

16-bit addition and subtraction

16-bit basic arithmetic is very easy. Using the carry flag, you can simply chain 8-bit operations to perform this simple task on longer integers. Starting from the least significant byte, work your way up to the MSB. The routine is the same for adding and subtracting, except for the fact that when adding, the carry flag is initially cleared and for subtracting, it is set to 1, in order for it being possible to borrow from the carry. You can expand the routines below, putting more bytes to the numbers and repeating the `lda, adc/sbc, sta` block for each of them in order, to create 24-bit, 32-bit or more add and sub routines.

```
; 16-bit addition and subtraction simple example by FMan/Tropyx

!to "16bitaddandsub.prg",cbm    ; compile using ACME

num1lo = $62
num1hi = $63
num2lo = $64
num2hi = $65
resultlo = $66
resulthi = $67

; adds numbers 1 and 2, writes result to separate location

add clc                ; clear carry
  lda num1lo
  adc num2lo
  sta reslo            ; store sum of LSBs
  lda num1hi
  adc num2hi          ; add the MSBs using carry from
  sta reshi          ; the previous calculation
  rts

; subtracts number 2 from number 1 and writes result out

sub sec                ; set carry for borrow purpose
  lda num1lo
  sbc num2lo          ; perform subtraction on the LSBs
  sta reslo
  lda num1hi          ; do the same for the MSBs, with carry
  sbc num2hi          ; set according to the previous result
  sta reshi
  rts
```

Added by enthusi. In case you subtract or add a constant 8bit value to a 16 bit value you can shorten the approach:

```
add clc
  lda my16bit_lsb
  adc #40
  sta my16bit_lsb
```

```
ok      bcc ok
        inc my16bit_msb
        rts
```

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:16bit_addition_and_subtraction

Last update: **2015-04-17 04:30**

