University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 160 | Name: btn_hold_main.c
ATmega Lab 2 | Due: see http://ece160.org

Write a program to turn on a LED while the button is pressed. Switch A5 should control the Red LED, switch A4 the Green, and switch A3 the Blue. The corresponding LED should remain on as long as the button is held down. The LED should go off when the button is released.

By pressing combinations of buttons, you should be able to create 8 different colors on the LED: Black, Red, Green, Blue, Magenta, Cyan, Yellow, and White, based on the color table in Arduino Lab 1.

The buttons are connected to PORTC as follows:

Button	Port C bit	Value when pressed	Value when NOT pressed
A5 (leftmost)	c5	0	1
A4	c4	0	1
A3 (rightmost)	c3	0	1

Note that when the button is NOT pressed, the value of that bit on the port is 1. When the button is pressed, it is zero. Each of these bits on PORTC should be set to an input, by writing a 0 into bits 3,4,5 of DDRC. These bits of PORTC may then be read via PINC.

To insure the value of a pushbutton is 1, you need to enable the pull up resistor inside of the microcontroller. This will be covered in greater detail in ECE 263. For now, the code you should use to setup the buttons for input is as follows:

```
DDRC =  DDRC  & 0b11000111;  // set bits 5,4,3 as inputs
PORTC = PORTC | 0b00111000; // enable pull up resistors on bits 5,4,3
```

You may refer to ATmega Lab 1 for more info on DDRx, PORTx, and PINx. Also refer to ATmega Lab 1 (or your code for lab 1) for info on which bit of PORTB controls which color LED.

This must be checked of by TA, AND submitted to your M: drive. When submitting, INSURE that your first ATmega Lab is named led_main.c, and NOT just main.c. This lab must be named btn_hold_main.c

Student Name: ______________________________

TA Signature: ______________________________

Date/Time: ______________________________

CheckSum: ______________________________