

PIC Programming: Loop

```
#include <p16f628.inc>
count equ 0x20 ;allocate memory for variable
org 0x000
start movlw 0x00
      banksel TRISB ;MPLAB Directive, Switch to Bank1
      movwf TRISB ;PortB = Output
      banksel PORTB ;Switch to Bank0
      movwf PORTB ;Set PortB all to '0'
again clrf count ;reset count
      movf count,W ;W = count
      movwf PORTB ;Display content of count
      incf count,F ;count++
      movlw 0x05 ;loop counter
      subwf count,W ;W = counter - 5
      btfss STATUS,Z ;Test Zero flag
      goto again
stop goto stop
end
```

PIC Programming: Loop

```
#include <p16f628.inc>
count equ 0x20 ;allocate memory for variable
org 0x000
start movlw 0x00
      banksel TRISB ;MPLAB Directive, Switch to Bank1
      movwf TRISB ;PortB = Output
      banksel PORTB ;Switch to Bank0
      movwf PORTB ;Set PortB all to '0'
      clrf count ;reset count
      movlw 0x05 ;Numbers of loop
      movwf PORTB ;Display
      movwf count ;count = 5
again decfsz count ;count--, count = 0?
      goto display
stop goto stop ;halt
display movf count,W ;W = count
        movwf PORTB ;Display count on PORTB
        goto again ;next
end
```

PIC Programming: Comparison

```
Movlw xx ;W = xx
Subwf register, F ;reg = reg - W
Btfsc STATUS, flag ;check Z or C flag
```

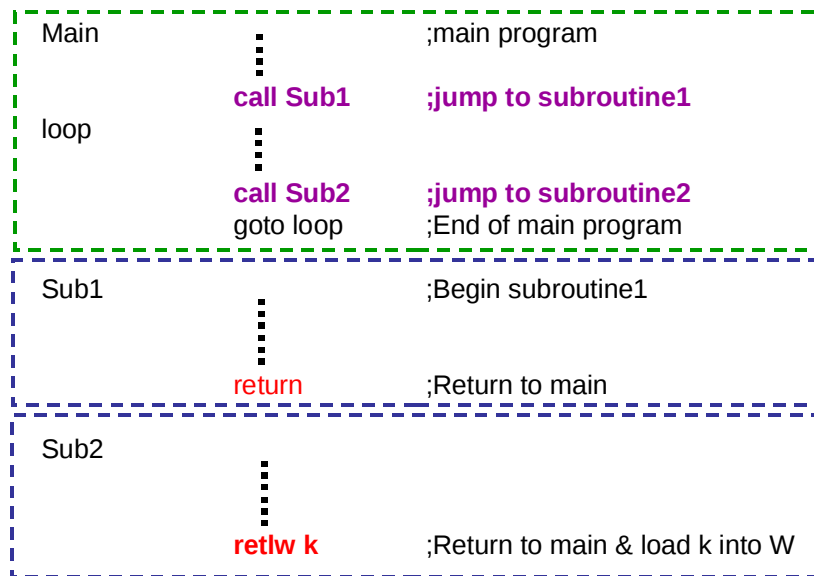
Result:

- Carry flag is set if $\text{reg} \leq W$
- Carry flag is clear if $\text{reg} > W$
- Zero flag is set if $W = \text{reg}$
- Zero flag is clear if $W \neq \text{reg}$

PIC Programming: Comparison

Test For	Use Subtraction followed by	Flag Tested
$W = f$	BTFSS	Z Set
$W \neq f$	BTFSC	Z Clear
$W \leq f$	BTFSS	C Set
$W > f$	BTFSC	C Clear

PIC Programming: Subroutine



PIC Programming: Subroutine

```

#include <p16f628.inc>
count      equ    0x20    ;declare variables
Mcount     equ    0x21    ;M
ncount     equ    0x22    ;N

start      movlw  0x00
           banksel TRISB    ;bank 1
           movwf  TRISB     ;PortB is output
           banksel PORTB    ;bank 0
           clrf   PORTB     ;PortBs are all '0'

get_cnt    clrf   count     ;reset counter
           movf   count,W    ;W = count
           movwf  PORTB     ;W -> PortB
           call   delay     ;call subroutine
           incf   count,F    ;count++
           goto   get_cnt   ;loop
  
```

PIC Programming: Subroutine

```

delay      ;***Subroutine***
           movlw  0x02      ;load M
           movwf  mcount    ;mcount = FFh
loadn      movlw  0x03      ;load N
           movwf  ncount    ;ncount = FFh
decn       decfsz ncount,F  ;ncount--
           goto   decn      ;until ncount=0
           decfsz mcount,F  ;mcount--
           goto   loadn     ;until mcount=0
           return
end
  
```

PIC Programming: Subroutine

```

start      movlw  0x00
           banksel TRISB    ;bank 1
           movwf  TRISB     ;PortB is output
           banksel PORTB    ;bank 0
           clrf   PORTB     ;PortBs are all '0'
           movlw  0x02      ;A number to display
           call   segment
           movwf  PORTB
           goto   loop      ;halt

loop       ;***Subroutine***
segment    addwf   PCL,F    ;PC = PC+W
           retlw  0x3f      ;0 pattern
           retlw  0x06      ;1 pattern
           retlw  0x5b      ;2
           retlw  0x4f      ;3
           retlw  0x66      ;4
           retlw  0x6d      ;5
           retlw  0x7d      ;6
           retlw  0x07      ;7
           retlw  0x7f      ;8
           retlw  0x6f      ;9
           end
  
```