## Microprocessors Lab 6 Notes

**Part 1:**

1. All answered with the code (a, b, c, and d)

**Part 2:**

1. the counting works as expected. The delay is 1 second. The rollover (transition from 255 to 0) works as expected.
2. The board flashed 0x03 five times with intervals of 1 second and then returned to counting where it left off.
3. Same as 2) but with 0xC0.
4. Same as 2) but with 0xAA.
5. It finished INT2 and then did it again (flashed 10 times total). Then it went to regular counting.
6. It flashed 0xC0 five times and then 0x03 five times. Then it went back to regular counting. No, INT0 cannot interrupt INT1 because when you’re in an interrupt process, the global interrupt flag is disabled, so no other interrupt can happen until the process is complete.
7. Same thing as before. Flashes 3 five times and then AA five times once is done. The interrupts were done in the order thy were pressed.
8. It fist flashes 0xAA for INT2, then it flashes 0x03, because INT0 has the highest priority of the interrupts in line. Finally it flashes 0xC0, which has lower priority than INT0. (Order of priority: INT0>INT1>INT2).
9. Now, Switch 1 is disabled. Both INT0 and INT1 are connected to switch 0. Once the button is pressed, the bard will first flash 0x03 five times, and then flash 0xC0 five times.

**Part 3:**

Summary of how priority works.