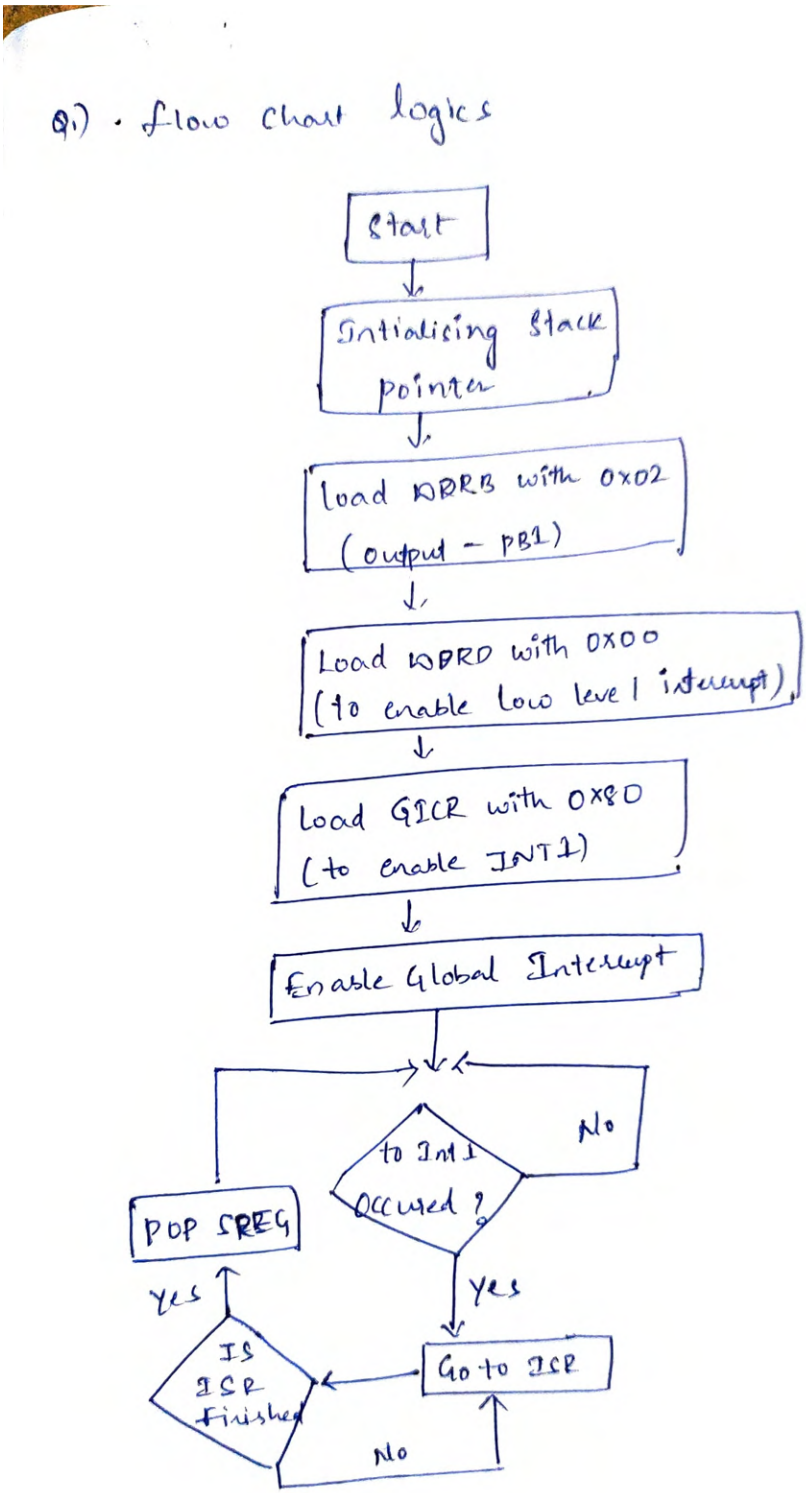
AVR INTERRUPT PROGRAMMING

NAME : SHAILESH PUPALWAR
ROLL NO : EE20B100

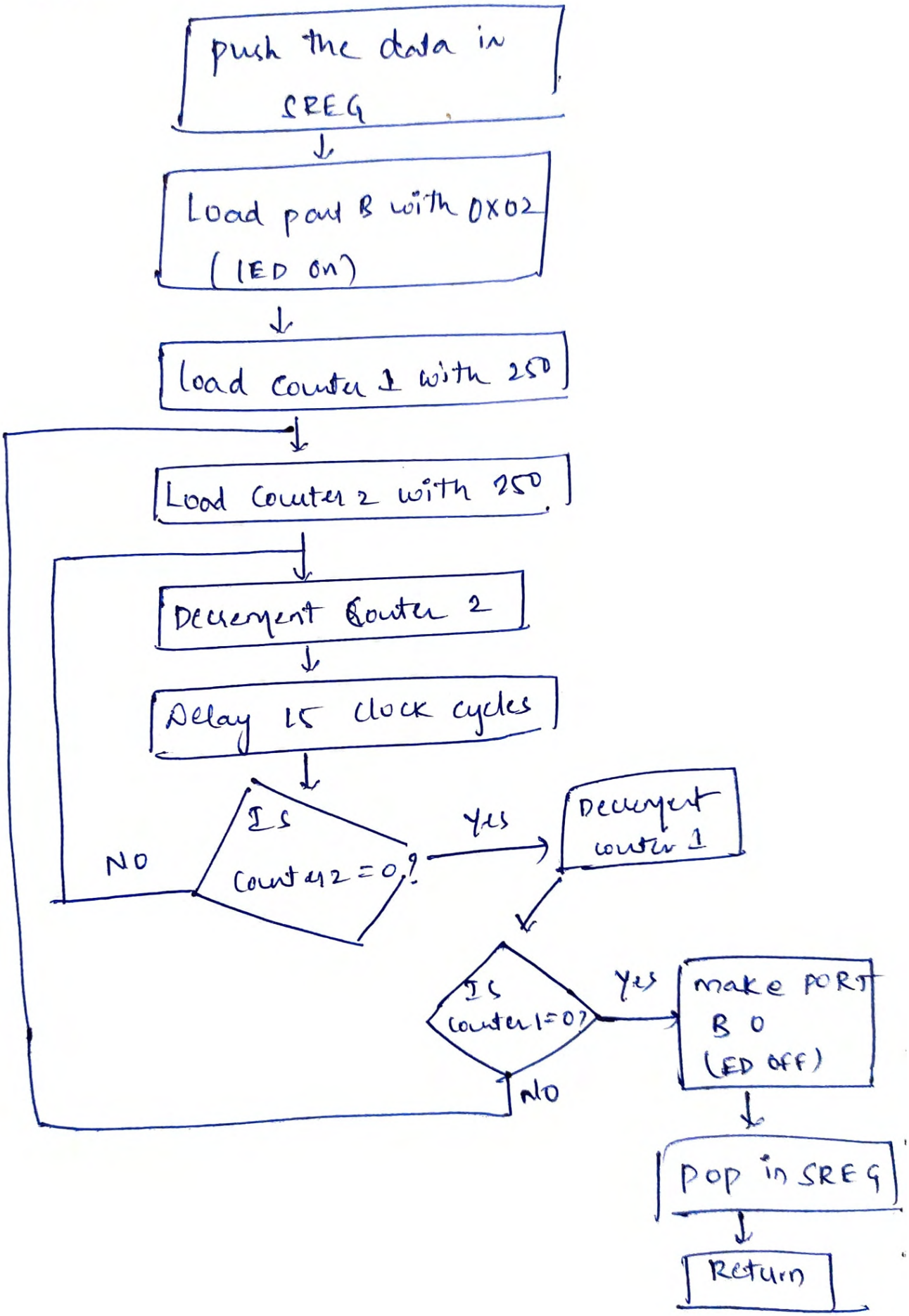
1) SWITCHING ON LED FOR 1 SEC USING INTERRUPT 1 IN ASSEMBLY

The main target of the experiment is to switch on the led for 1 sec when the switch is on using interrupt 1 from pin d.3. The led is connected to pin b.1 which is the output. Whenever the interrupt_1 occurs (i,e. low level in pind.3) the ISR executes and led will be on for 1 sec and off. After ISR is completed the control comes back to the main program.

Flow chart:



ISR:


push the data in SREG
Load port B with 0X02 (LED ON)
load counter 1 with 250
Load Counter 2 with 250
Decrement Counter 2
Delay 15 clock cycles
Is Count 2 = 0 ?
NO
Yes
Decrement counter 1
Is counter 1 = 0 ?
Yes
NO
make PORT B 0 (LED OFF)
Pop in SREG
Return

CODE:

```
.org 0x0000
rjmp reset

.org 0x0002
rjmp int1_ISR

.org 0x0100

reset:
         ;Loading stack pointer address
          LDI R16,0x70
         OUT SPH,R16
         LDI R16,0x00
         OUT SPL,R16

         ;Interface port B pin1 to be output
         LDI R16,0x02
               OUT DDRB,R16
         ;so to view LED blinking

         ; port d as input
         LDI R16,0x00
         OUT DDRD,R16



         ;Set MCUCR register to enable low level interrupt
      LDI R16,0X00
         OUT MCUCR,R16

         ;Set GICR register to enable interrupt 1
      LDI R16,0X80
         OUT GICR,R16

      ; CLEARING THE PORT B
         LDI R16,0x00
         OUT PORTB,R16

         SEI
ind_loop:rjmp ind_loop    ; REMAINS HERE FOR A WHILE

int1_ISR:IN R16,SREG
               PUSH R16

               LDI R16,0x0A
               MOV R0,R16
               ;Modify below loops to make LED blink for 1 sec
       c1:     LDI R16,0x02
               OUT PORTB,R16
```

```
        LDI R16,250
a1:     LDI R17,250
a2:     DEC R17
     NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        BRNE a2
        DEC R16
        BRNE a1

         ; Calculation of delay = (250*16 -1)*250 + 4*250 -1 + 1
        ; without considering over head we get 1000000 which corresponds to 1
sec

        LDI R16,0x00
        OUT PORTB,R16

         ; recovering the value in SREG
        POP R16
        OUT SREG,R16
        ; returning from interrupt
               RETI
```

2) BLINKING LED FOR 10 TIMES WHEN SWITCH IS ON USING INTERRUPT 0 (ASSEMBLY)

The main target of this experiment is to blink the led for 1 sec (i.e. 1 sec on and 1 sec off) for 10 times. This has to be implemented using interrupt 0. Port B.1 wil be connected to led and acts as a output. When the interrupt 0 occurs , the led blinks 10 times and switches off. Inorder to switch on the interrupt we have to make pin d.2 as 0 (low level interrupt). After the ISR is completed , the control comes back to the previous program.

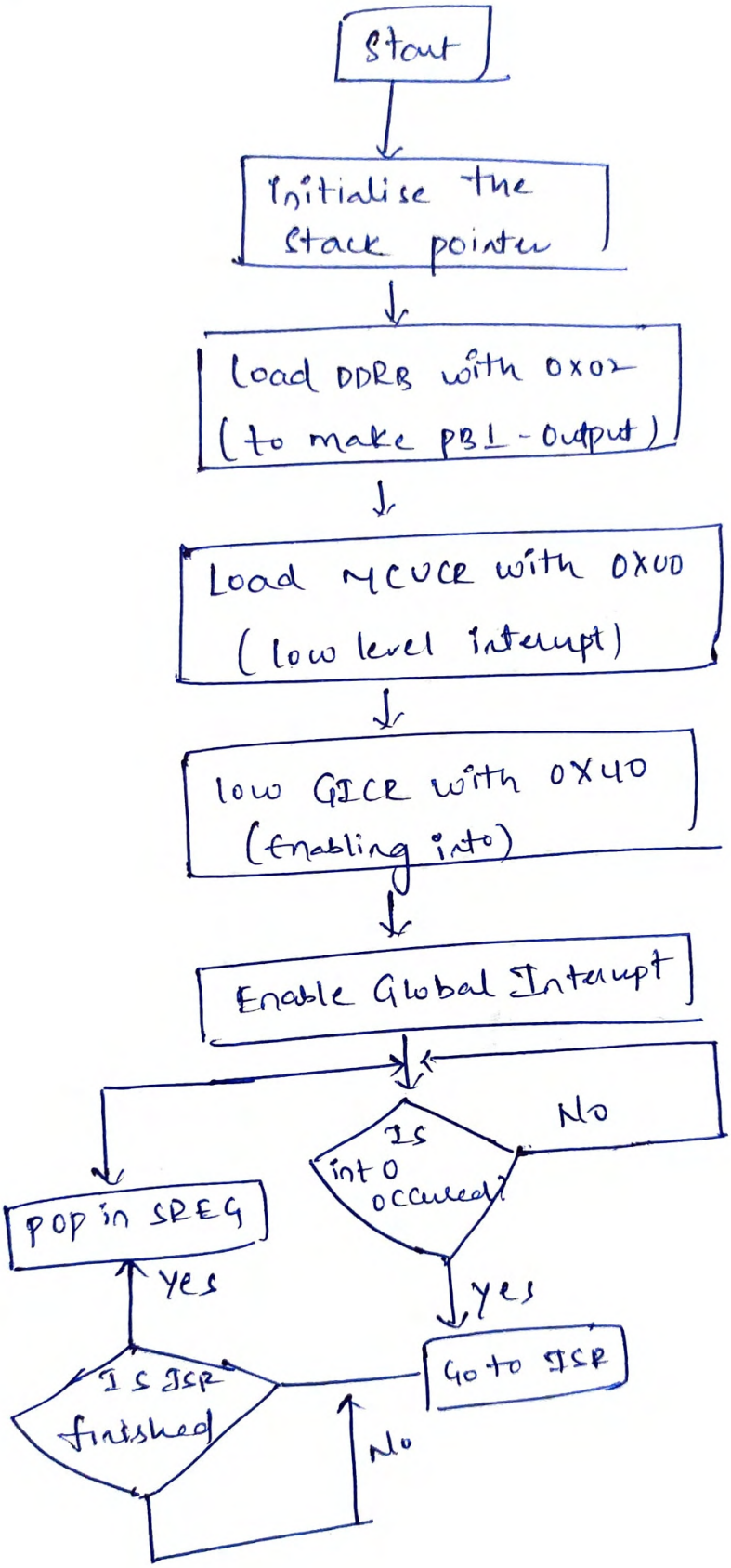
2. Use int0 to redo the same in the demo program (duely filled in). Once the switch is pressed the LED should blink 10 times (ON (or OFF) - 1 sec, duty cycle could be 50 %). Demonstrate both the cases.

CODE:

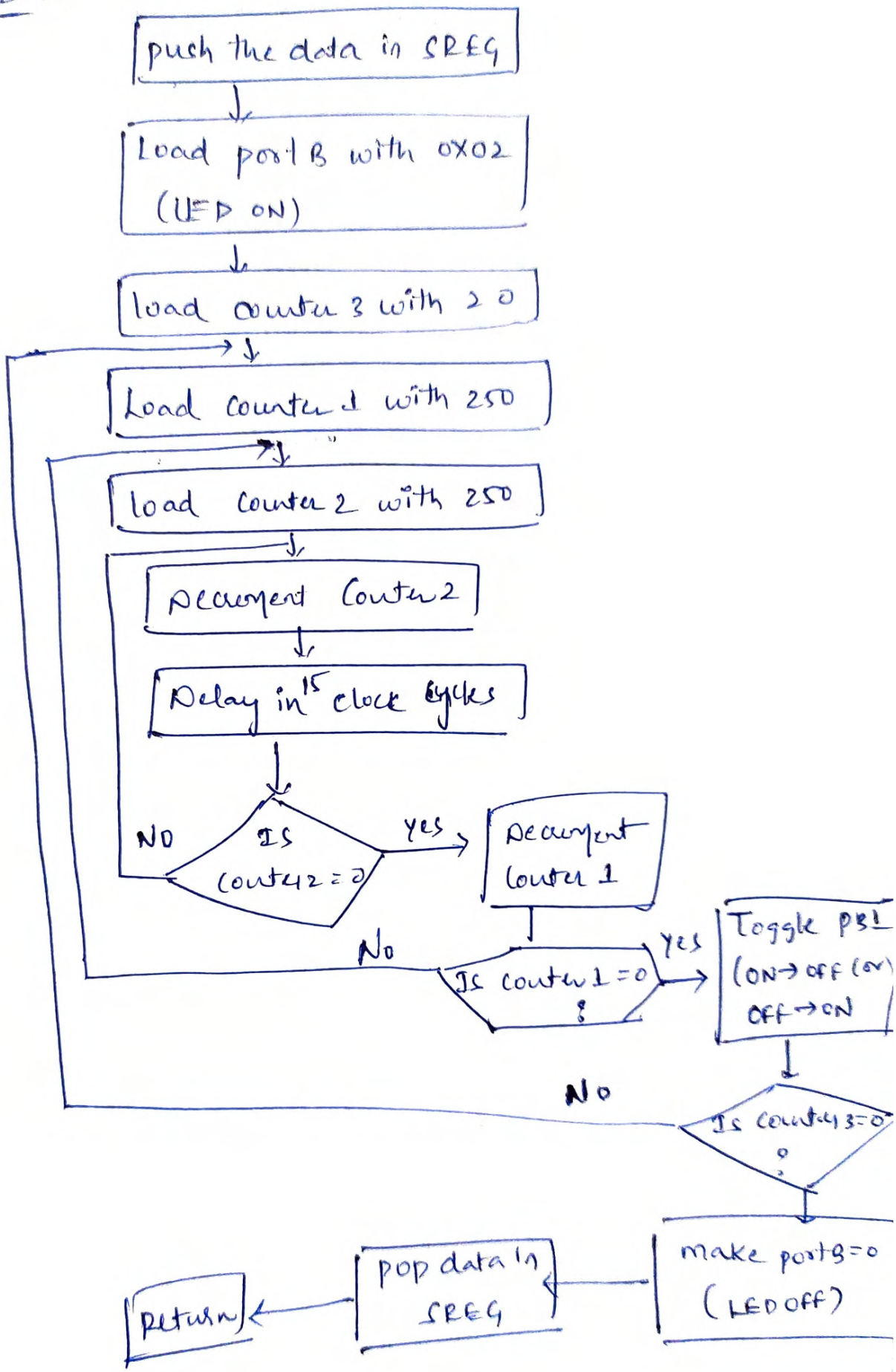
```
; Replace with your application code
.org 0X0000
rjmp reset
.org 0X0001
rjmp int0_ISR
.org 0X0100
reset:   ; Loading stack pointer address
        LDI R16,0x70
        OUT SPL,R16
        LDI R16,0X00
        OUT SPH,R16

              ;Interface port B pin1 to be output
              LDI R16,0x02
              OUT DDRB,R16
              ;So to view LED Blinking

              LDI R16,0x00
              OUT DDRD,R16  ;making PORT D as input port


        ;SET MCUCR registor to enable LOW level interrupt
              LDI R16,0X00
              OUT MCUCR,R16

              ;SET  GICR register to enable interrupt  1
              LDI R16, 0X40
              OUT GICR, R16
              SEI      ;setting global interupt enable

              ind_loop : rjmp ind_loop  ;making infinite loop


              .org 0X200
              int0_ISR:
                          IN R16,SREG
                          PUSH R16
                                  LDI R16,0x0A
                                  MOV R0, R16

        ; to make Led BLINK for 1 sec and switch off for 1 sec for 10 times

        c1:  LDI R16,0x02
             OUT PORTB,R16


               LDI R20,20 ; since led should on for 10 secs and off for 10 secs
again:   LDI R16,250
        a1: LDI R17,250
        a2: DEC R17
            NOP
               NOP
               NOP
               NOP
               NOP
               NOP
               NOP
               NOP
               NOP
               NOP
```

```
            NOP
            NOP
            NOP
      BRNE a2
            DEC R16
            BRNE a1
            ; Calculation of delay
            ; (250*16 -1)*250 + 4*250 -1
            ; without considering over head we get 1000000 which corresponds to 1
sec
            ; as frequency is 1 MHZ
      IN  R21,PORTB
            LDI R22,0x02
            EOR R21,R22
            OUT PORTB,R21
            DEC R20
            BRNE again

             LDI R16,0X00
            OUT PORTB,R16
            ; recovering the value in SREG
            POP R16
            OUT SREG,R16
            ; returning from interrupt
                  RETI
```

FLOW CHART:


Start
Initialise the stack pointer
Load DDRB with 0x02 (to make PB1 - Output)
Load MCUCR with 0x00 (low level interupt)
low GICR with 0x40 (Enabling int0)
Enable Global Interupt
Is int 0 occured?
No
yes
Go to ISR
Is ISR finished
No
yes
POP in SREG

ISR:


push the data in SREG
Load portB with 0X02 (LED ON)
load counter 3 with 20
Load counter 1 with 250
load counter 2 with 250
Decrement Counter2
Delay in 15 clock cycles
IS counter2 = 0
NO
yes
Decrement Counter 1
Is counter 1 = 0 ?
No
yes
Toggle PB1 (ON → OFF (or) OFF → ON
Is counter 3 = 0 ?
NO
make portB = 0 (LED OFF)
pop data in SREG
return

3) Blinking led 10 times when switch is clicked using interrupt1 (C).

The main target of this experiment is to write the above assembly code in C

3. Rewrite the program in 'C' (int1).

CODE:

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

ISR (INT1_vect)
{
	int i;
	for (i=1;i<=10;i++) // 10 TIMES BLINKING FOR LED

	{
		PORTB=0x02;
		_delay_ms(1000);    // DELAY OF 1SEC
		PORTB=0x00;
		_delay_ms(1000);

	}


}
int main(void)
{
	DDRD=0x00;   //Set appropriate data direction for D
	DDRB=0x02;   //Make PB0 as output
	MCUCR=0x00;  //Set MCUCR to level triggered
	GICR=0x80;   //Enable interrupt 1
	PORTB=0x00;
	PORTD=0X08;
	sei();       // global interrupt flag

	while (1) //wait
	{

	}
}
```

1) Blinking led 10 times when switch is clicked using interrupt0 (C).

The main target of this experiment is to write the above C code using interrupt 0.

CODE:

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

ISR (INT1_vect)
{
	int i;
	for (i=1;i<=10;i++) // 10 TIMES BLINKING FOR LED

	{
		PORTB=0x02;
		_delay_ms(1000);    // DELAY OF 1SEC
		PORTB=0x00;
		_delay_ms(1000);

	}


}
int main(void)
{
	DDRD = 0x00 ;   //Set appropriate data direction for D
	DDRB=0x02;   //Make PB0 as output
	MCUCR=0x00;  //Set MCUCR to level triggered
	GICR=0x40;   //Enable interrupt 0
	PORTB=0x00;
	PORTD=0X04;
	sei();       // global interrupt flag

	while (1) //wait
	{

	}
}
```

Final Outcomes of Experiment :

1. Usage of loops in assembly language for hardware
2. Use of stack pointer
3. Using different interrupts for hardware
4. Understanding I/O programming
5. Hardware and software aspects in Avr - Atmega 8.
6. Handling timers and delays
7. Clear understanding of Microchip studio IDE
8. Programming timers and interrupts with C language and assembly language