For this problem, you have to "build" (on paper only) a circuit connected to an Arduino that would turn a red LED on if the voltage of a 9V battery were to go below 7V.

Using the material presented in the Arduino Unit 3, lecture, come up with:

1. the schematics for circuit that can be added to an arduino powered by a 9V battery
2. a sketch that can be loaded in the arduino, and that will turn ON an LED if the battery voltage goes below 7V.
3. the calculations you did to create the various constants used in your sketch.

Extra points will be given to schematics generated using Fritzing (see this link for a video tutorial.)
* hw6-1
* Nox Yu
* This program turns on a red LED if the voltage of a 9V-battery goes below 7V
*/

int analogPin = A3;
int threshold = 716;
int v = 0;
int builtinLed = 13;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
pinMode(builtinLed, OUTPUT);
digitalWrite(builtinLed, LOW);
}

void loop() {
    // put your main code here, to run repeatedly:
    v = analogRead(analogPin);
    if (v < threshold) {
        digitalWrite(builtinLed, HIGH);
    }
}
Problem 2: Integer Size

Write a sketch that will help you figure out what the size of an integer (8 bits, 16 bits, 32 bits, 64 bits?) is on the Mega2560 Arduino. Explain how to use your sketch in the header of your sketch. Please include screen captures of your running sketch to support your claim.

```
int count = 1;
int exponent = 0;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    if(count >= 0){
        Serial.println(count);
        exponent++;
        count = pow(2, exponent);
    }
    delay(500);
}
```

So the size of an int in Arduino is 16 bit.
In the first two screenshots are the sketch I wrote that helps me figure out what size an integer is on the Mega2560 Arduino. The third screenshot are some examples of integers I entered that support my claim of this sketch working correctly.

**PROBLEM 3:**
To use the Arduino to measure the voltage of a battery we can use a potentiometer on an analog input pin. Potentiometers are not very precise; there is always slight variation that is expected when using them. In the example given from lecture, when the analogRead() function is called, it typically prints an integer to the Serial Monitor as its output. At 2.500V the value expected to be returned by the Arduino is: \( \frac{1024}{5V} \times 2.500V = 512 \). At 2.501V the value expected is: \( \frac{1024}{5V} \times 2.501V = 512.2048 \). Therefore, if the voltage changes from 2.500V to 2.501V the number returned by the function is most likely going to be the same value as before, or at least in the same range.

**PROBLEM 4:**

```cpp
/*
 * Julia Chen
 * CSC270, Homework 6, Problem 4
 * 
 * Modified from Arduino Unit 3 demo video and Fade tutorial.
 * 
 * Sketch that makes red LED go OFF-->ON gradually,
 * then ON-->OFF gradually using PWM.
 * Green LED behaves opposite of red simultaneously.
 */

int LED0 = 3; // PWM pin for red LED
int LED1 = 4; // PWM pin for green LED
int brightness = 0; // how bright the LED is
int fadeAmt = 5; // how many points to fade the LED by

void setup() {
  // put your setup code here, to run once:
  pinMode(LED0, OUTPUT); // red LED set as output
  pinMode(LED1, OUTPUT); // green LED set as output
}

void loop() {
  // put your main code here, to run repeatedly:
  // set the brightness of LEDs
  analogWrite(LED0, brightness);
  analogWrite(LED1, 255 - brightness);

  // change the brightness for next loop
  brightness = brightness + fadeAmt;

  // reverse the direction of the fading at the ends of the fade
  if (brightness <= 0 || brightness >= 255) {
    fadeAmt = -fadeAmt;
  }
  delay(50); // wait 50 milliseconds to see dimming
}
```
Above is the sketch used to make 2 LEDs (green and red) change their intensities from totally OFF to totally ON, gradually, using pulse width modulation (PWM). The red LED goes OFF to ON gradually then ON to OFF gradually, with the green LED acting opposite simultaneously. I adapted the sketch from the PWM demo example given and the sketch given in the Fade tutorial. Below is my Arduino wired up and behaving as expected.