movlw 5 ; move 5 to W
addwf A,W ; add to A, store the result in W
movwf A ; move W to A

movlw 5 ; move 5 to W
addwf A,F ; add to A, store the result in A

There are several ways to do the same thing. Some are more efficient than others

clr A ; clear A

movlw 0 ; this also works
movwf A
A += 2*B;

rlf B,W ; W = 2*B
addwf A,F ; A = A + 2*B
for loops

for (A=8; A>0; A++) B += 1;

movlw 8 ; A = 8
movwf A
Loop incf B,F ; B = B + 1
decfsz A,F ; decrement A
goto Loop ; and repeat until A = 0
End nop
C = max(A,B)
A,B unsigned chars

movf A,W
subwf B,W ; W = A - B. Changes C bit
movf A,W ; W = A. No affect on C bit
btfsc STATUS,C ; bit test, skip if clear
movf B,W ; if B > A, replace W with B
movwf C
Clear bits 0,1,5 of A
- Technique 1: Brute Force
  bcf A,0
  bcf A,1
  bcf A,5
- Technique #2: AND with b1101 1100 = 0xDC
  movlw 0xDC
  andwf A,F
Toggle bit #4 of A
XOR with b0001 0000 = 0x10

movlw 0x10
xorwf A,F
• if (A=5) B = B + 1;
  movlw 5
  subwf A,W
  btfsc STATUS,Z goto End
  incf B,F
End nop
do {
    A = A - 5;
} while (A > 0)

Loop movf A,W ; A = A - 5
    sublw 5
    movf A
    btfss A,7 ; is A < 0? (the sign bit = 1?)
    goto Loop ; no, keep looping
    nop ; yes, exit
16-bit addition

int A, B, C
C = A + B;

movf AL,W ; low 8 bits, CL = AL + BL
addwf BL,W ; sets carry bit
movwf CL ; no affect on C
movf AH,W ; no affect on C
btfsc STATUS,C ; high 8 bits.
addlw 1 ; if a carry, CH = (AH+1) + BH
addwf BH,W ; else CH = AH + BH
movwf CH