```c
// OregonState EECS
// Microcontroller System Design
// lab1_code.c
// Joshua Reed
// Sep. 25, 2015

// This program represents S0 button presses as a BCD counter on the
// PORTB LEDs

#include <avr/io.h>
#include <util/delay.h>

//************************************************************************
*
//                              inc_as_bcd
// Increment 2 digit BCD counter and return a single integer representing 2 BCD
digits.
//************************************************************************
*
int8_t inc_as_bcd() {

    // digit one and two for BCD on PORTB LEDs
    static uint8_t dig1 = 0;
    static uint8_t dig2 = 0;

    if (dig1 < 9) { // increment dig1
        dig1 += 1;
        }
    else if (dig2 < 9) { // rollover dig1 // increment dig2
        dig1 = 0;
        dig2 += 1;
        }
    else { // rollover both digits
        dig1 = 0;
        dig2 = 0;
        }

    // return single integer representing the counter as BCD
    return (dig2<<4) | dig1;
    }



//************************************************************************
*
//                              debounce_switch
// Check pushbutton S0.
// Adapted from Roger's debounce function.
// Shift in a one when the button is depressed else a zero.
// Return a one once per debounced press.
// Expects an active low pushbutton on PORTD bit zero.
// Debounce time is determined by external loop delay times 4.
//************************************************************************
*
int8_t debounce_switch() {

    // button press shift register
    static uint8_t [8] SR = 0;
    uint8_t i = 0;
    for (i=0; i<8; i++) {
        // bit_is_clear() returns a one when button pushed
        SR[i]= (SR[i] << 1) | bit_is_clear(PIND, i);
        }

    if (SR[i] == 0x0F) { // if shift register = 00001111
        return i;
        }
    return 9;
    }
```

```c
//************************************************************************
*
//                              to_digs
//
//************************************************************************
*
uint8_t * to_digs(num) {
    static uint8_t [4] digs;
    digs[0] =  num      % 10;
    digs[1] = (num/10)  % 10;
    digs[2] = (num/100) % 10;
    digs[3] = (num/100);
    return digs;
    }

//************************************************************************
*
//                              main
// Check switch S0.
// If low for 4 passes of debounc_switch() increment BCD counter in inc_as_bcd()
.
// The BCD count is then displayed on PORTB's LEDs.
//************************************************************************
*
int main(){

    // set all of PORTB to output
    DDRB = 0xFF;

    while(1) { // loop forever
        if (debounce_switch()) { // if switch true for 4 passes, increment PORTB
            PORTB = inc_as_bcd();
            }
        // debounce 8ms
        _delay_ms(2);
        }
    }
```