Computer Technology
(1DT301)

Lab 1:

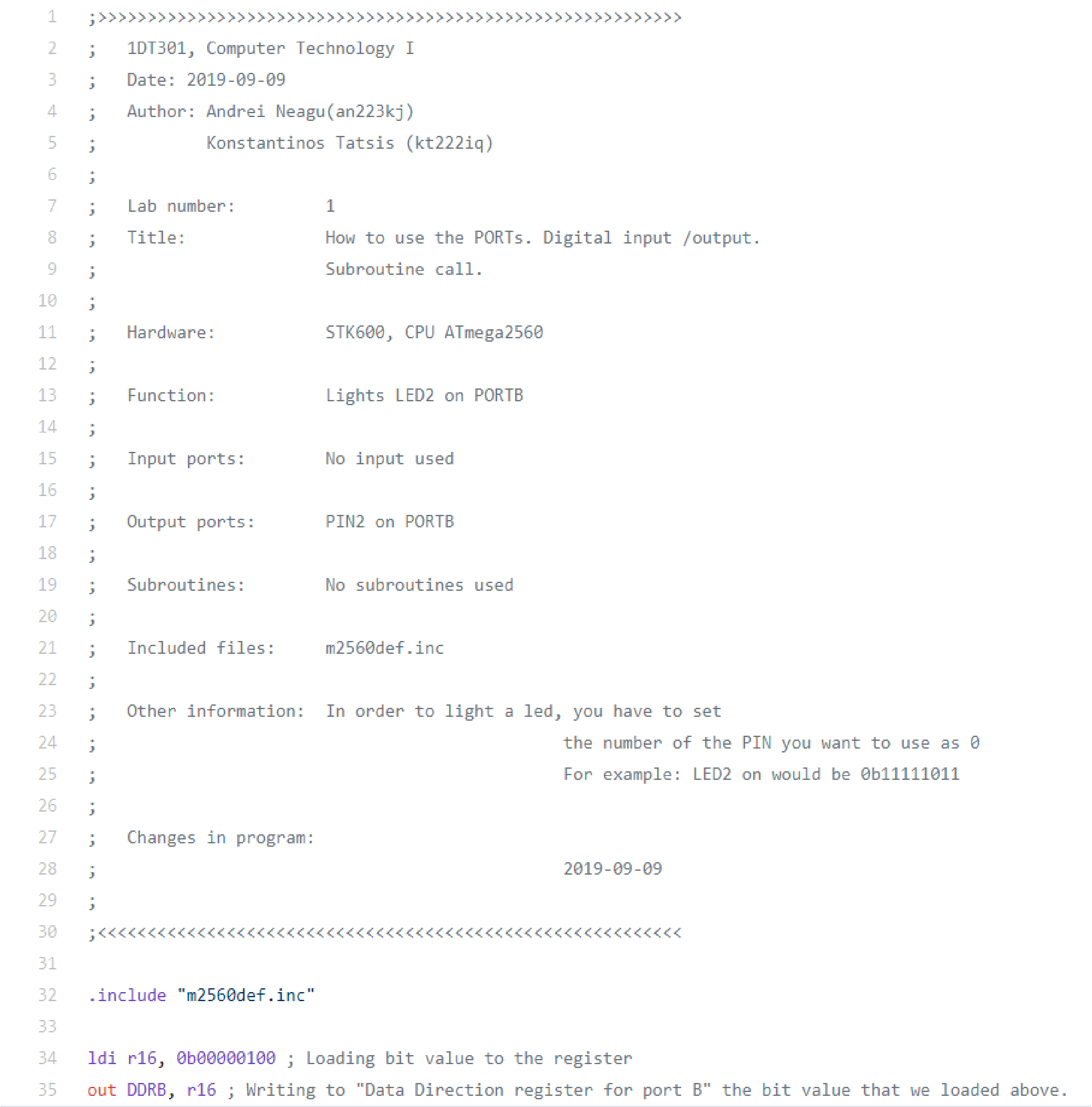
(How to use the PORTs, Digital input/output, Subroutine call)

Authors:
Andrei Neagu (AN223KJ)
Konstantinos Tatsis (KT222IQ)
GitHub: https://github.com/neaguandrei101/1DT30

Table of Contents

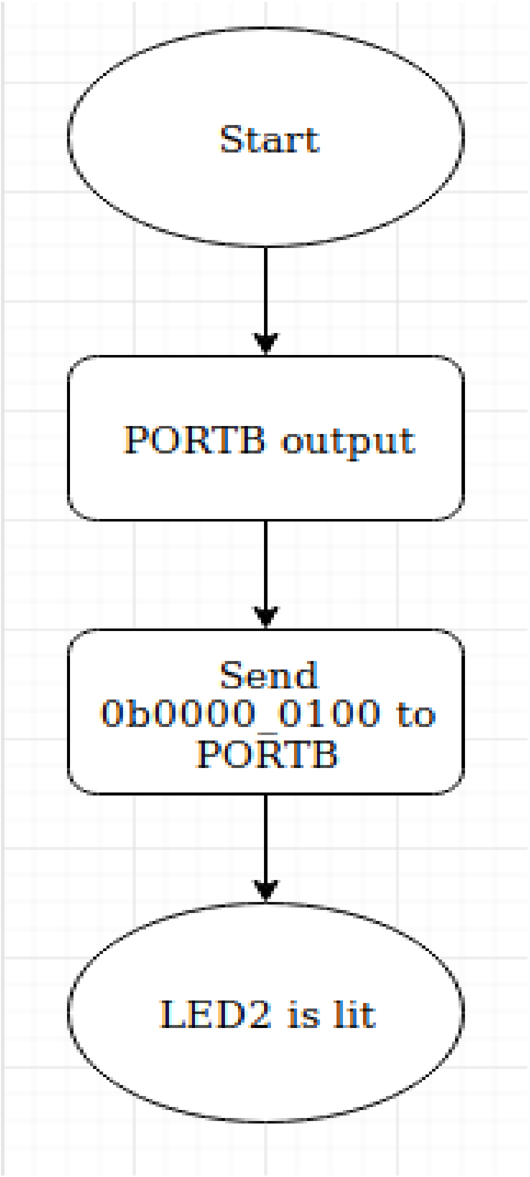
Task 1:

Write a program in Assembly language to light LED 2. You can use any of the four ports, but start with PORTB. The program should be very short! How many instructions is minimum number?



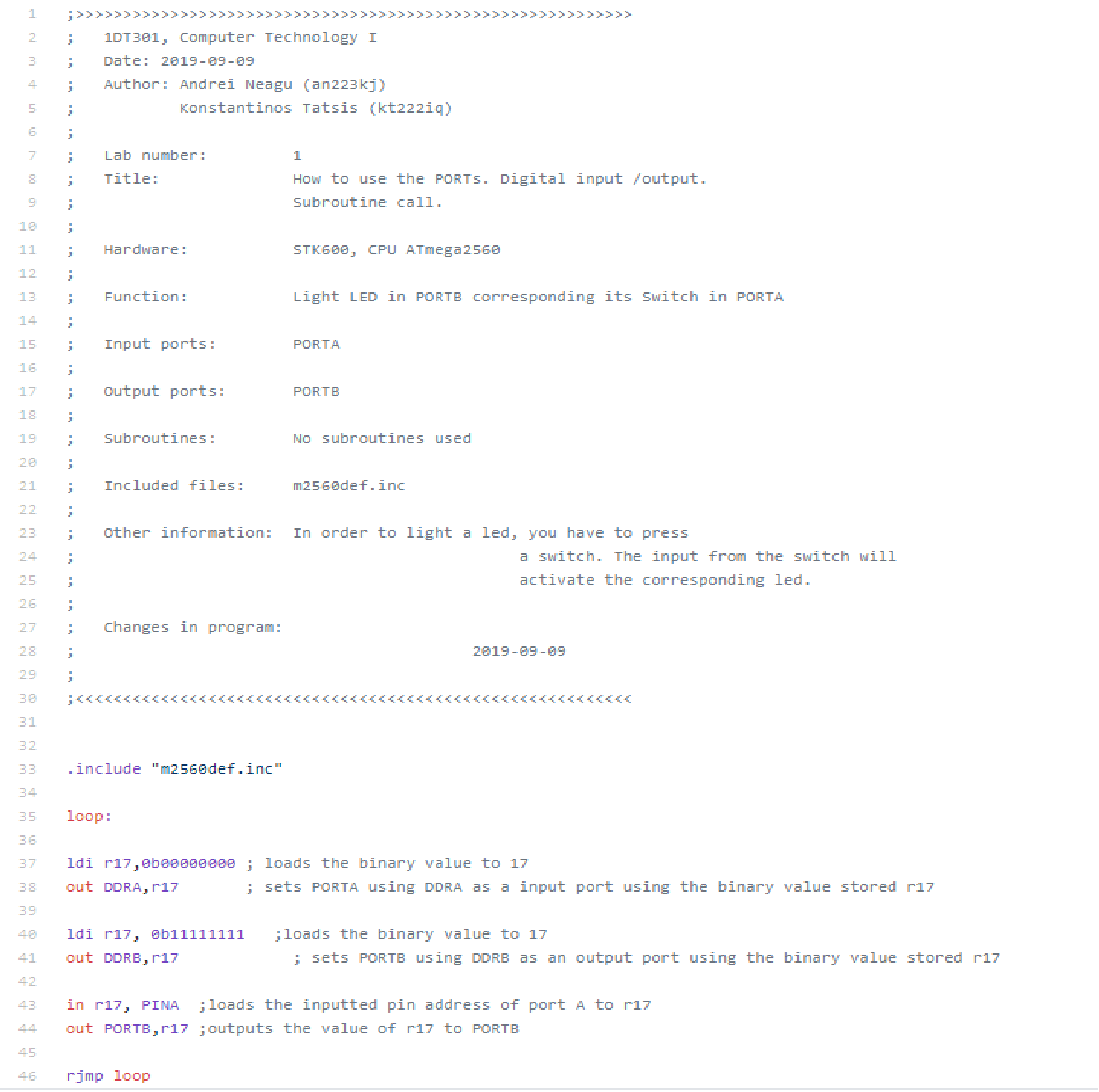
```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;   1DT301, Computer Technology I
;   Date: 2019-09-09
;   Author: Andrei Neagu(an223kj)
;           Konstantinos Tatsis (kt222iq)
;
;   Lab number:         1
;   Title:              How to use the PORTs. Digital input /output.
;                       Subroutine call.
;
;   Hardware:           STK600, CPU ATmega2560
;
;   Function:           Lights LED2 on PORTB
;
;   Input ports:        No input used
;
;   Output ports:       PIN2 on PORTB
;
;   Subroutines:        No subroutines used
;
;   Included files:     m2560def.inc
;
;   Other information:  In order to light a led, you have to set
;                                               the number of the PIN you want to use as 0
;                                               For example: LED2 on would be 0b11111011
;
;   Changes in program:
;                                               2019-09-09
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"

ldi r16, 0b00000100 ; Loading bit value to the register
out DDRB, r16 ; Writing to "Data Direction register for port B" the bit value that we loaded above.
```

Flowchart 1:



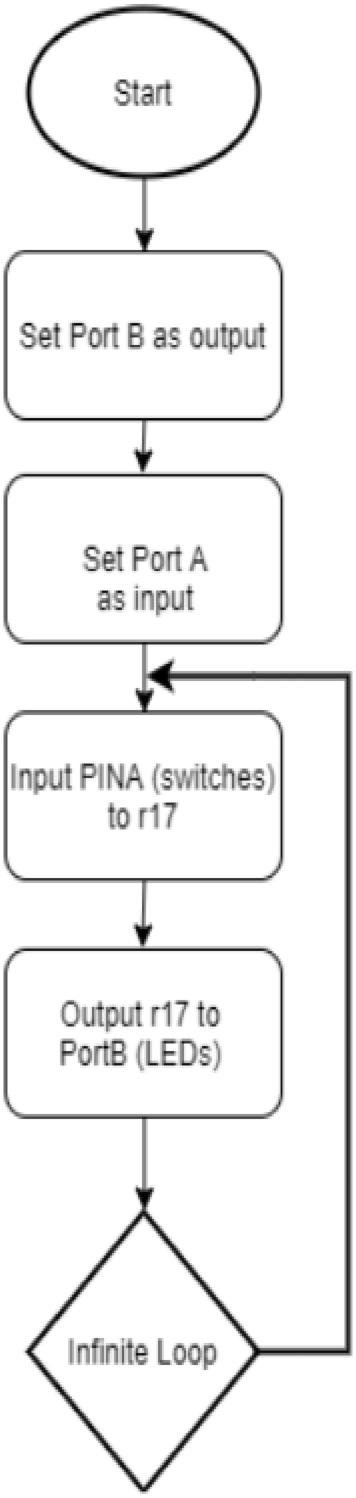
Task 2:

Write a program in Assembly language to read the switches and light the corresponding LED. Example: When you press SW5, LED5 so should light. Make an initialization part of the program and after that an infinite loop.



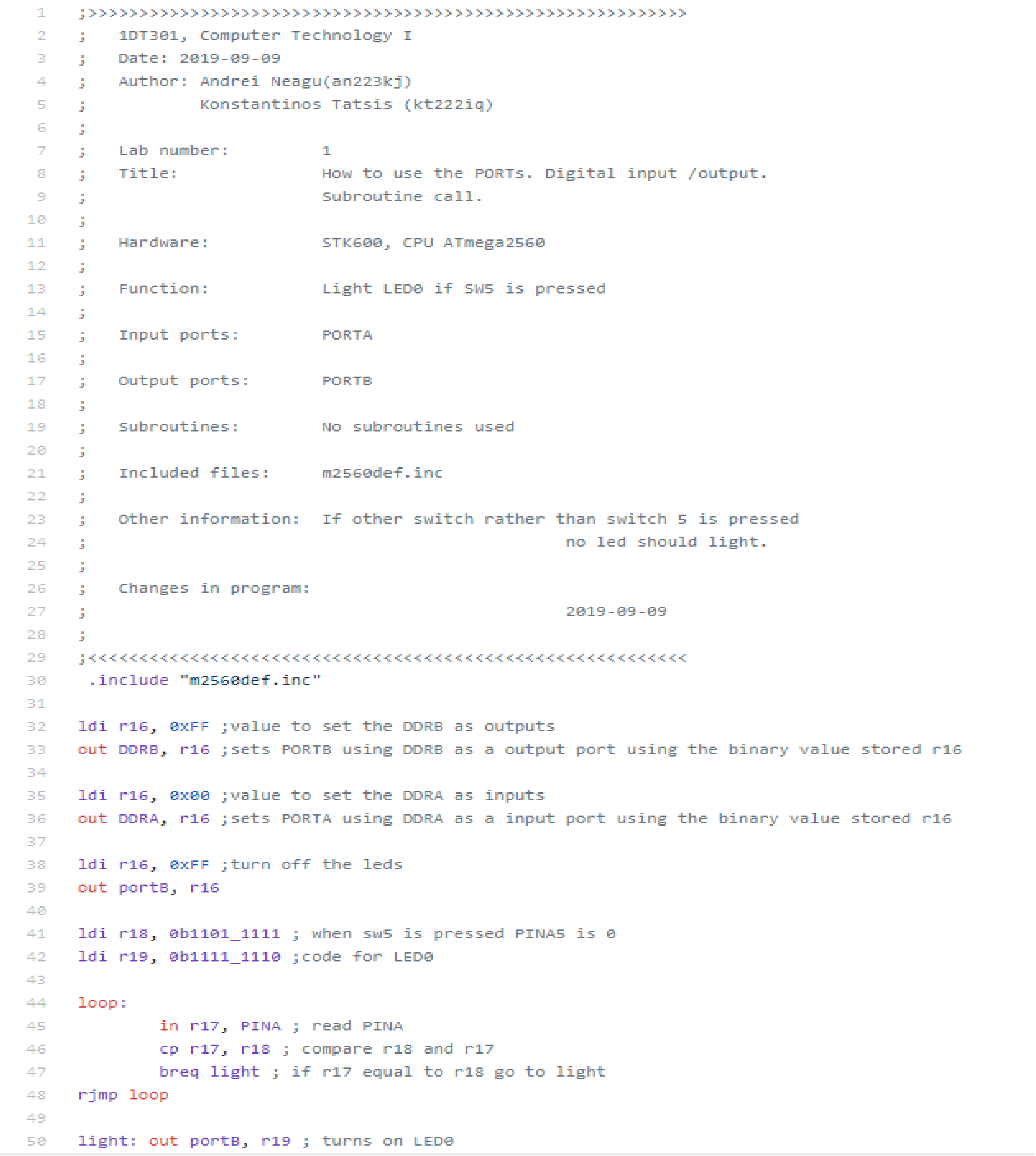
```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;   1DT301, Computer Technology I
;   Date: 2019-09-09
;   Author: Andrei Neagu (an223kj)
;           Konstantinos Tatsis (kt222iq)
;
;   Lab number:         1
;   Title:              How to use the PORTs. Digital input /output.
;                       Subroutine call.
;
;   Hardware:           STK600, CPU ATmega2560
;
;   Function:           Light LED in PORTB corresponding its Switch in PORTA
;
;   Input ports:        PORTA
;
;   Output ports:       PORTB
;
;   Subroutines:        No subroutines used
;
;   Included files:     m2560def.inc
;
;   Other information:  In order to light a led, you have to press
;                                               a switch. The input from the switch will
;                                               activate the corresponding led.
;
;   Changes in program:
;                                          2019-09-09
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<


.include "m2560def.inc"

loop:

ldi r17,0b00000000 ; loads the binary value to 17
out DDRA,r17       ; sets PORTA using DDRA as a input port using the binary value stored r17

ldi r17, 0b11111111   ;loads the binary value to 17
out DDRB,r17            ; sets PORTB using DDRB as an output port using the binary value stored r17

in r17, PINA  ;loads the inputted pin address of port A to r17
out PORTB,r17 ;outputs the value of r17 to PORTB

rjmp loop
```

Flowchart 2:


Start
Set Port B as output
Set Port A as input
Input PINA (switches) to r17
Output r17 to PortB (LEDs)
Infinite Loop

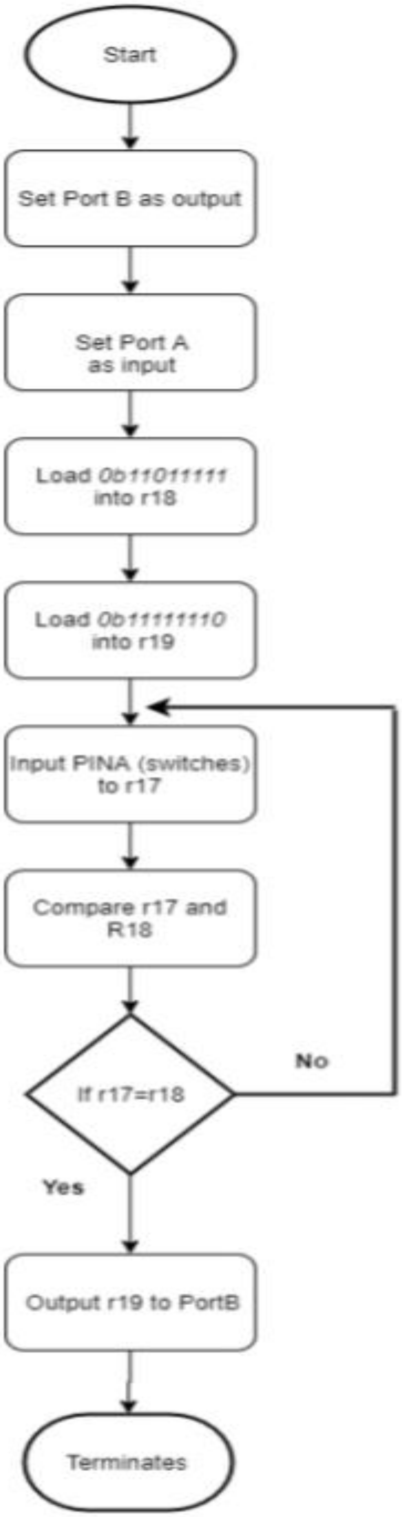
Task 3:

Write a program in Assembly language to read the switches and light LED0 when you press SW5. For all other switches there should be no activity.



```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;   1DT301, Computer Technology I
;   Date: 2019-09-09
;   Author: Andrei Neagu(an223kj)
;           Konstantinos Tatsis (kt222iq)
;
;   Lab number:         1
;   Title:              How to use the PORTs. Digital input /output.
;                       Subroutine call.
;
;   Hardware:           STK600, CPU ATmega2560
;
;   Function:           Light LED0 if SW5 is pressed
;
;   Input ports:        PORTA
;
;   Output ports:       PORTB
;
;   Subroutines:        No subroutines used
;
;   Included files:     m2560def.inc
;
;   Other information:  If other switch rather than switch 5 is pressed
;                                               no led should light.
;
;   Changes in program:
;                                               2019-09-09
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 .include "m2560def.inc"

ldi r16, 0xFF ;value to set the DDRB as outputs
out DDRB, r16 ;sets PORTB using DDRB as a output port using the binary value stored r16

ldi r16, 0x00 ;value to set the DDRA as inputs
out DDRA, r16 ;sets PORTA using DDRA as a input port using the binary value stored r16

ldi r16, 0xFF ;turn off the leds
out portB, r16

ldi r18, 0b1101_1111 ; when sw5 is pressed PINA5 is 0
ldi r19, 0b1111_1110 ;code for LED0

loop:
        in r17, PINA ; read PINA
        cp r17, r18 ; compare r18 and r17
        breq light ; if r17 equal to r18 go to light
rjmp loop

light: out portB, r19 ; turns on LED0
```

Flowchart 3:


Start
Set Port B as output
Set Port A as input
Load *0b11011111* into r18
Load *0b11111110* into r19
Input PINA (switches) to r17
Compare r17 and R18
If r17=r18
No
Yes
Output r19 to PortB
Terminates

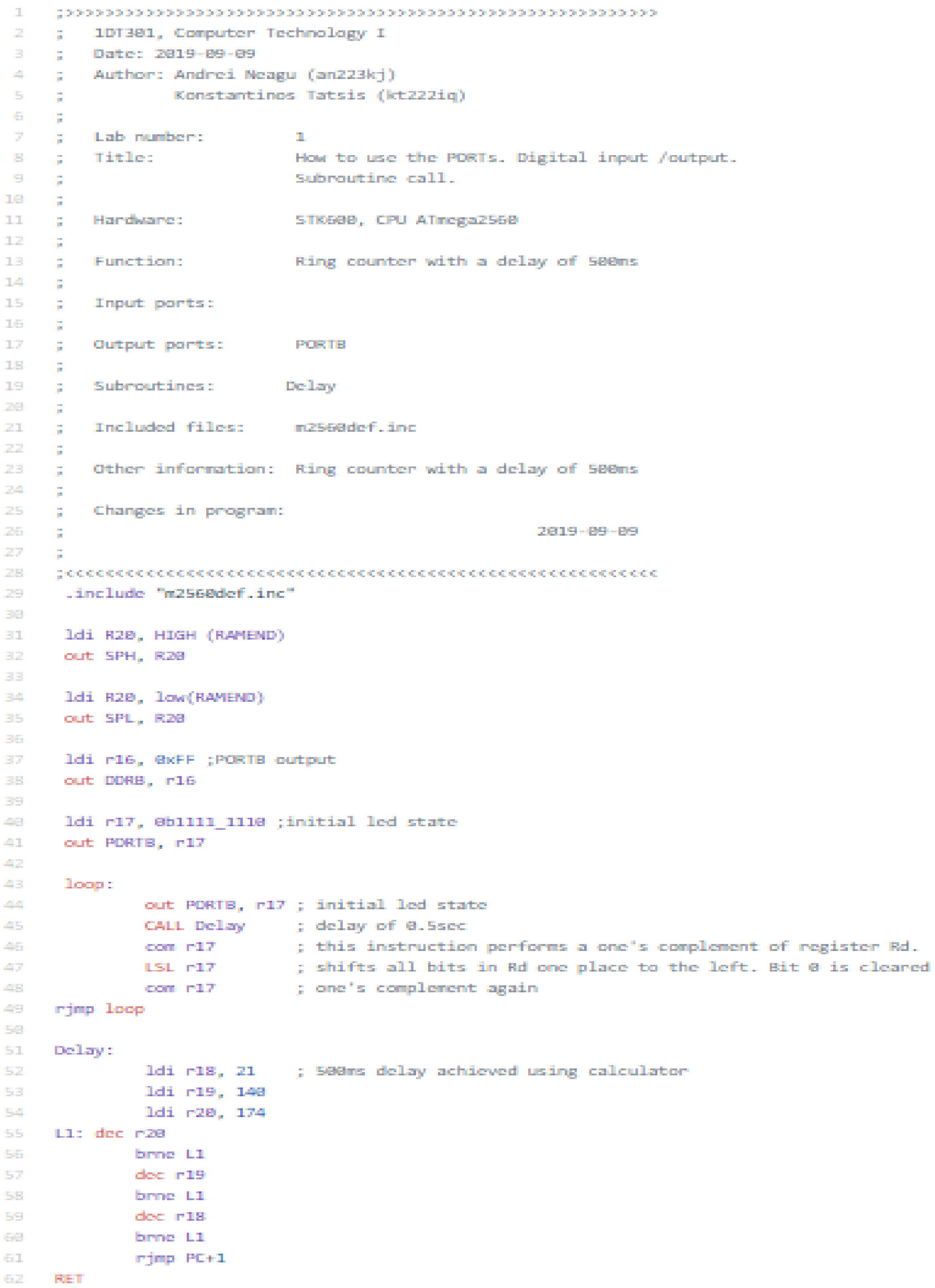
Task 4:

Task 4 must be performed on the computer cannot be displayed.
Continue to task 5.

Task 5:

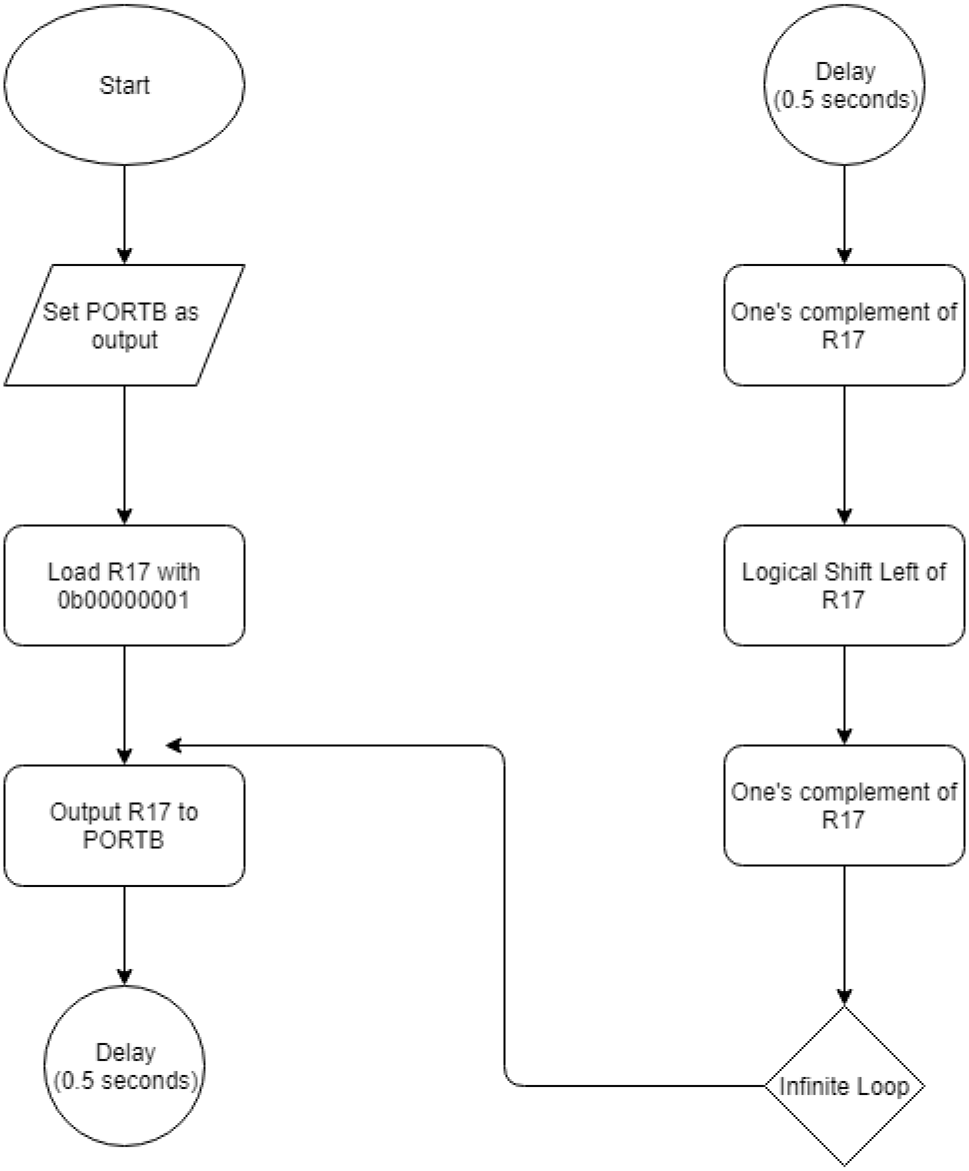
Write a program in Assembly language that creates a Ring Counter. The values should be displayed with the LEDs. Use shift instructions, LSL or LSR. Make a delay of approximately 0.5 sec in between each count. Write the delay as a subroutine. For using the subroutine, you must initialize the Stack Pointer, SP.
Include the following instructions in beginning of your program:

```
=====================================================================

; Initialize SP, Stack Pointer

ldi r20, HIGH(RAMEND) ;        ; R20 = high part of RAMEND address

out SPH,R20 ;                  ; SPH = high part of RAMEND address

ldi R20, low(RAMEND)          ; R20 = low part of RAMEND address

out SPL,R20                   ; SPL = low part of RAMEND address

=====================================================================
```



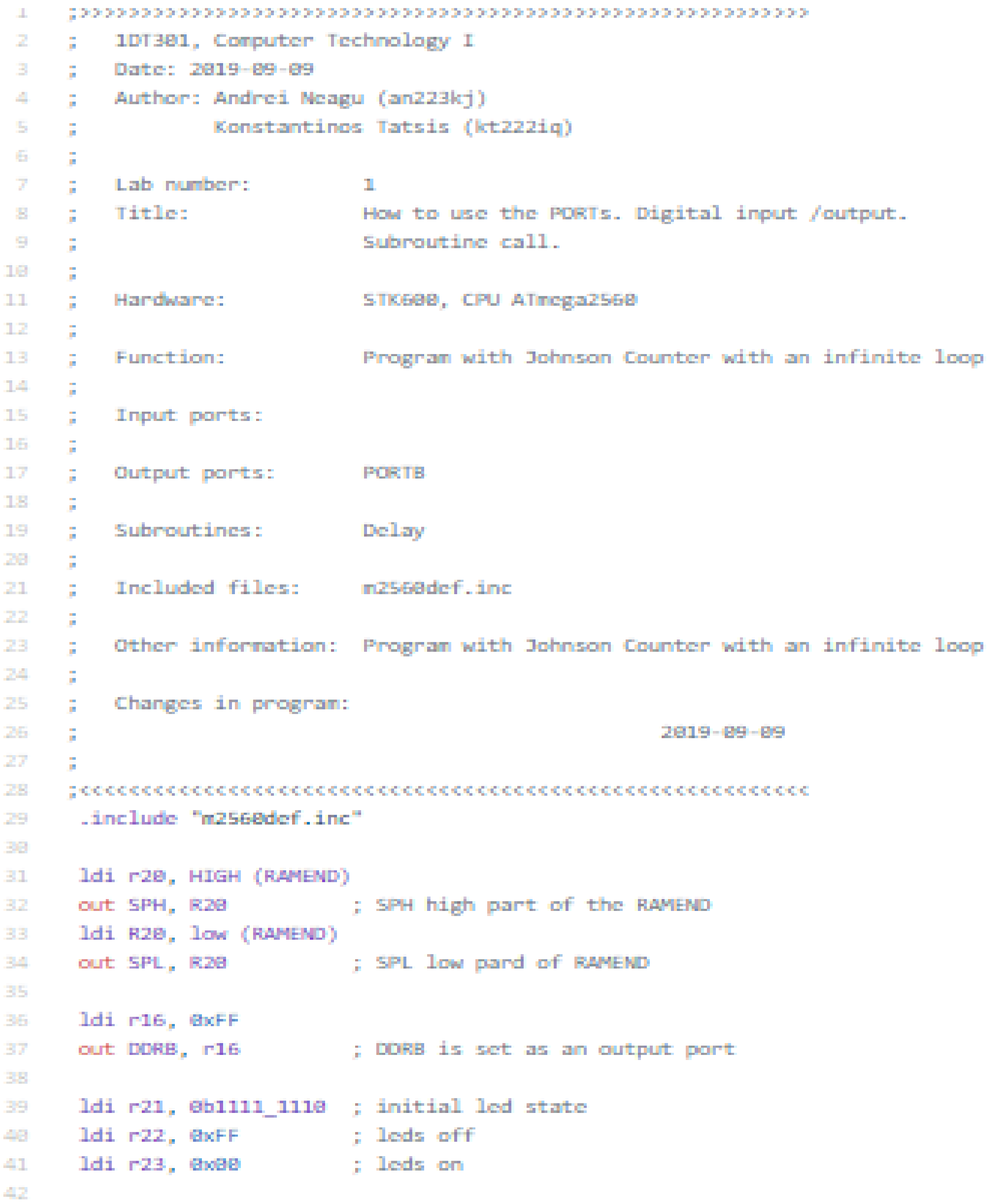
```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;   1DT301, Computer Technology I
;   Date: 2019-09-09
;   Author: Andrei Neagu (an223kj)
;           Konstantinos Tatsis (kt222iq)
;
;   Lab number:           1
;   Title:                How to use the PORTs. Digital input /output.
;                         Subroutine call.
;
;   Hardware:             STK600, CPU ATmega2560
;
;   Function:             Ring counter with a delay of 500ms
;
;   Input ports:
;
;   Output ports:         PORTB
;
;   Subroutines:         Delay
;
;   Included files:       m2560def.inc
;
;   Other information:  Ring counter with a delay of 500ms
;
;   Changes in program:
;                                                 2019-09-09
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 .include "m2560def.inc"

 ldi R20, HIGH (RAMEND)
 out SPH, R20

 ldi R20, low(RAMEND)
 out SPL, R20

 ldi r16, 0xFF ;PORTB output
 out DDRB, r16

 ldi r17, 0b1111_1110 ;initial led state
 out PORTB, r17

 loop:
         out PORTB, r17 ; initial led state
         CALL Delay     ; delay of 0.5sec
         com r17        ; this instruction performs a one's complement of register Rd.
         LSL r17        ; shifts all bits in Rd one place to the left. Bit 0 is cleared
         com r17        ; one's complement again
rjmp loop

Delay:
         ldi r18, 21    ; 500ms delay achieved using calculator
         ldi r19, 140
         ldi r20, 174
L1: dec r20
        brne L1
        dec r19
        brne L1
        dec r18
        brne L1
        rjmp PC+1
RET
```

Flowchart 5:

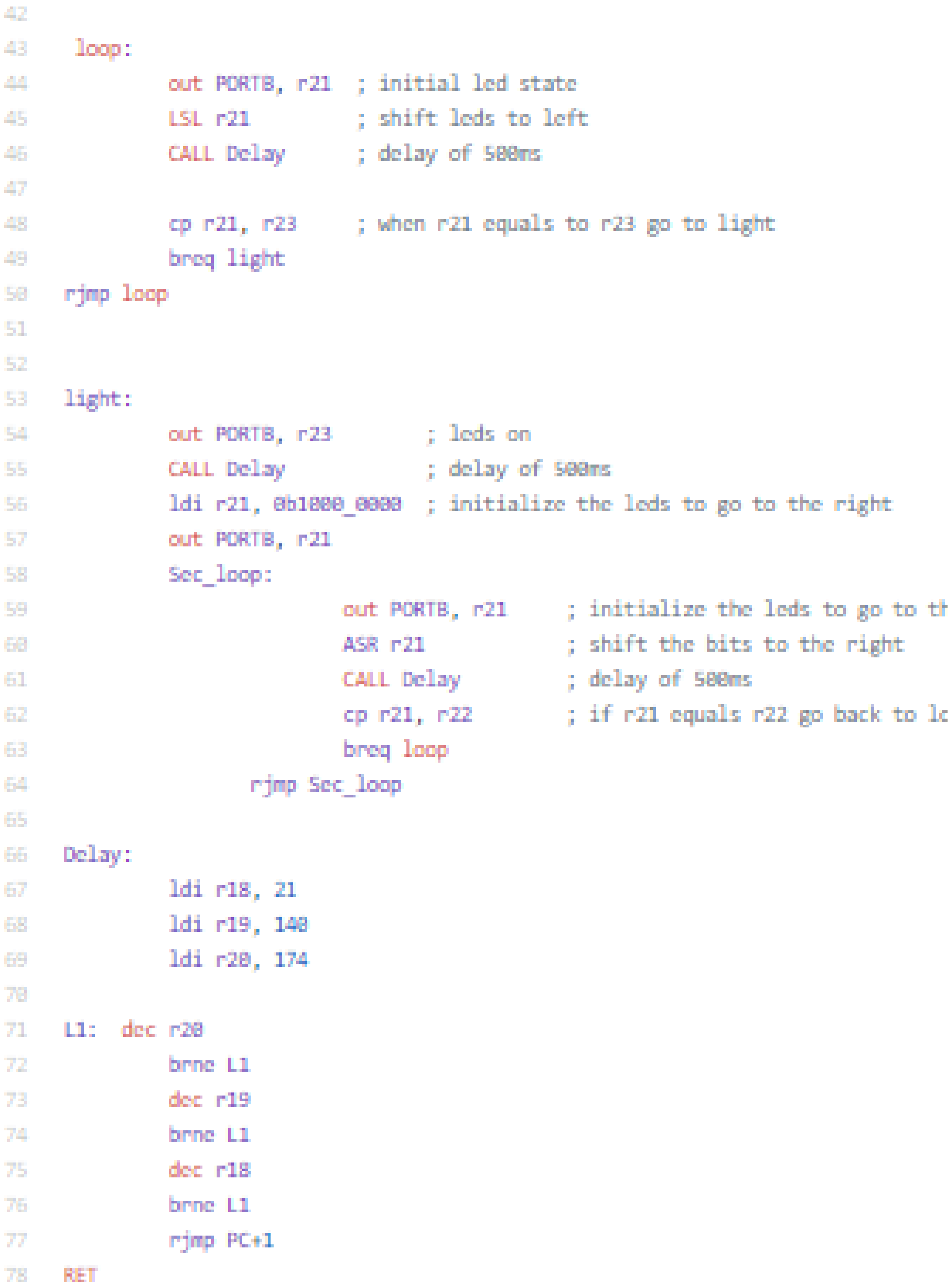

Start
Set PORTB as output
Load R17 with 0b00000001
Output R17 to PORTB
Delay (0.5 seconds)
Delay (0.5 seconds)
One's complement of R17
Logical Shift Left of R17
One's complement of R17
Infinite Loop

Task 6:

Write a program in Assembly language that creates a Johnson Counter in an infinite loop.

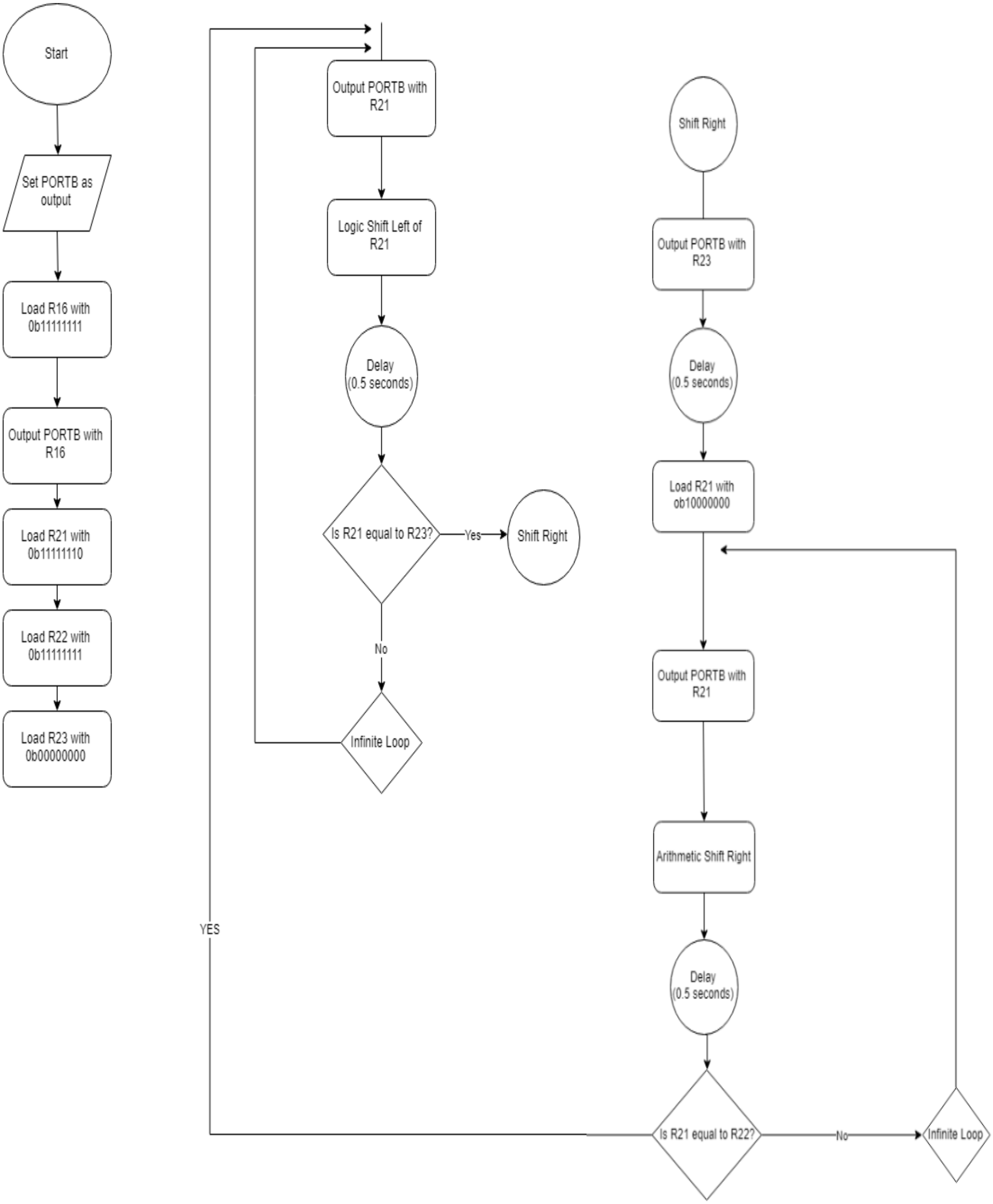


```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;   1DT301, Computer Technology I
;   Date: 2019-09-09
;   Author: Andrei Neagu (an223kj)
;           Konstantinos Tatsis (kt222iq)
;
;   Lab number:         1
;   Title:              How to use the PORTs. Digital input /output.
;                       Subroutine call.
;
;   Hardware:           STK600, CPU ATmega2560
;
;   Function:           Program with Johnson Counter with an infinite loop
;
;   Input ports:
;
;   Output ports:       PORTB
;
;   Subroutines:        Delay
;
;   Included files:     m2560def.inc
;
;   Other information:  Program with Johnson Counter with an infinite loop
;
;   Changes in program:
;                                               2019-09-09
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 .include "m2560def.inc"

 ldi r20, HIGH (RAMEND)
 out SPH, R20          ; SPH high part of the RAMEND
 ldi R20, low (RAMEND)
 out SPL, R20          ; SPL low pard of RAMEND

 ldi r16, 0xFF
 out DDRB, r16         ; DDRB is set as an output port

 ldi r21, 0b1111_1110  ; initial led state
 ldi r22, 0xFF         ; leds off
 ldi r23, 0x00         ; leds on
```



```
 loop:
        out PORTB, r21  ; initial led state
        LSL r21         ; shift leds to left
        CALL Delay      ; delay of 500ms

        cp r21, r23     ; when r21 equals to r23 go to light
        breq light
rjmp loop


light:
        out PORTB, r23        ; leds on
        CALL Delay            ; delay of 500ms
        ldi r21, 0b1000_0000  ; initialize the leds to go to the right
        out PORTB, r21
        Sec_loop:
                        out PORTB, r21     ; initialize the leds to go to th
                        ASR r21            ; shift the bits to the right
                        CALL Delay         ; delay of 500ms
                        cp r21, r22        ; if r21 equals r22 go back to lo
                        breq loop
                rjmp Sec_loop

Delay:
        ldi r18, 21
        ldi r19, 140
        ldi r20, 174

L1:  dec r20
        brne L1
        dec r19
        brne L1
        dec r18
        brne L1
        rjmp PC+1
RET
```

Flowchart 6:


Start
Set PORTB as output
Load R16 with 0b11111111
Output PORTB with R16
Load R21 with 0b11111110
Load R22 with 0b11111111
Load R23 with 0b00000000
Output PORTB with R21
Logic Shift Left of R21
Delay (0.5 seconds)
Is R21 equal to R23?
Yes
Shift Right
No
Infinite Loop
YES
Shift Right
Output PORTB with R23
Delay (0.5 seconds)
Load R21 with ob10000000
Output PORTB with R21
Arithmetic Shift Right
Delay (0.5 seconds)
Is R21 equal to R22?
No
Infinite Loop