The prelab portion of this lab has to be completed at home and presented to a TA at the BEGNIINING of your lab section. The prelab may contain written assignments and/or assignments to implement using your own Arduino kit. Before starting each lab, present your work papers and/or your working implementations to a TA and get their signature and keep this sheet for your records. Prelabs are individual assignments. The lab portions may be worked on in groups of two.

Lab 1 - Prelab/Home Portion

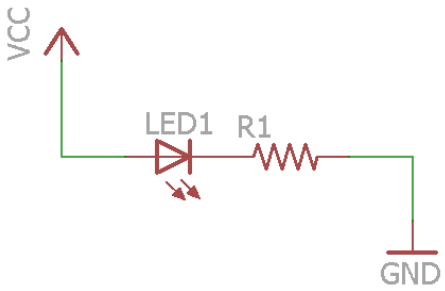
1) There are two functions that must be included in every Arduino program – list them and explain what they do.

2) Describe what the **pinMode** function is used for

3) Write some code (at most five lines) that will: a) configure pin `5` as a digital input and pin `13` as a digital output; b) read a value from pin `5`; and c) write the value from pin `5` to pin `13`

4) Write some code (at most five lines) that will: a) configure pin `9` as a PWM output; b) read a value from pin `A0`; c) write the value from pin `A0` to pin `9`.
    a. ***Hint:*** *The values from the* ***analogRead*** *function are from 0 to 1023 while the* ***analogWrite*** *command only accepts values from 0 to 255. You'll need to adjust the value before writing it to pin 9!*

5) Calculate the value of R1 in the following circuit (V_{cc} = 5 volts; forward voltage drop of diode = 2 volts; recommended current = 20 milliamps)



6) In a few sentences, describe clearly the principle behind how software-based button debouncing is achieved in Listing 2-5 of the textbook.

Lab 1 - Lab Portion

Part 1. Copy the Blink program from Figure 1-13 (page 17) in the textbook and run it on your Arduino. After verifying that the Blink program works, modify the program to make the on-board LED attached to pin #13 blink out "SOS" in Morse code. Use a 250 millisecond for short pulses and a 750 millisecond for long pulses.

Part 2. Write a program that uses two pushbuttons to increase or decrease the brightness of the LED on pin #11

a. Your circuit should include an "up" button and a "down" button
b. Pressing the up button should add ≈ 10% brightness to the LED
c. Pressing the down button should subtract ≈ 10% brightness from the LED

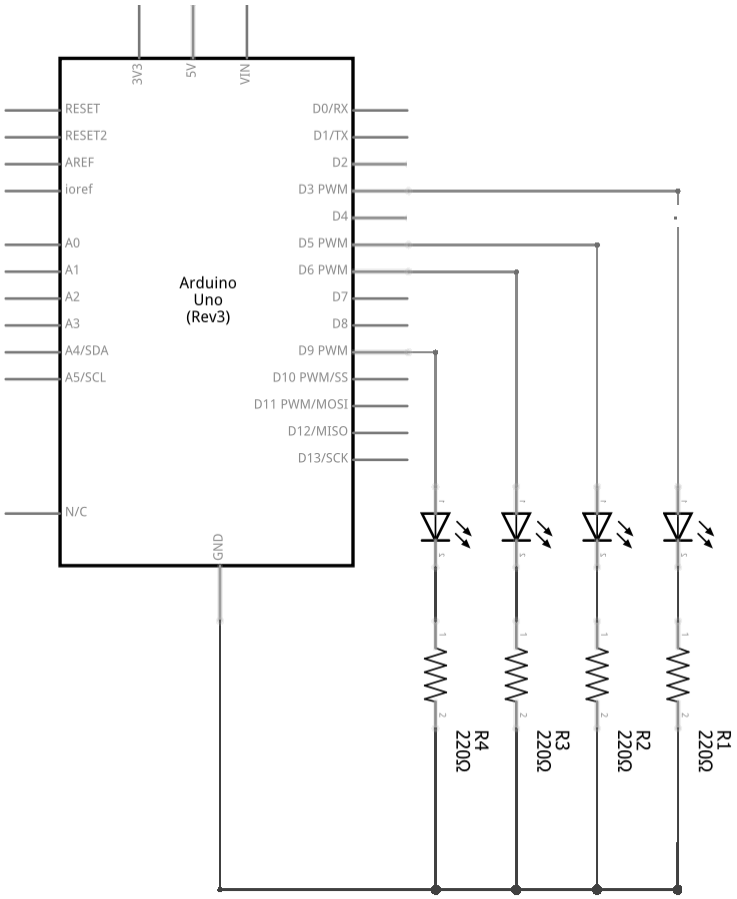
Hint: use a variable to hold the current duty cycle (dc) value. In loop(), check the up button; if pressed, add 25 to the variable. Also check the down button; if pressed, subtract 25 from the variable. Observe what happens as you exceed max dc or min dc values possible and be ready to explain it to the TA. How would you fix it if you had more time?

Part 3. Write a program that implements a four-bit binary up counter, displaying the output on LEDs. The program should automatically increment every 1000 milliseconds. Once the counter reaches its maximum value, it should rollover back to zero.

You will need the following components:

- 4x LEDs
- 4x 220Ω resistors
- Jumper wires
- Breadboard

The schematic for this circuit is included on the next page.


3V3
5V
VIN
RESET
RESET2
AREF
Ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
Arduino
Uno
(Rev3)
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
R4
220Ω
R3
220Ω
R2
220Ω
R1
220Ω