5.10 Write a program to subtract 7F9AH to BC48H and save the result in RAM memory locations starting at 40H

```
f1hnum equ 0x20
filnum equ 0x21
sehnum equ 0x22
selnum equ 0x23
reshgh equ 0x40
serlow equ 0x41

movlw 0x7f
movwf fihnum
movlw 0xah
movwf filnum
movlw 0xbc
movwf sehnum
movlw 0x48
movwf selnum

movf filnum
subwf selnum,w
movwf reslow
movf fihnum
subwfb selnum,w
movwf reshgh
```

5.11 Write a program to add BCD 7795H to 9548H and save the BCD result in Ram memory locations starting at 40H

```
 firstlow  equ 0x20
 firsthigh  equ 0x21
 resultlow  equ 0x40
 resultmedium  equ 0x41
 resulthigh  equ 0x42

   movlw 0x95
   movwf firstlow
   movlw 0x77
   movwf firsthigh
   movlw 0x48
   addwf firstlow,w
   daw
   bnc next
   incf firsthigh
 next:
   movwf resultlow
   movlw 0x95
   addwf firsthigh,w
   daw
   movwf resultmedium
   bnc ending
   incf resulthigh
   ending:
   bra $
```

5.28 Assume that MYREG = 85H indicate if it skips after comare is executed in each of the following cases:

a) Movlw 0x90	b) movlw 0x70	c) movlw 0x85	d)Movlw 0x5D
cpfsgt Myreg	cpfsgt myreg	cpfseq Myreg	cpfslt Myreg
incf Myreg, f	incf myreg,f	incf Myreg,f	incf Myreg,f
addwl 0x2	addlw 0x2	addlw 0x2	addlw 0x2

a) cpfsgt does not make program to skip "incf myreg, f" operation

b) cpfsgt makes program skip "incf myreg, f" operation

c) cpfseq makes program skip "incf myreg, f" operation

d) cpfslt does not make program to skip "incf myreg, f" operation

5.32 write a program that finds the number of zeros in an 8-bit data item.

```
counter equ 0x20
dataitem equ 0x21
result equ 0x22

movlw 0x08
movwf counter
movlw 0
movwf result
again:
        BTFSS dataitem,1
        incf result,f
        RRNCF dataitem,f
        decf counter,f
        bnz again
        bra $
```

5.34 Write a program that finds the position of the first high in an 8-bit data item.

Data is scanned from D7 to D0 Give the result to 68H

```
counter equ 0x20
dataitem equ 0x21
position equ 0x68

movlw 0x08
movwf counter
movlw 0x07
movwf position
BCF STATUS,C
again:
        RRCF dataitem,f
        BTFSS dataitem,1
        decf position,f
        BC ending
        decf counter,f
        bnz again
        ending:
        bra $
```

9.20 assume that XTAL = 20MHz Find the TMR1H,TMR1L value needed to generates time delay of 2ms

Use 16-bit mode and the largest prescaler possible

TCON:

TMR1ON = 1

TMR1CS = 0

T1SYNC = 0

T1OSCEN = 0

T1CKPS0 = 1

T1CKPS1 = 1

-- = 0

RD16 = 1

T1CON 1011 0001 = 0xB1

Calculating TMR1H and TMR1L:

20Mhz/4 = 5Mhz

2ms= (1/(5Mhz)*8)*(FFFF)-init+1

2ms= (1/(5Mhz)*8)*(2^16)-init+1

2ms= (1/(0.2us*8))*(2^16)-init+1

init= (2^16)-(2ms/(0.2us*8))+1 ;

init= 65536-1250+1 = 64285 = FB1DH

Answer:

TMR1H = 0xFB

TMR1L = 0x1D

T1CON 0xB1

9.25 Program Timer0 to generate square wave of 1kHz assume that XTAL=10MHz

TCON:

TMR0ON =0

T08BIT = 0 (using 16-bit mode)

T0CS = 0 (Using instruction cycle)

T0SE = 0

PSA = 1 (no prescaler is being used)

T0PS 0-2 =0

TCON = 00001000 = 0x08

Calculating TMR1H and TMR1L:

1khz =1ms

1ms= (1/(2.5Mhz))*(FFFF)-init+1

1ms= (1/(2.5Mhz))*(2^16)-init+1

1ms= (1/(0.4us))*(2^16)-init+1

init= (2^16)-(1ms/(0,4us))+1 ;

init= 65536-2500+1 = 63037 = F63DH

TMR0H = 0xF6

TMR0L = 0x3D

TCON 0x08

didn't know where this assignment wanted to send the wave so I just picked a random port and pin

```
---Assign inital values to timer
bcf TRISC,2
restart:
movlw 0x08
movlw T0CON
movlw 0xF6
movwf TMR0H
movlw 0x3D
```

```
movwf TMR0L
---clear flags
bcf INITCON,TMR0IF
---start timer
bsf T0CON,TMR0ON
----moitor the flag
loop:
btfss INTCON, TMR0IF
bra loop
----toggle bit
btg PORTC,2
bra restart
```

9.51Program timer2 in Assebly to toggle pin RB3 when it counts up from 0 to 200 Assume that XTAL = 10Mhz

```
T2CON:
        TOUTPS 0-3 = 0000
        TMR2ON = 0
        T2CKPS 0-1 = 0
T2CON = 0x00

BCF TRISBB,3
bcf PORTB,3
Movlw 0x00
movwf T2CON
movlw 0x0
movwf TMR2
Movlw D'200'
Movwf PR2
BSF T2CON,TMR2ON
loop:
        btfss PIR1,TMR2IF
        bra loop
bsf PORTB,3
BCF T2CON,TMR2ON
bra $
```