monitor
}
```

Listing 1   Shell (incomplete code) of `stepRampAnalogScaling.ino`.