Challenge 1 normal:

```
// LED pins in an array.
const int led[] = {14, 13, 12, 11, 10, 9, 8, 7, 6};

// Counts the LEDs in order.
const int ledCount = 8;

// Push Button pin.
const int PB = 3;

// Pushbutton state.
bool state = 1;

void setup()
{
  // States that the pushbutton is an input.
  pinMode(PB, INPUT);

  // if "i" or the counter of the LEDs  is equal to 0, and if i
  // is less than or equal to the led count of the LEDs, increase
  // the counter "i".
  for (int i = 0; i <= ledCount; i++){

    // Declares that the LEDs are outputs based on the value of
    // the counter.
          pinMode(led[i], OUTPUT);
  }
}

void loop()
{

  // When the pushbutton is pressed:
  if(digitalRead(PB)== 1){

    // Switch the state to 0 or 1.
    state = !state;

    // This code loops after the pushbutton has been pressed.
    while(digitalRead(PB) == 1){}
  }

  // If the state of the pushbutton is 1:
```

```
if(state == 1)    {

    // If the counter of the LEDs "i" is equal to 0, and if i
   // is less than or equal to the led count of the LEDs,
  // increase the value of the counter "i".
    for(int i = 0; i <= ledCount; i++) {

       // The led corresponding to the value of counter "i"
      // is turned on.
        digitalWrite(led[i], HIGH);

      // 0.2 second delay so led's do not turn on at the same
     // time.
        delay(200);
    }

    // If the counter of the LEDs "i" is equal to 0, and if i
   // is less than or equal to the led count of the LEDs,
  // increase the value of the counter "i".
    for(int i = 0; i <= ledCount; i++){

      // The led corresponding to the value of counter "i"
      // is turned off.
      digitalWrite(led[i], LOW);

      // 0.2 second delay.
      delay(200);
    }

}

// If the state of the pushbutton is 0:
if(state == 0){

  // If the counter of the LEDs "i" is equal to the led count
  // of the LEDs, and if "i" is greater than 0, decrease the
  // value of the counter "i".
  for(int i = ledCount; i > 0; i--) {

      // The led corresponding to the value of counter "i"
     // is turned on.
         digitalWrite(led[i], HIGH);

      // 0.2 second delay.
```

```
        delay(200);
    }

    // If the counter of the LEDs "i" is equal to the led count
   // of the LEDs, and if "i" is greater than 0, decrease the
  // value of the counter "i".
  for(int i = ledCount; i > 0; i--){

      // The led corresponding to the value of counter "i" is
     // turned off.
      digitalWrite(led[i], LOW);

      // 0.2 second delay.
      delay(200);
    }
 }
}
```

Challenge 1 with function:

```
// LED pins in an aray.
const int led[] = {14, 13, 12, 11, 10, 9, 8, 7, 6, 5};

// Counts the LEDs in order.
const int ledCount = 8;

// Pushbutton pin.
const int PB = 3;

// Pushbutton state.
bool state = 1;

void setup()
{
  // States that the pushbutton is an input.
  pinMode(PB, INPUT);

  // if "i" or the counter of the LEDs  is equal to 0, and if i
  // is less than or equal to the led count of the LEDs, increase
  // the counter "i".
  for (int i = 0; i <= ledCount; i++){
```

```
    // Declares that the LEDs are outputs based on the value of
    // the counter.
          pinMode(led[i], OUTPUT);
  }
}

/*****************************************************************
This segment includes functions that get called in if you press
the pushbutton. These functions controll the movement of the led
either turned on and off normally or reversed.
*****************************************************************/


// Function thats declates the start of the normal motion of the
// led's.
int RegularMotion(){

    //If the counter of the LEDs "i" is equal to 0, and if i
   // is less than or equal to the led count of the LEDs,
  // increase the value of the counter "i".
    for(int i = 0; i <= ledCount; i++) {

            // The led corrospinding to the value of counter "i"
           // is turned on.
         digitalWrite(led[i], HIGH);

            // 0.2 second delay so the led's do not turn on
           // at the same time.
         delay(200);
    }

    // If the counter of the LEDs "i" is equal to 0, and if "i"
   // is less than or equal to the led count of the LEDs,
  // increase the value of the counter "i".
    for(int i = 0; i <= ledCount; i++){

       // The led corrospinding to the value of counter "i"
      // is turned off.
       digitalWrite(led[i], LOW);

       // 0.2 second delay so the led's do not turn off at the
      // same time
       delay(200);
```

```
    }
}

// Function that declares the start of the reversed motion of the
// led's.
int ReverseMotion(){

    // If the counter of the LEDs "i" is equal to the led count
   // of the LEDs, and if "i" is greater than 0, decrease the
  // value of the counter "i".
    for(int i = ledCount; i > 0; i--) {

        // The led corrosponding to the value of counter "i"
       // is turned on.
            digitalWrite(led[i], HIGH);

        // 0.2 second delay so the led's do not turn on
       // at the same time.
            delay(200);
    }

    // If the counter of the LEDs "i" is equal to the led count
   // of the LEDs, and if "i" is greater than 0, decrease the
  // value of the counter "i".
    for(int i = ledCount; i > 0; i--){

        // The led corrosponding to the value of counter "i"
       // is turned off.
        digitalWrite(led[i], LOW);

        // 0.2 second delay so the led's do not turn off
       // at the same time.
        delay(200);
    }
}

void loop()
{

  // When the pushbutton is pressed:
  if(digitalRead(PB)== 1){

    // Switch the state to 0 or 1.
    state = !state;
```

```
    // This code loops after the pushbutton has been pressed.
    while(digitalRead(PB) == 1){}
  }


  // If the state of the pushbutton is 1:
  if(state == 1){

    // Calls in the function that declares regular motion.
    RegularMotion();
  }

  // If the state of the pushbutton is 0:
  if(state == 0){

    // Calls in the function thatd eclares reversed motion.
    ReverseMotion();
  }

}
```

Challenge 1 with port register:

```
// Declares pin for pushbutton.
const int PB = 12;

// Declares state of pushbutton.
bool state = 1;

void setup()
{
          // Declares leds as outputs cossosponding to what pin they
   // are connected to.
    DDRD = 0b111111111;
}

void loop()
{
  // If the pushbutton is pressed:
  if(digitalRead(PB)==1){
```

```
  // switch state to 0 or 1.
  state = !state;

  // This code loops after the pushbutton has been pressed.
  while(digitalRead(PB) == 1){}
}

// If the state of the pushbutton is 1:
if(state == 1){

  // Turns on led in 7th pin.
        PORTD = 0b10000000;
        delay(200); // 0.2 second delays

  // Turns on led in 6th pin and keeps led in 7th pin on.
  PORTD = 0b11000000;
  delay(200);

  // Turns on led in 5th pin and keeps leds in 6th and 7th pin
  // on.
  PORTD = 0b11100000;
  delay(200);

  // Turns on led in 4th pin and keeps leds in 5th, 6th, and 7th
  // pin on.
  PORTD = 0b11110000;
  delay(200);

  // Turns on led in 3rd pin and keeps leds in 4th, 5th, 6th,
  // and 7th pin on.
  PORTD = 0b11111000;
  delay(200);

  // Turns on led in 2rd pin and keeps leds in 3rd, 4th, 5th,
  // 6th, and 7th pin on.
  PORTD = 0b11111100;
  delay(200);

  // Turns on led in 1st pin and keeps leds in 2nd, 3rd, 4th, 5th
  // 6th, and 7th pin on.
  PORTD = 0b11111110;
  delay(200);

  // Turns on led in the 0th pin and keeps leds in 1st, 2nd, 3rd,
```

```
  // 4th, 5th, 6th, and 7th pin on.
  PORTD = 0b11111111;
  delay(200);

  // Turns off led in 7th pin.
  PORTD = 0b01111111;
  delay(200);

  // Turns off led in 6th pin and keeps led in 7th pin off.
  PORTD = 0b00111111;
  delay(200);

  // Turns off led in 5th pin and keeps led in 6th and 7th pin
  // off.
  PORTD = 0b00011111;
  delay(200);

  // Turns off led in 4th pin and keeps leds in 5th, 6th, and
  // 7th pin off.
  PORTD = 0b00001111;
  delay(200);

  // Turns off led in 3rd pin and keeps leds in 4th, 5th, 6th
  // and 7th pin off.
  PORTD = 0b00000111;
  delay(200);

  // Turns off led in 2nd pin and keeps leds in 3rd, 4th, 5th
  // 6th, and 7th pin off.
  PORTD = 0b00000011;
  delay(200);

  // Turns off led in 1st pin and keeps leds in 2nd, 3rd, 4th,
  // 5th, 6th, and 7th pin off.
  PORTD = 0b00000001;
  delay(200);

  // Turns off led in the 0th pin and keeps leds in 1st, 2nd,
  // 3rd, 4th, 5th, 6th, and 7th pin off.
  PORTD = 0b00000000;
  delay(200);
  }

// If the state of the pushbutton is 0:
```

```
if(state == 0){

  // Turns on led in 0th pin.
  PORTD = 0b00000001;
  delay(200); // 0.2 second delay

  // Turns on led in 1st pin and keeps led on in 0th pin.
  PORTD = 0b00000011;
  delay(200);

  // Turns on led in 2rd pin and keeps led on in 0th and 1st
  // pin.
  PORTD = 0b00000111;
  delay(200);

  // Turns led on in 3rd pin and keeps led on in 0th, 1st, and
  // 2nd pin.
  PORTD = 0b00001111;
  delay(200);

  // Turns on led in 4th pin and keeps led on in 0th, 1st, 2nd,
  // and 3rd pin.
  PORTD = 0b00011111;
  delay(200);

  // Turns on led in 5th pin and keeps led on in 0th, 1st, 2nd,
  // 3rd, and 4th pin.
  PORTD = 0b00111111;
  delay(200);

  // Turns on led in 6th pin and keeps led on in 0th, 1st, 2nd,
  // 3rd, 4th, and 5th pin.
  PORTD = 0b01111111;
  delay(200);

  // Turns on led in 7th pin and keeps led on in 0th, 1st, 2nd,
  // 3rd, 4th, 5th, and 6th pin.
  PORTD = 0b11111111;
  delay(200);

  // Turns off led in 0th pin,
  PORTD = 0b11111110;
  delay(200);
```

```
    // Turns off led in 1st pin and keeps led off in 0th pin.
    PORTD = 0b11111100;
    delay(200);

    // Turns off led in 2nd pin and keeps led off in 0th and 1st
    // pin.
    PORTD = 0b11111000;
    delay(200);

    // Turns off led in 3rd pin and keeps led off in 0th, 1st, and
    // 2nd pin.
    PORTD = 0b11110000;
    delay(200);

    // Turns off led in 4th pin and keeps led off in 0th, 1st, 2nd,
    // and 3rd pin.
    PORTD = 0b11100000;
    delay(200);

    // Turns off led in 5th pin and keeps led off in 0th, 1st, 2nd,
    // 3rd, and 4th pin.
    PORTD = 0b11000000;
    delay(200);

    // Turns off led in 6th pin and keeps led off in 0th, 1st, 2nd,
    // 3rd, 4th, and 5th pin.
    PORTD = 0b10000000;
    delay(200);

    // Turns off led in 7th pin and keeps led off in 0th, 1st, 2nd,
    // 3rd, 4th, 5th, and 6th pin.
    PORTD = 0b00000000;
    delay(200);
  }

}
```

Traffic light normal:

```
// vabiables for traffic lights (west).
const int r1 = 13, y1 = 12, g1 = 11;
```

```
// vabiables for traffic lights (south).
const int r2 = 10, y2 = 9, g2 = 8;
// vabiables for traffic lights (east).
const int r3 = 7, y3 = 6, g3 = 5;
// vabiables for traffic lights (north).
const int r4 = 4, y4 = 3, g4 = 2;


void setup() // Starting output pins.
{
  pinMode(r1, OUTPUT);
  pinMode(y1, OUTPUT);
  pinMode(g1, OUTPUT);
  pinMode(r2, OUTPUT);
  pinMode(y2, OUTPUT);
  pinMode(g2, OUTPUT);
  pinMode(r3, OUTPUT);
  pinMode(y3, OUTPUT);
  pinMode(g3, OUTPUT);
  pinMode(r4, OUTPUT);
  pinMode(y4, OUTPUT);
  pinMode(g4, OUTPUT);
}

void loop()
{
  // Red light (east) and red light (west) are on, and green
  // light (north) and green light (south) are on, thus cars can
  // drive north/south.
  digitalWrite(r1, HIGH);
  digitalWrite(r3, HIGH);
  digitalWrite(r2, LOW);
  digitalWrite(r4, LOW);
  digitalWrite(g2, HIGH);
  digitalWrite(g4, HIGH);
  delay(5000); // 5 second delay.

  // Yellow light (north) and yellow light (south) are on, so
  // cars driving north/south should slow down. East and west
  // red lights are still on and north and south green lights turn
  // off.
  digitalWrite(g2, LOW);
  digitalWrite(g4, LOW);
  digitalWrite(y2, HIGH);
```

```
  digitalWrite(y4, HIGH);
  delay(2000); // 2 second delay.

  // // North and south red lights turn on, and it is a
  // short period of time where all red lights are on, minimizing
  // accidents.
  digitalWrite(y2, LOW);
  digitalWrite(y4, LOW);
  digitalWrite(r2, HIGH);
  digitalWrite(r4, HIGH);
  delay(1000);// 1 second delay.

  // Red light (north) and red light (south) are on, and green
  // light (east) and green light (west) are on, thus cars can
  // drive east/west.
  digitalWrite(r1, LOW);
  digitalWrite(r3, LOW);
  digitalWrite(g1, HIGH);
  digitalWrite(g3, HIGH);
  delay(5000); // 5 second delay.

  // Yellow light (east) and yellow light (west) are on, so cars
  // driving east/west should slow down. North and south red
  // lights are still on, and east and west green lights turn off.
  digitalWrite(g1, LOW);
  digitalWrite(g3, LOW);
  digitalWrite(y1, HIGH);
  digitalWrite(y3, HIGH);
  delay(2000); // 2 secold delay.

  // East and west red lights turn on and it is a
  // short period of time where all red lights are on,
  // minimizing accidents.
  digitalWrite(y1, LOW);
  digitalWrite(y3, LOW);
  digitalWrite(r1, HIGH);
  digitalWrite(r3, HIGH);
  delay(1000); // 1 second delay.

}
```

Traffic light functions:

```
// Declaring input pins using arrays.

// Pins for north lights.
const int setA1[] = {4, 3, 2};

// Pins for south lights.
const int setA2[] = {10, 9, 8};

// Pins for east lights.
const int setB1[] = {7, 6, 5};

// Pins for west lights.
const int setB2[] = {13, 12, 11};


void setup() // Stating output pins.
{
  // "i" starts at 0 and goes through setA1 array and declares
  //each index as an output.
  for (int i = 0; i < 3; i ++)
    pinMode(setA1[i], OUTPUT);

  // "i" starts at 0 and goes through setA2 array and declares
  //each index as an output.
  for (int i = 0; i < 3; i ++)
    pinMode(setA2[i], OUTPUT);

  // "i" starts at 0 and goes through setB1 array and declares
  //each index as an output.
  for (int i = 0; i < 3; i ++)
    pinMode(setB1[i], OUTPUT);

  // "i" starts at 0 and goes through setB2 array and declares
  //each index as an output.
  for (int i = 0; i < 3; i ++)
    pinMode(setB2[i], OUTPUT);
}

/*****************************************************************
This segment includes functions that turn on or off various lights
from different cardinal directions, thus controlling traffic
lights.
*****************************************************************/
```

```
// Function that declares red light east and west to turn on and
// green light north and south to turn on.
int GreenLightNS (){

  // Red lights east and west on.
  digitalWrite(setB1[0], HIGH);
  digitalWrite(setB2[0], HIGH);

  // red lights north and south off.
  digitalWrite(setA1[0], LOW);
  digitalWrite(setA2[0], LOW);


  // Green lights north and south on.
  digitalWrite(setA1[2], HIGH);
  digitalWrite(setA2[2], HIGH);

  // 4 second delay.
  delay(5000);

}

// Function that declates yellow lights north and south to turn
// on and green lights north and south to turn off.
int YellowLightNS (){

  // Green lights north and south off.
  digitalWrite(setA1[2], LOW);
  digitalWrite(setA2[2], LOW);

  // Yellow lights north and south on.
  digitalWrite(setA1[1], HIGH);
  digitalWrite(setA2[1], HIGH);

  // 2 second delay.
  delay(3000);
}

// Function that declares yellow lights north and south to turn
// off and red lights north and south to turn on, minimizing
// accidents.
int AllRedLights (){
```

```
  // Yellow lights north and south off.
  digitalWrite(setA1[1], LOW);
  digitalWrite(setA2[1], LOW);

  // Red lights north and south on
  digitalWrite(setA1[0], HIGH);
  digitalWrite(setA2[0], HIGH);

  // 2 second delay
  delay(2000);
}

// Function thats turns on green lights east and west and turns
// off red lights east and west.
int GreenLightEW (){

  // Turns off led lights east and west.
  digitalWrite(setB1[0], LOW);
  digitalWrite(setB2[0], LOW);

  // Turns on green lights east and west.
  digitalWrite(setB1[2], HIGH);
  digitalWrite(setB2[2], HIGH);

  // 4 secold delay.
  delay(5000);

}

// Function that declares yellow lights east and west to turn on.
int YellowLightEW (){

  // Green lights east and west turn off.
  digitalWrite(setB1[2], LOW);
  digitalWrite(setB2[2], LOW);

  // YEllow lights east and west turn on.
  digitalWrite(setB1[1], HIGH);
  digitalWrite(setB2[1], HIGH);

  // 2 secold delay.
  delay(3000);
}
```

```
// Function that declares red lights east and west to turn on.
// while red lights north and south are on, minimizing accidents.
int AllRedLightsTwo (){

  // Yellow lights east and west turn off.
  digitalWrite(setB1[1], LOW);
  digitalWrite(setB2[1], LOW);

  // Red lights east and west turn on.
  digitalWrite(setB1[0], HIGH);
  digitalWrite(setB2[0], HIGH);

  // 1 second delay.
  delay(2000);
}

void loop()
{

  // Calls in the function thats turns on green lights north and
  // south.
  GreenLightNS ();

  // Calls in function that turns on north and south yellow lights.
  YellowLightNS ();

  // Calls in function that turns on north and south red lights.
  AllRedLights ();

  // Calls in function that turns on east and west green lights.
  GreenLightEW ();

  // Calls in function that turns on east and west yellow lights.
  YellowLightEW ();

  // Calls in function that turns on east and west red lights.
  AllRedLightsTwo ();

}
```

Traffic light port register:

```
void setup()
{
  // Declaring pins and outputs of leds.
  DDRD = 0b111111100;
}

void loop()
{
  // Declaring that red light (east and west) are on aswell
  // as green light (north and south) are on.
  PORTD = 0b10000100;
  delay(5000); // 5 second delay.

  // Declaring that yellow lights (north and south) are on and
  // red lights (east and west) are still on.
  PORTD = 0b10001000;
  delay(2000); // 2 second delay.

  // Declaring that all red lights are on for a breif moment.
  PORTD = 0b10010000;
  delay(1000); // 1 second delay.

  // Decraling that red lights (north and south) are on aswell
  // as green lights (east and west) are on.
  PORTD = 0b00110000;
  delay(5000); // 5 second delay.

  // Declaring that yellow lights (east and west) are on and red
  // lights (north and south) are still on.
  PORTD = 0b01010000;
  delay(2000); // 2 second delay.

  // Declaring that all red lights are on for a breif moment.
  PORTD = 0b10010000;
  delay(1000); // 1 second delay.
}
```

RGB normal:

```
// RGB LED pins
const int R = 6;
```

```
const int B = 5;
const int G = 3;

// Potentiometers
const int PM = A0;
const int PM2 = A1;
const int PM3 = A2;

//Potentiometer readings
int p;
int p2;
int p3;

//hue of red, blue, and green.
int h;
int h2;
int h3;

void setup()
{

  // States the input and output pins.
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(PM, INPUT);
  pinMode(PM2, INPUT);
  pinMode(PM3, INPUT);
}

void loop()
{

  // The first potentiometer reading is the value of
  // potentiometer 1.
  p = analogRead(PM);

  // the hue of red is the map of the first potentiometer reading.
  h = map(p, 0, 1023, 0, 255);

  // The second potentiometer reading is the value of
  // potentiometer 2.
  p2 = analogRead(PM2);
```

```
  // the hue of blue is the map of the second potentiometer
  // reading.
  h2 = map(p2, 0 ,1023, 0, 255);

  // The third potentiometer reading is the value of
  // potentiometer 3.
  p3 = analogRead(PM3);

  // the hue of green is the map of the third potentiometer
  // reading.
  h3 = map(p3,0, 1023, 0, 255);

  // displays the hue of red on the RGB led.
  analogWrite(R, h);

  // displays the hue of blue on the RGB led.
  analogWrite(B, h2);

  // displays the hue of green on the RGB led.
  analogWrite(G, h3);




}
```

RGB function:

```
// RGB LED pins in an array.
const int RGB[] = {6, 5, 3};

// Potentiometers in an array.
const int PM[] = {A0, A1, A2};

//Potentiometer readings
int p;
int p2;
int p3;

//hue of red, blue, and green.
int h;
```

```
int h2;
int h3;

void setup()
{
  // "i" starts at 0 and goes through RGB array and declares
  //each index as an output.
  for(int i = 0; i < 3; i++)
    pinMode(RGB[i], OUTPUT);

  // "i" starts at 0 and goes through PM array and declares
  //each index as an output.
  for(int i = 0; i < 3; i++)
    pinMode(PM[i], OUTPUT);
}

/*****************************************************************
This section includes functions that help to determine the hues
of each colour in the RGB led aswell as mix each hue together,
forming new colours.
*****************************************************************/

// This function declates the reading of the first potentiometer
// and the hue of the red section of the RGB led.
int RedSpectrum(){

  // The first potentiometer reading is the value of
  // potentiometer 1 on index value 0.
  p = analogRead(PM[0]);

  // the hue of red is the map of the first potentiometer reading.
  h = map(p, 0, 1023, 0, 255);
}

// This function declates the reading of the second potentiometer
// and the hue of the blue section of the RGB led.
int BlueSpectrum(){

  // The second potentiometer reading is the value of
  // potentiometer 2 on index value 1.
  p2 = analogRead(PM[1]);

  // the hue of blue is the map of the second potentiometer
  // reading.
```

```
  h2 = map(p2, 0 ,1023, 0, 255);
}

// This function declates the reading of the third potentiometer
// and the hue of the green section of the RGB led.
int GreenSpectrum(){

  // The third potentiometer reading is the value of
  // potentiometer 3.
  p3 = analogRead(PM[2]);

  // the hue of green is the map of the third potentiometer
  // reading on indx value 2.
  h3 = map(p3,0, 1023, 0, 255);
}

// This function declares the mixture of all the hues of each
// colour in the RGB led.
int HuesMixture(){

  // displays the hue of red (index value 0) on the RGB led.
  analogWrite(RGB[0], h);

  // displays the hue of blue (index value 1) on the RGB led.
  analogWrite(RGB[1], h2);

  // displays the hue of green (index value 2) on the RGB led.
  analogWrite(RGB[2], h3);
}


void loop()
{

  // Calls in the function controlling red hues.
  RedSpectrum();

  // Calls in the function controlling blue hues.
  BlueSpectrum();

  // Calls in the function controlling green hues.
  GreenSpectrum();

  // Calls in function that mixes the hues together.
```

```
  HuesMixture();

}
```