```
//All the Codes Part B-G (excluding F)
//Part B:
//counter for if statements used in ISR function
int flagman = 0;
//setting up the LED segments and pins
const int ledseg1 = 12;
const int ledseg2 = 11; //pin 2 is the button, so seg 2 is pin 11
const int ledseg3 = 3;
const int ledseg4 = 4;
const int ledseg5 = 5;
const int ledseg6 = 6;
const int ledseg7 = 7;
const int ledseg8 = 8;
const int ledseg9 = 9;
const int ledseg10 = 10;

void setup(void) {
  pinMode(2, INPUT); //sets the button pin to input
  sei(); //Enables global interrupts;
  EIMSK |= (1 << INT0); //Enables External Interrupt INT0, which is the button
  EICRA |= (1 << ISC01); // (is falling edge mode, this triggers INT) on falling edge mode.

  //sets Led segments to be output to
  pinMode(ledseg1, OUTPUT);
  pinMode(ledseg2, OUTPUT);
  pinMode(ledseg3, OUTPUT);
  pinMode(ledseg4, OUTPUT);
  pinMode(ledseg5, OUTPUT);
  pinMode(ledseg6, OUTPUT);
  pinMode(ledseg7, OUTPUT);
  pinMode(ledseg8, OUTPUT);
  pinMode(ledseg9, OUTPUT);
  pinMode(ledseg10, OUTPUT);
}

//loop
void loop() {
} //Function that runs switch statements activated in the ISR Loop

void next_number(int button_click)
{
  switch (button_click)
  {
    //number 9
    case 0:
    digitalWrite(ledseg1, LOW);
    digitalWrite(ledseg2, LOW);
    digitalWrite(ledseg3, LOW);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, HIGH);
    digitalWrite(ledseg10, HIGH);
    break;
    //number 8
```

```
case 1:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 7
case 2:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, LOW);
break;
//number 6
case 3:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, LOW);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 5
case 4:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, LOW);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 4
case 5:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 3
case 6:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 2
case 7:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, LOW);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 1
case 8:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, LOW);
break;
//number 0
case 9:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, HIGH);
```

```
        digitalWrite(ledseg10, LOW);
        break;
    }
}

ISR(INT0_vect) //Interrupt service routine, uses a button
{
    if (flagman > 9) // if statement to reset flagman anytime flagman is greater than 9, this creates a loop for the switch statement.
    {
        flagman = 0; //resets flagman when flagman exceeds 9, this causes a loop
    }

    next_number(flagman); //this calls the switch statement function and uses flagman as a counter to cycle through cases.
    flagman++; //increments flagman
}

//Part C:
//counter for if statement used in ISR function
int flagman = 0;
//setting up the LED segments and pins
const int ledseg1 = 12;
const int ledseg2 = 11; //pin 2 is the button, so seg 2 is pin 11
const int ledseg3 = 3;
const int ledseg4 = 4;
const int ledseg5 = 5;
const int ledseg6 = 6;
const int ledseg7 = 7;
const int ledseg8 = 8;
const int ledseg9 = 9;
const int ledseg10 = 10;

//Setup
void setup(void) {
    pinMode(A0, INPUT);
    sei(); //Enables global interrupts;
    PCICR |= (1<<PCIE1); //Pin Change Interrupt Control Register 1
    PCMSK1 |= (1<<PCINT8); //pin change mask 1 (mask = enable), pc interrupt 8 for A0

    //setting LED segments to be output to
    pinMode(ledseg1, OUTPUT);
    pinMode(ledseg2, OUTPUT);
    pinMode(ledseg3, OUTPUT);
    pinMode(ledseg4, OUTPUT);
    pinMode(ledseg5, OUTPUT);
    pinMode(ledseg6, OUTPUT);
    pinMode(ledseg7, OUTPUT);
    pinMode(ledseg8, OUTPUT);
    pinMode(ledseg9, OUTPUT);
    pinMode(ledseg10, OUTPUT);
}

//loop
void loop() {
}

//Function that runs switch statements activated in the ISR Loop
void next_number(int button_click)
```

```
{
    switch (button_click)
    {
        //number 9
        case 0:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 8
        case 1:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 7
        case 2:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, LOW);
        break;
        //number 6
        case 3:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, LOW);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 5
        case 4:
```

```
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, LOW);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 4
case 5:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 3
case 6:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 2
case 7:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, LOW);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 1
case 8:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
```

```
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, LOW);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, LOW);
        break;
        //number 0
        case 9:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, LOW);
        break;
    }
}

ISR(PCINT1_vect) //This ISR utilizes any change in the pins to change numbers, thus when you press the button in, it will increment, and when you release the button it will increment.
{
    if (flagman > 9) //this if statement creates a loop when flagman exceeds 9
    {
        flagman = 0; //if the counter "flagman" exceeds 9 the counter resets to create a loop
    }

    next_number(flagman); // this calls the function with the case statements using flagman to cycle through cases.
    flagman++; // this function increments flagman, which is the counter.
}

//Part D:
//counters for if statements used in ISR functions
int flagman = 0;
//setting up the LED segments and pins
const int ledseg1 = 12;
const int ledseg2 = 11; //pin 2 is the button, so seg 2 is pin 11
const int ledseg3 = 3;
const int ledseg4 = 4;
const int ledseg5 = 5;
const int ledseg6 = 6;
const int ledseg7 = 7;
const int ledseg8 = 8;
const int ledseg9 = 9;
const int ledseg10 = 10;

//setup
void setup(void) {
    cli(); //Disable global interrupts
    TCCR1A = 0; //timer counter control at 0
    TCCR1B = 0; //timer counter control at 0
    TCNT1 = 34286; //preload timer (65536-16MHz/256/2Hz)
    TCCR1B |= (1<<CS12); //(originally a 256 prescaler, but it is mixed with the next prescaler now)
    TCCR1B |= (1<<CS10); //timer counter control 1, 0, this mixed with cs12, create an input clk i/o at 1024 (prescaler),
    TIMSK1 |= (1<<TOIE1); //Enable timer overflow interrupt
    sei(); //Enable Global Interrupts

    //setting LED segments to be output to
    pinMode(ledseg1, OUTPUT);
    pinMode(ledseg2, OUTPUT);
    pinMode(ledseg3, OUTPUT);
    pinMode(ledseg4, OUTPUT);
    pinMode(ledseg5, OUTPUT);
    pinMode(ledseg6, OUTPUT);
    pinMode(ledseg7, OUTPUT);
    pinMode(ledseg8, OUTPUT);
    pinMode(ledseg9, OUTPUT);
    pinMode(ledseg10, OUTPUT);
}

//loop
void loop() {
}

//Function that runs switch statements activated in the ISR Loops
void next_number(int button_click)
{
    switch (button_click)
    {
        //number 9
        case 0:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 8
        case 1:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 7
        case 2:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, LOW);
        break;
        //number 6
        case 3:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, LOW);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 5
        case 4:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, LOW);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 4
        case 5:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, LOW);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 3
        case 6:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, LOW);
```

```
        digitalWrite(ledseg10, HIGH);
        break;
        //number 2
        case 7:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, LOW);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 1
        case 8:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, LOW);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, LOW);
        break;
        //number 0
        case 9:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, LOW);
        break;
    }
}

ISR(TIMER1_OVF_vect) //Runs the ISR based on a clock. In this case the clock waits for 2 seconds on each number
{
    TCNT1 = 34286;
    if (flagman > 9) //this if statement creates a loop when flagman exceeds 9

    {
        flagman = 0; //if the counter "flagman" exceeds 9 the counter resets to create a loop
    }

    next_number(flagman); // this calls the function with the case statements using flagman to cycle through cases.
    flagman++; // this function increments flagman, which is the counter.
}
```

```
//Part E:
//counters for if statement used in ISR function
int flagman = 0;
//setting up the LED segments and pins
const int ledseg1 = 12;
const int ledseg2 = 11; //pin 2 is the button, so seg 2 is pin 11
const int ledseg3 = 3;
const int ledseg4 = 4;
const int ledseg5 = 5;
const int ledseg6 = 6;
const int ledseg7 = 7;
const int ledseg8 = 8;
const int ledseg9 = 9;
const int ledseg10 = 10;

//Setup
void setup(void) {
    TCCR1A = 0; //Timer counter control
    TCCR1B = 0; //Timer counter control
    TCNT1 = 0; //preload timer (65536-16MHz/256/2Hz)
    OCR1A = 31250; //makes the timer run at about 2 second between switching case statements
    TCCR1B |= (1<<WGM12); //timer counter control
    TCCR1B |= (1<<CS12); //prescaler at 256, combos with the cs10 to make 2 second delay
    TCCR1B |= (1<<CS10); //timer counter control 1, 0, this mixed with cs12, create an input clk i/o at 1024
    TIMSK1 |= (1<<OCIE1A); //Enable Timer Overflow Interrupt
    sei(); //Enables Interrupts

    //setting LED segments to be output to
    pinMode(ledseg1, OUTPUT);
    pinMode(ledseg2, OUTPUT);
    pinMode(ledseg3, OUTPUT);
    pinMode(ledseg4, OUTPUT);
    pinMode(ledseg5, OUTPUT);
    pinMode(ledseg6, OUTPUT);
    pinMode(ledseg7, OUTPUT);
    pinMode(ledseg8, OUTPUT);
    pinMode(ledseg9, OUTPUT);
    pinMode(ledseg10, OUTPUT);
}
//loop

void loop() {
}

//Function that runs switch statements activated in the ISR Loop
void next_number(int button_click)
{
    switch (button_click)
    {
        //number 9
        case 0:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 8
        case 1:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 7
        case 2:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, LOW);
        digitalWrite(ledseg3, LOW);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, HIGH);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, LOW);
        digitalWrite(ledseg10, LOW);
        break;
        //number 6
        case 3:
        digitalWrite(ledseg1, HIGH);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, LOW);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, HIGH);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
        //number 5
        case 4:
        digitalWrite(ledseg1, LOW);
        digitalWrite(ledseg2, HIGH);
        digitalWrite(ledseg3, HIGH);
        digitalWrite(ledseg4, HIGH);
        digitalWrite(ledseg5, LOW);
        digitalWrite(ledseg6, LOW);
        digitalWrite(ledseg7, HIGH);
        digitalWrite(ledseg8, LOW);
        digitalWrite(ledseg9, HIGH);
        digitalWrite(ledseg10, HIGH);
        break;
```

```
//number 4
case 5:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 3
case 6:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 2
case 7:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, LOW);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 1
case 8:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, LOW);
break;
//number 0
case 9:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
```

```
      digitalWrite(ledseg4, HIGH);
      digitalWrite(ledseg5, LOW);
      digitalWrite(ledseg6, HIGH);
      digitalWrite(ledseg7, HIGH);
      digitalWrite(ledseg8, HIGH);
      digitalWrite(ledseg9, HIGH);
      digitalWrite(ledseg10, LOW);
      break;
   }
}

ISR(TIMER1_COMPA_vect) //Interrupt service routine, clear timer on compare match
{
   if (flagman > 9) //this if statement creates a loop when flagman exceeds 9
   {
      flagman = 0; //if the counter "flagman" exceeds 9 the counter resets to create a loop
   }

   next_number(flagman); // this calls the function with the case statements using flagman to cycle through cases.
   flagman++; // this function increments flagman, which is the counter.
}

//Part G:
//counters for if statements used in ISR functions
int flagman = 0;
int flagman_2 = 0;
//setting up the LED segments and pins
const int ledseg1 = 12;
const int ledseg2 = 11; //pin 2 is the button, so seg 2 is pin 11
const int ledseg3 = 3;
const int ledseg4 = 4;
const int ledseg5 = 5;
const int ledseg6 = 6;
const int ledseg7 = 7;
const int ledseg8 = 8;
const int ledseg9 = 9;
const int ledseg10 = 10;

//Setup
void setup(void) {
   pinMode(2, INPUT); //sets the button pin 2 input
   sei(); //Enables global interrupts;
   EIMSK |= (1 << INT0); //Enables External Interrupt INT0, which is the button
   EICRA |= (1 << ISC01); // (is falling edge mode, this triggers INT0) on falling edge mode.
   cli(); //disables interrupts
   TCCR1A = 0; //timer counter control
   TCCR1B = 0; //timer counter control
   TCNT1 = 0; //preload timer (65536-16MHz/256/2Hz)
   OCR1A = 62500; //makes the timer run at about 1 second between switching case statements
   TCCR1B |= (1<<WGM12); //timer counter control
   TCCR1B |= (1<<CS12); //timer counter control, 256 prescaler originally, but it's mixed now to provide 1 second delay.
   TIMSK1 |= (1<<OCIE1A); //Enable timer overflow interrupt
   sei(); //enables interrupts

   //setting LED segments to be output to
   pinMode(ledseg1, OUTPUT);
   pinMode(ledseg2, OUTPUT);
```

```
    pinMode(ledseg3, OUTPUT);
    pinMode(ledseg4, OUTPUT);
    pinMode(ledseg5, OUTPUT);
    pinMode(ledseg6, OUTPUT);
    pinMode(ledseg7, OUTPUT);
    pinMode(ledseg8, OUTPUT);
    pinMode(ledseg9, OUTPUT);
    pinMode(ledseg10, OUTPUT);
}

//loop
void loop() {
} //Function that runs switch statements activated in the ISR Loops

void next_number(int button_click)
{
    switch (button_click)
    {
    //number 9
    case 0:
    digitalWrite(ledseg1, LOW);
    digitalWrite(ledseg2, LOW);
    digitalWrite(ledseg3, LOW);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, HIGH);
    digitalWrite(ledseg10, HIGH);
    break;
    //number 8
    case 1:
    digitalWrite(ledseg1, HIGH);
    digitalWrite(ledseg2, HIGH);
    digitalWrite(ledseg3, HIGH);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, HIGH);
    digitalWrite(ledseg10, HIGH);
    break;
    //number 7
    case 2:
    digitalWrite(ledseg1, LOW);
    digitalWrite(ledseg2, LOW);
    digitalWrite(ledseg3, LOW);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, LOW);
    digitalWrite(ledseg10, LOW);
    break;
```

```
//number 6
case 3:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, LOW);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, HIGH);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 5
case 4:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, LOW);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 4
case 5:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, LOW);
digitalWrite(ledseg3, HIGH);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, LOW);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, HIGH);
digitalWrite(ledseg10, HIGH);
break;
//number 3
case 6:
digitalWrite(ledseg1, LOW);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, LOW);
digitalWrite(ledseg4, HIGH);
digitalWrite(ledseg5, LOW);
digitalWrite(ledseg6, HIGH);
digitalWrite(ledseg7, HIGH);
digitalWrite(ledseg8, LOW);
digitalWrite(ledseg9, LOW);
digitalWrite(ledseg10, HIGH);
break;
//number 2
case 7:
digitalWrite(ledseg1, HIGH);
digitalWrite(ledseg2, HIGH);
digitalWrite(ledseg3, HIGH);
```

```
    digitalWrite(ledseg4, LOW);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, LOW);
    digitalWrite(ledseg10, HIGH);
    break;
    //number 1
    case 8:
    digitalWrite(ledseg1, LOW);
    digitalWrite(ledseg2, LOW);
    digitalWrite(ledseg3, LOW);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, LOW);
    digitalWrite(ledseg8, LOW);
    digitalWrite(ledseg9, LOW);
    digitalWrite(ledseg10, LOW);
    break;
    //number 0
    case 9:
    digitalWrite(ledseg1, HIGH);
    digitalWrite(ledseg2, HIGH);
    digitalWrite(ledseg3, HIGH);
    digitalWrite(ledseg4, HIGH);
    digitalWrite(ledseg5, LOW);
    digitalWrite(ledseg6, HIGH);
    digitalWrite(ledseg7, HIGH);
    digitalWrite(ledseg8, HIGH);
    digitalWrite(ledseg9, HIGH);
    digitalWrite(ledseg10, LOW);
    break;
    }
}

ISR(TIMER1_COMPA_vect) //This Interrupt Service routine utilizes concepts of Timer Overflow and a button to keep activate after each loop
{
    if(flagman > 9) //this if statement creates a loop when flagman exceeds 9
    {
        flagman = 0; //if the counter "flagman" exceeds 9 the counter resets to create a loop
        flagman_2 = 1; //after the first loop of cases a second counter is incremented, this counter pauses things until it is reset in the second ISR function.
    }

    if (flagman_2 == 1) //this if statement effectively keeps resetting flagman to 0 when the second counter is 1. The 2nd ISR statement frees flagman
from being 0 by making flagman two low.
    {
        flagman = 0; //this blocks flagman from changing out of number 9 LED state, until flagman_2 becomes 0 again.
    }

    next_number(flagman); // this calls the function with the case statements using flagman to cycle through cases.
    flagman++; // this function increments flagman, which is the counter.
}

ISR(INT0_vect) //Interrupt service routine, uses a button
{
    flagman_2 = 0; //This statement frees flagman_2 so that the first ISR can perform another cycle.
}
```