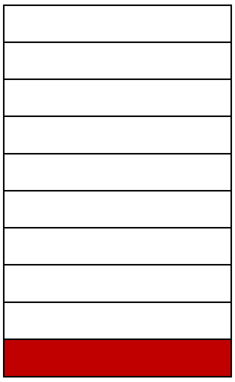
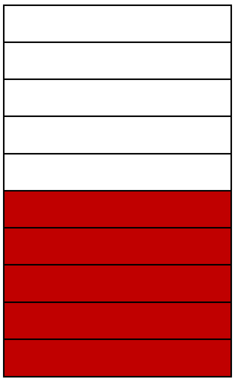
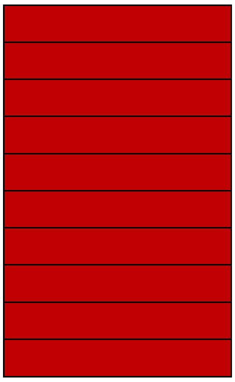
Lab 3

Instructions

- Complete each task and demonstrate the working program to your tutor. Tasks should be demonstrated using the board provided. You will have two lab sessions to work on this lab, and all questions must be marked by the end of the second session.
- Make reasonable assumptions if something is not explicitly specified. State all those assumptions to the tutor when you are getting marked.

Part A – lift floor display (2 mark)

Use the LED bar to show the current floor number of the lift. The current floor number can be assumed to be defined by a *".equ"* assembler directive as a constant in your source file. Assume there are only 10 floors in the building numbered 1-10. The LEDs should display a filling bar corresponding to the floor number as per the following examples.

.equ floor_number = 1	.equ floor_number = 5	.equ floor_number = 10
		

Part B – lift movement (3 marks)

Use the timer0 and the LED bar to demonstrate the movement of the lift. Modify part A such that the floor number is now stored in a register of your choice. The direction of the lift can also be stored in another register as 1 or 0. The lift takes 2 seconds to travel between floors. You must use timer0 to generate an interrupt that determines when the floor number and the LEDs are updated. In this exercise, the lift doesn't stop. instead, it can be assumed to move continuously up and down.

Part C – lift stopping (2 marks)

Extend part B such that the lift stops in requested floor numbers. The list of requested floor numbers is stored in the program memory using the *".db"* assembler directive. The lift should stop for 5 seconds on each requested floor. The LED bar should blink several times per second when the lift is not moving - while still displaying the current floor number.

Example: Current floor = 3 Current direction = up Requests = 4, 7, 9, 2, 1
Lift starts from 3rd floor moves upward until 9th floor while stopping at floors 4, 7 and 9. Then the lift moves downwards to 1st floor while stopping at floors 2 and 1.

Part D – door open and close buttons (3 marks)

Extend part C to simulate the door opening and closing buttons. Use the left push button as the door open button and the right push button as the door close button inside the lift. If the door open button is pressed (and then released instantly) when the lift is at the stop, the lift should stay on the current floor for 3 additional seconds. When the close button is pressed (and then released instantly), the lift should start moving immediately. Note that the open and close buttons are non-effective when the lift is moving. For this lab exercise, you do not need to handle the situation where a button is kept pressed continuously.

You must use falling-edge external interrupts 0 and 1 to detect the button pushes. The buttons must be software debounced so that one button press reliably generates only a single trigger. You should implement debouncing by ignoring spurious interrupts, **not by disabling interrupts or busy-waiting**.

If one press leads to two triggers even after debouncing is implemented, clear the INTn interrupt flag as the last step in the interrupt subroutine - by writing a logical one to its Interrupt Flag bit (INTFn) in the EIFR Register.