```
// lab1_code.c
// Andrey Kornilovich
// 10.03.19

//This program increments a binary display of the number of button pushes on switch
//S0 on the mega128 board.

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

//***********************************************************************************
//                                debounce_switch
// Adapted from Ganssel's "Guide to Debouncing"
// Checks the state of pushbutton S0 It shifts in ones till the button is pushed.
// Function returns a 1 only once per debounced button push so a debounce and toggle
// function can be implemented at the same time.  Expects active low pushbutton on
// Port D bit zero.  Debounce time is determined by external loop delay times 12.
//***********************************************************************************
int8_t debounce_switch() {
  static uint16_t state = 0; //holds present state
  state = (state << 1) | (! bit_is_clear(PIND, 0)) | 0xE000;
  if (state == 0xF000) return 1;
  return 0;
}

//***********************************************************************************
// Check switch S0.  When found low for 12 passes of "debounce_switch(), increment
// PORTB.  This will make an incrementing count on the port B LEDS.
//***********************************************************************************
int main()
{
DDRB = 0xFF;  //set port B to all outputs

uint16_t x = 0;   //counter from 0 to 99

while(1){     //do forever
  if(debounce_switch()) {x++;}  //if switch true for 12 passes, increment x

  if(x > 99) {x = 0;}     //reset

  uint8_t MSB = x / 10;   //floor the count to get the MSB
  uint8_t LSB = x % 10;  //mod the count to get the LSB

  _delay_ms(2);          //keep in loop to debounce 24ms

  //shift the MSB by
  PORTB = (MSB << 4) + LSB;
  } //while
} //main
```