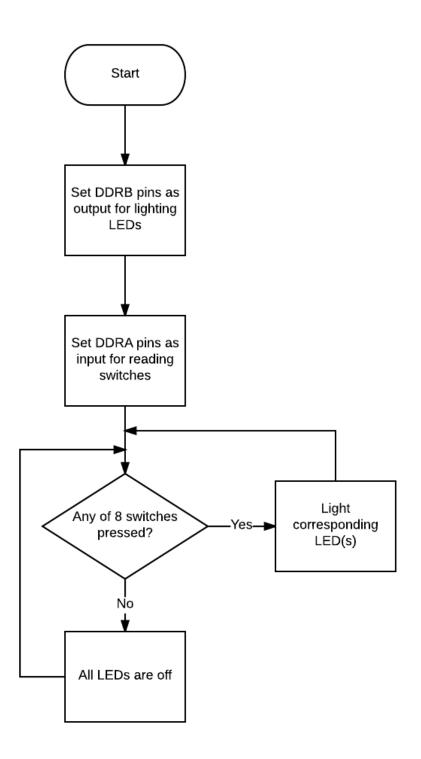
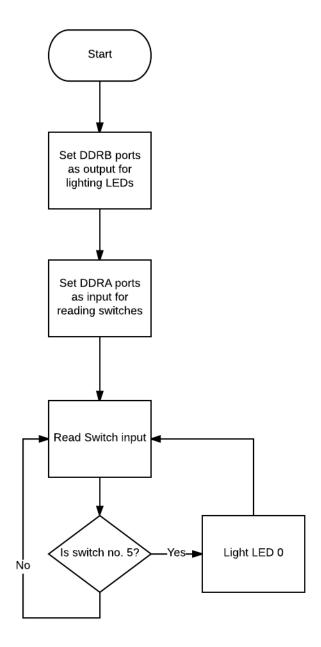
```
; 1DT301, Computer Technology I
; Lab number: 1
; Title: Task 1
; Hardware: STK600, CPU ATmega2560
; Function: To light LED 2 using PORTB as output
; Input ports: None
; Output ports: on-board LEDs connected to PORTB.
; Subroutines: If applicable.
ldi r16, 0 \times 04 ; put decimal value 4 in register
out 0 \times 04, r16
              ; output value of r16 to address 0x04 (DDRB)
      Start
   Assign bit 2 as
   output value of
   DDRB, to light
      LED 2
```

```
; 1DT301, Computer Technology I
; Lab number: 1
; Title: Task 2
; Hardware: STK600, CPU ATmega2560
; Function: To read the switches and light the corresponding LED \,
; Input ports: Switches connected to port A
; Output ports: Leds connected to port B.
;<<<<<<<<<<<<<<<<<
ldi r16, 0xFF
out DDRB, r16
                 ; set DDRB as output for lighting LEDs
ldi r16, 0x00
                    ; set DDRA as input for reading switches
out DDRA, r16
                     ; Keep a loop running indefinitely
loop:
                    ; Read switch info from PIN A
 in r15, PINA
 out PORTB, r15
                    ; Send this info to light LED on Port B.
rjmp loop
```



Task 3 + 4

```
; 1DT301, Computer Technology I
; Lab number: 1
; Title: Task 3
; Hardware: STK600, CPU ATmega2560
; Function: To make a program to light LEDO when Switch 5 is pressed.
; Input ports: Switches connected to port A
; Output ports: Leds connected to port B.
ldi r16, 0xFF
out DDRB, r16
                    ; set DDRB as output for lighting LEDs
ldi r16, 0x00
out DDRA, r16
                     ; set DDRA as input for reading switches switches
ldi r16, 0xFF
out portB, r16
1di \ r17, 0b110111111; values set in register to compare to in loop
ldi r18, 0b11111110
loop:
 in r16, PINA
                    ; Read switch info from PIN A
  cp r16, r17
                    ; Compare with r17, to check if SW5 is pressed
                    ; if so, branch to 'equal'
  breq equal
rjmp loop
```



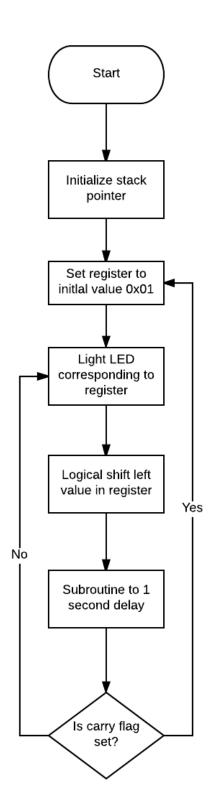
Notes regarding task 4: To light a LED, the bits representing the pins need to be inverted, because the pins are connected to ground on one side and will pull the port pin to ground (0) once they're pressed. This will result in a zero for a pressed button and a 1 for a released one. Once the port pin is low, it will sink current and the LED is lit.

Simplified: because of current and voltage specifications of the board, we have to invert the bits on the port pins to light the LED.

This was a little 'gotcha' while running the code in the simulator.

```
; 1DT301, Computer Technology I
; Lab number: 1
; Title: Task 5
; Hardware: STK600, CPU ATmega2560
; Function: Run a ring counter on the LEDS \,
; Input ports: None
; Output ports: Leds connected to port B.
; Subroutines: delay: accounts for 1 second delay between changing LED.
; Initialize SP, Stack Pointer
ldi r19, high(ramend)
out sph, r19
ldi r19, low(ramend)
out spl, r19
init:
  ldi r19, 0x01
loop:
  out DDRB, r19
                   ; logical shifts value of r19 to the left
  lsl r19
                  ; the delay is good for 16.000.000 cpu cycles,
  rcall delay
                   ; representing 1 second. (since cpu runs on 16Mhz)
  brcs init
                   ; branch to start at init again if carry flag is set
                   ; (resets LED to first bit in Port B).
rjmp loop
```

```
; 16MHz -> 16000000 cycles = 1s,
delay:
                  ; Cycles = 3a + 4ab + 3abc -> a(3 + b(4 + 3c))
  ldi r16, 11
                  ; -> a
delay_1:
 ldi r17, 237 ; -> b
delay_2:
  ldi r18, 255 ; -> c (a,b,c ~> 0.5s)
delay 3:
                ; decrements value of r18 until 0 because
  dec r18
                  ; it will branch to delay_3 again if not zero.
                 ; 3c - 1  -> d = 3c - 1
  brne delay 3
  dec r17
  nop
                  ; delay_2 resets r18 to 255 for 237 times because
                  ; r17 is decremented each time when delay_3 is zero.
  dec r16
                 ; the whole process above is repeated 11 times.
  brne delay 1 ; 5a - 1 + ae -> f = 3a + 5ab + 3abc - ab
  ret
                  ; f - 1
```



```
; 1DT301, Computer Technology I
; Lab number: 1
; Title: Task 6
; Hardware: STK600, CPU ATmega2560
; Function: To run a Johnson Counter
; Input ports: None
; Output ports: Leds connected to port B.
; Subroutines: delay: accounts for 1 second delay between changing LED.
;<<<<<<<<<<<<<<<<<<<><
; Initialize stack pointer
ldi r16, high(ramend)
out sph, r16
ldi r20, low(ramend)
out spl, r20
ldi r19, 0x00
ldi r20, 0x01
_add:
  out DDRB, r19
  add r19, r20
                  ; adds current value r19 with logical left shifted r20
  lsl r20
                   ; to keep lighting LEDs until bit 7.
                  ; have a delay of 1 second between lighting LEds.
  rcall delay
  cpi r19, 0xff
```

```
brne add ; run add until all LEDs are lit.
_sub:
  out DDRB, r19
  lsr r19
  rcall delay
  brcs _sub ; branch if carry flag is set in status register
  ldi r20, 0x01
  rjmp add
delay:
                   ; 16MHz -> 16000000 cycles = 1s,
                    ; Cycles = 3a + 4ab + 3abc -> a(3 + b(4 + 3c))
  ldi r16, 11
                    ; -> a
delay 1:
  ldi r17, 237 ; -> b
delay 2:
  ldi r18, 255 ; -> c (a,b,c ~> 0.5s)
delay 3:
  dec r18
                   ; decrements value of r18 until 0 because
                    ; it will branch to delay_3 again if not zero.
  brne delay 3
                   ; 3c - 1  -> d = 3c - 1
  dec r17
                    ; delay_2 resets r18 to 255 for 237 times because
  nop
                   ; r17 is decremented each time when delay 3 is zero.
  brne delay 2
                   ; 5b - 1 + bd -> e = 5b - 1 + 3cb - b
  dec r16
                   ; the whole process above is repeated 11 times.
  brne delay 1 ; 5a - 1 + ae -> f = 3a + 5ab + 3abc - ab
  ret
                    ; f - 1
```

