

Problems 1, 2 and 4 should use the Direct Solution Format. Problem 3 should use the Engineering Solution Format.

1. Write an Arduino sketch that prints the integers in increasing order from 1 to n and then in decreasing order from n-1 to 1. For this problem, “print” means send the output to the serial monitor. Use n=7, but declare n as variable. (See problem 2 for the motivation.) Make sure that the value of n appears only once in the serial monitor display. Print all of the values on a single row (not a column), and leave at least two blank spaces between each value on the row.

Have the printing occur *only once* by putting the code in the setup() function, and leaving the loop() function empty. The following shell of an Arduino sketch shows the basic structure of the solution.

```
int n = 7;

void setup() {
    // -- your code goes here
}

void loop() {    // Leave loop() empty
}
```

On the solution you turn in for grading, list the text of your program in a word-processing document. Include the output of your program in the word-processing document.

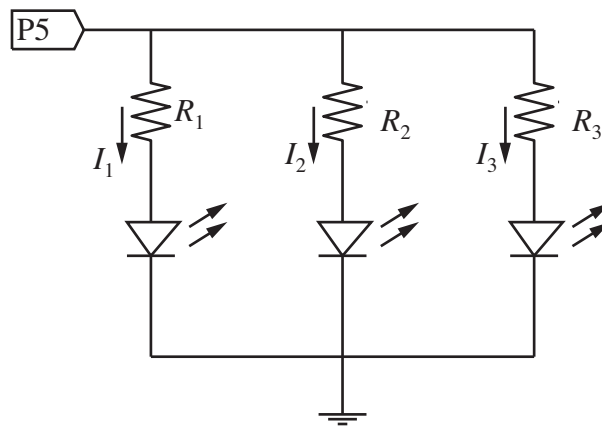
Hint: Since the program executes once, you may not always catch the output in the serial monitor. In that case, just press the reset button on the board, which will cause the program to start again.

2. Extend the preceding program by repeating the print out to values of n=3, 5, 7, 9 in the same program. The output of your program in the serial monitor should look like the following:

```
1 2 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
```

Bring your Arduino and laptop to class. Print out your program listing and have your program running at the start of class. The instructors will check the function of your program.

3. The following circuit diagram shows how an engineer plans to use a single Arduino output pin (pin 5) to control three LEDs. The engineer knows that a current-limiting resistor is needed for each LED. The engineer also knows that the maximum current that Pin 5 can supply is 40 mA. The design will use three identical LEDs and three identical current limiting resistors, $R_1 = R_2 = R_3$.
- Apply Kirchoff's current law to find an equation for I (the current) from P5. The result is a formula
 - Find the equivalent resistance for the circuit.
 - What is an appropriate value of the resistance if the current limit on the Arduino is going to be met?



4. Read exercise CIRC-02 in the manual of the Sparkfun Inventor's Kit and build the 8 LED circuit on a breadboard. Enter or download from the homework web page, the two codes listed on the next page: `LED_row_lights_manual.pde` and `LED_row_lights.pde`. Run the codes and answer the following questions.
- Describe the lighting pattern for the two codes. Are the lighting patterns the same?
 - Suppose the following code was entered at the bottom (and inside) of the `loop()` function. Describe the pattern of LED lights. Note that adding this code will result in two loops inside the `loop()` function.

```
for ( i=7; i>=0; i-- ) {
  digitalWrite( ledPin[i], HIGH );
  delay( wait_interval );
  digitalWrite( ledPin[i], LOW );
}
```

LED_row_lights_manal.pde

```
// File: LED_row_lights_manual.pde
//
// Blink a row of LEDs connected to digital output pins
// Use manual calls to each LED -- not the smart way to do it

void setup() {
  pinMode( 2, OUTPUT );    // Prepare pins 2 through 9 for digital output
  pinMode( 3, OUTPUT );
  pinMode( 4, OUTPUT );
  pinMode( 5, OUTPUT );
  pinMode( 6, OUTPUT );
  pinMode( 7, OUTPUT );
  pinMode( 8, OUTPUT );
  pinMode( 9, OUTPUT );
}

void loop() {
  int wait_interval = 200;

  // -- Turn each pin on, wait, then turn the pin off
  digitalWrite( 2, HIGH ); delay( wait_interval ); digitalWrite( 2, LOW );
  digitalWrite( 3, HIGH ); delay( wait_interval ); digitalWrite( 3, LOW );
  digitalWrite( 4, HIGH ); delay( wait_interval ); digitalWrite( 4, LOW );
  digitalWrite( 5, HIGH ); delay( wait_interval ); digitalWrite( 5, LOW );
  digitalWrite( 6, HIGH ); delay( wait_interval ); digitalWrite( 6, LOW );
  digitalWrite( 7, HIGH ); delay( wait_interval ); digitalWrite( 7, LOW );
  digitalWrite( 8, HIGH ); delay( wait_interval ); digitalWrite( 8, LOW );
  digitalWrite( 9, HIGH ); delay( wait_interval ); digitalWrite( 9, LOW );
}
```

LED_row_lights.pde

```
// File: LED_row_lights.pde
//
// Blink a row of LEDs connected to digital output pins

int ledPin[] = {2, 3, 4, 5, 6, 7, 8, 9};

void setup() {
  for ( int i=0; i<8; i++) {
    pinMode( ledPin[i], OUTPUT );
  }
}

void loop() {
  int wait_interval = 200;
  int i;

  for ( i=0; i<8; i++ ) {
    digitalWrite( ledPin[i], HIGH );
    delay( wait_interval );
    digitalWrite( ledPin[i], LOW );
  }
}
```