Works properly in simulation only: 2  
Bonus assignment done: +1

2-1 Binary numbers  
Write a program that displays binary numbers (8 bits). Digits should be set according to state of the  
switches (SW0-SW3, reorder SW2 to its proper position). The lower 4 bits are updated on BT0  
press, the upper 4 bits are updated on BT1 press. The digits of the binary number are displayed on  
the LEDs after pressing BT1.

2-2 Button press / button bouncing counter  
Write an assembly program for counting the number INT button presses. The INT button should be  
handled by interrupt. The value of the button counter should be displayed on the LEDs.  
Bonus: interrupt counter should stop at 0b1111 and reset to 0 on BT0 press.

2-3 Blinking LEDs  
Write a program that creates a 1 Hz blinking effect on a selected LED. The effect should be  
stopped as long as BT0 is pressed.  
You may use the timer interrupt or a software delay loop,  
Bonus: effect should stop/start (alternating) on each BT0 press.

2-4 Adder  
Create a program that sums the binary numbers set on the switches (SW0-SW3, reorder SW2 to its  
proper position). Input is 4 bit unsigned, result is accumulated as an 8 bit unsigned value. On each  
INT button press the value set on the switches is added to the sum and displayed on LEDs. BT0  
resets the sum to 0 and clears the LED output as well.  
The value should be added once per button press. Protect your program against long button presses!  
Bonus: debounce the INT button!

2-5 Led stepper  
Write a program that lights LED0 initially. On each BT2 button press move it to the next LED  
(LED7 is followed by LED0 again). One button press (regardless of its length) should cause only  
one LED step, but debouncing is not required.  
Bonus: BT0 should move the LEDs in the opposite direction.

2-6 Dice roll simulator  
Create an assembly code that simulates a 6-sided dice roll. On each INT button press a random  
number between 1 and 6 should be displayed on the LEDs. (1 → only LED0, 2 → LED0&1, 6 → all  
LEDs from LED0 to LED5).  
Random numbers can be generated by e.g. a fast internal counter sampled right at the moment of  
the button press, as human interaction timing is unpredictable at that rate.

2-7 Car lights  
Create a car lights simulator based on the following specs:  
Red LEDs: brake lamps (LED0, 1, 4, 5)  
   Turned on as long as INT button (brake) is pressed.  
Yellow LEDs: turn signal (LED2, 6)  
   [SW1 = 0: off], [SW1 = 1: on, SW0 = direction: 0 for left, 1 for right]  
Green LEDs: headlamps (front lights) (LED 3, 7)  
   SW3 = 1: on  

Bonus: turn signals should blink at around 1 Hz.