Introduction:

Learning how to C Programming in Atmel Studio for the AVR Microcontroller.

Equipment:

- Simon Board
- Atmel Studio 7

Procedure:

- Open Atmel Studio 7 and create a new assembly project.
- Set PortA to input mode and PortD to output mode.
- Write if conditions to detect if Pins are being pressed.
- If one of the Pins are pressed compliment the pin.
- After complimenting the pin, output the result to PortD.
- Delay the loop for 1 second and repeat.
- In order to do multiplication, addition is repeated multiple times.
- First PortA, and PortD are set to be inputs and outputs.
- By adding three to the register five times, the result will be equal to 3*5.
- Map the sum to PortD, the LEDs light up to represent 15.
- For part 2, use the first three buttons, and the last three buttons to then multiply and map the result to the LEDs.
- Use the input pins to determine what is being added and how many times.
- Since the LEDs for each pin in PortD are flipped, reverse the bits.
- Check if each bit was active then map it to the corresponding LED.
- Output the bits to the PortD to the LEDs.

Results:

The two 3 bit numbers were multiplied as expected by the written code and the push buttons lit up the LEDs as expected.

Questions:

1. The frequency must be known to properly calculate the time delay needed.
2. A for loop set to a certain duration to act as a delay loop.
3. The C code was easier to put together because many of the steps were easier to understand. The C code was larger because the assembly code is more efficient.

Conclusion:

I learned how to create a project in Atmel Studio 7 to multiply two 3 bit numbers in embedded C. I also learned how to reverse bits to correctly output the bits to the LEDs using push buttons in C.

Appendices:

```
/*
 * toggleProject.c
 *
 * Created: 2/26/2020 10:02:43 AM
 * Author : argk4
 */

#include <avr/io.h>
#define F_CPU 16000000UL
#include "util/delay.h"
#define SW1 PINA0
#define SW2 PINA1
#define SW3 PINA2
#define SW4 PINA3

int main(void)
{
    DDRD = 0xFF;
       DDRA = 0x00;
       PORTA = 0xFF;
       unsigned char temp = 0x00;
       unsigned char num1 = 0x00;
       unsigned char num2 = 0x00;

    while (1)
    {
              PORTD = 0xFF;
              if ( ( ~PINA & ( 1 << SW1 ) ) )
              {
                     num1 += 0b00000001;
              }
              if ( ( ~PINA & ( 1 << SW2 ) ) )
              {
                     num1 += 0b00000010;
              }
              if ( ( ~PINA & ( 1 << SW3 ) ) )
              {
                     num1 += 0b00000100;
              }
              if ( ( ~PINA & ( 1 << SW4 ) ) )
              {
                     num1 += 0b00001000;
              }
              PORTD = ~temp;
              _delay_ms(1000);
    }
}

/*
```

```
 * _3bitMult.c
 *
 * Created: 2/26/2020 11:21:18 AM
 *  Author: argk4
 */

#include <avr/io.h>
#define F_CPU 16000000UL
#include "util/delay.h"

int main(void)
{
	DDRD = 0xFF;
	PORTD = 0xFF;
	DDRA = 0x00;
	PORTA = 0xFF;
	unsigned char temp = 0x00;
	unsigned char num1 = 0x00;
	unsigned char num2 = 0x00;

	while (1)
	{
		temp = ~PINA;
		temp = (temp & 0b11110000) >> 4 | (temp & 0b00001111) << 4;
		temp = (temp & 0b11001100) >> 2 | (temp & 0b00110011) << 2;
		temp = (temp & 0b10101010) >> 1 | (temp & 0b01010101) << 1;
		num1 = (temp & 0b00000111);
		num2 = (temp & 0b11100000) >> 5;
		temp = 0;

		for( ;num2 > 0; num2-- )
		{
			temp += num1;
		}

		temp = (temp & 0b11110000) >> 4 | (temp & 0b00001111) << 4;
		temp = (temp & 0b11001100) >> 2 | (temp & 0b00110011) << 2;
		temp = (temp & 0b10101010) >> 1 | (temp & 0b01010101) << 1;
		num1 = (temp & 0b00000111);
		num2 = (temp & 0b11100000) >> 5;
		temp = 0;

		for( ;num2 > 0; num2-- )
		{
			temp += num1;
		}

		PORTD = ~temp;
		_delay_ms(1000);
	}
}
```