```
/*
 * adeja001_lab1_challenge.c − April 8, 2013
 * Name: Ariana DeJaco  E−mail:adeja001@ucr.edu
 * CS Login: adeja001
 * Partner Name: Joshua DeForest−Williams jdefo002@ucr.edu
 * Lab Section: 022
 * Assignment: Lab#1 Exercise# Challenge
 * Exercise Description: Challenge problem. LSB used for if weight > 140. Second LSB used to check if difference between A and C is > 80.
 *                                        6 MSB are used to show weight. Accurate except for 3kg b/c the first two bits are already used.
 */

#include <avr/io.h>
#include <avr/sfr_defs.h>

// DDR = Data Direction Register. Shows which bits are outputs and which are inputs. Inputs = 0. Outputs = 1.

// Current Port Definitions
#define LED_DDR                 DDRD
#define LED_INPORT              PIND
#define LED_OUTPORT             PORTD
#define SENSORA_DDR             DDRA
#define SENSORA_INPORT  PINA  // PINA is when you read
#define SENSORA_OUTPORT PORTA //PORTA is when you write
#define SENSORB_DDR             DDRB
#define SENSORB_INPORT  PINB
#define SENSORB_OUTPORT PORTB
#define SENSORC_DDR             DDRC
#define SENSORC_INPORT  PINC
#define SENSORC_OUTPORT PORTC

// Additional macros not defined in sfr_defs.h
#define SET_PORT_BIT(OUTPORT, BIT)              OUTPORT |= (1 << BIT)
#define CLEAR_PORT_BIT(OUTPORT, BIT)    OUTPORT &= ~(1 << BIT)

int main(void)
{
        SENSORA_DDR = 0x00;
        SENSORB_DDR = 0x00;
        SENSORC_DDR = 0x00;
        LED_DDR = 0xFF;

        unsigned long totalWeight;
        unsigned char ACDifWeight;

        while(1)
        {
                totalWeight = 0;
                ACDifWeight = 0;

                totalWeight += PINA;
                totalWeight += PINB;
                totalWeight += PINC;

                if(PINA > PINC)
                {
                        ACDifWeight = PINA − PINC;
                }
                else
                {
                        ACDifWeight = PINC − PINA;
                }

                if(totalWeight > 140)
                {
                        SET_PORT_BIT(LED_OUTPORT, 0);
                }
                if(ACDifWeight > 80)
                {
                        SET_PORT_BIT(LED_OUTPORT, 1);
                }
```



```

                while(totalWeight > 3)
                    {
                            if(totalWeight >= 128)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 7);
                                    totalWeight −= 128;
                            }
                            else if(totalWeight >= 64)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 6);
                                    totalWeight −= 64;
                            }
                            else if(totalWeight >= 32)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 5);
                                    totalWeight −= 32;
                            }
                            else if(totalWeight >= 16)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 4);
                                    totalWeight −= 16;
                            }
                            else if(totalWeight >= 8)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 3);
                                    totalWeight −= 8;
                            }
                            else if(totalWeight >= 4)
                            {
                                    SET_PORT_BIT(LED_OUTPORT, 2);
                                    totalWeight −= 4;
                            }
                    }
            }

}
```

```
/*
 * adeja001_lab1_part1.c - April 8, 2013
 * Name: Ariana DeJaco  E-mail:adeja001@ucr.edu
 * CS Login: adeja001
 * Partner Name: Joshua DeForest-Williams jdefo002@ucr.edu
 * Lab Section: 022
 * Assignment: Lab#1 Exercise# Part 1
 * Exercise Description: Program that illuminates an LED connected to PB0 if the
 garage door is open at night.
 */


#include <avr/io.h>
#include <avr/sfr_defs.h>

#define IS_GARAGE_DOOR_SET  (bit_is_set(PINA, PINA0))
#define IS_LIGHT_SENSOR_SET (bit_is_set(PINA, PINA1))
#define SET_LED          PORTB |= (1 << PINB0)
#define CLEAR_LED        PORTB &= ~(1 << PINB0)

int main(void)
{
        DDRA = 0x00;
        DDRB = 0x01;
        PORTB = 0x00;

    while(1)
    {
                if(!(IS_LIGHT_SENSOR_SET) && IS_GARAGE_DOOR_SET)
                {
                  SET_LED;
                }
                else
                {
                  CLEAR_LED;
                }
    }
}
```

```
/*
 * adeja001_lab1_part2.c − April 8, 2013
 * Name: Ariana DeJaco  E−mail:adeja001@ucr.edu
 * CS Login: adeja001
 * Partner Name: Joshua DeForest−Williams jdefo002@ucr.edu
 * Lab Section: 022
 * Assignment: Lab#1 Exercise# Part2
 * Exercise Description: Program outputs in binary on port C the number of avail
able spaces.
 */


#include <avr/io.h>
#include <avr/sfr_defs.h>

// Current Port Definitions
#define LED_DDR                 DDRC
#define LED_INPORT              PINC
#define LED_OUTPORT             PORTC
#define SENSOR_DDR              DDRA
#define SENSOR_INPORT   PINA
#define SENSOR_OUTPORT  PORTA

// Additional macros not defines in sfr_defs.h
#define SET_PORT_BIT(OUTPORT, BIT)              OUTPORT |= (1 << BIT)
#define CLEAR_PORT_BIT(OUTPORT, BIT)    OUTPORT &= ~(1 << BIT)

int main(void)
{
        SENSOR_DDR = 0x00;
        LED_DDR = 0xFF;

        char freespaces;

    while(1)
    {
                freespaces = 0;
                for(char i = 0; i < 4; i++)
                {
                        if(!(bit_is_set(SENSOR_INPORT, i)))
                        {
                                freespaces++;
                        }
                }

                LED_OUTPORT = freespaces;
    }
}
```

```c
/*
 * adeja001_lab1_part3.c − April 8, 2013
 * Name: Ariana DeJaco  E−mail:adeja001@ucr.edu
 * CS Login: adeja001
 * Partner Name: Joshua DeForest−Williams jdefo002@ucr.edu
 * Lab Section: 022
 * Assignment: Lab#1 Exercise# Part3
 * Exercise Description: Extension of the second part of this lab that writes the available spaces number to only PC3..PC0,
 *                                          and to set PC7 to 1 if the lot is full.
 */

#include <avr/io.h>
#include <avr/sfr_defs.h>

// DDR = Data Direction Register. Shows which bits are outputs and which are inputs. Inputs = 0. Outputs = 1.

// Current Port Definitions
#define LED_DDR                 DDRC
#define LED_INPORT              PINC
#define LED_OUTPORT             PORTC
#define SENSOR_DDR              DDRA
#define SENSOR_INPORT   PINA
#define SENSOR_OUTPORT  PORTA

// Additional macros not defines in sfr_defs.h
#define SET_PORT_BIT(OUTPORT, BIT)              OUTPORT |= (1 << BIT)
#define CLEAR_PORT_BIT(OUTPORT, BIT)    OUTPORT &= ~(1 << BIT)

int main(void)
{
        SENSOR_DDR = 0x00;
        LED_DDR = 0x8F;

        char freespaces;

        while(1)
        {
                freespaces = 0;
                for(char i = 0; i < 4; i++)
                {
                        if(!(bit_is_set(SENSOR_INPORT, i)))
                        {
                                freespaces++;
                        }
                }

                if(freespaces == 0)
                {
                        LED_OUTPORT = 0x80;
                }
                else
                {
                        LED_OUTPORT = freespaces;
                }
        }
}
```