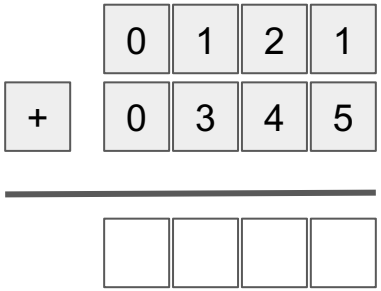
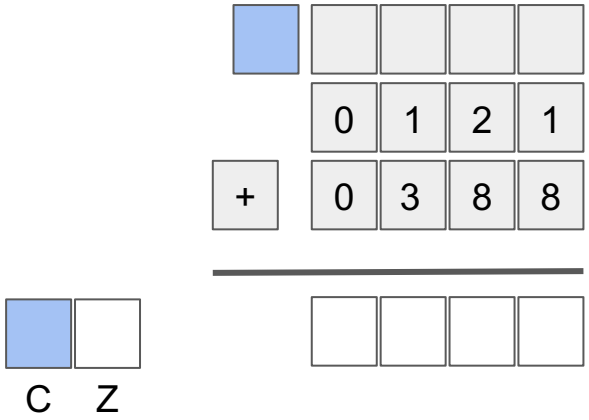
Mathematical Operations

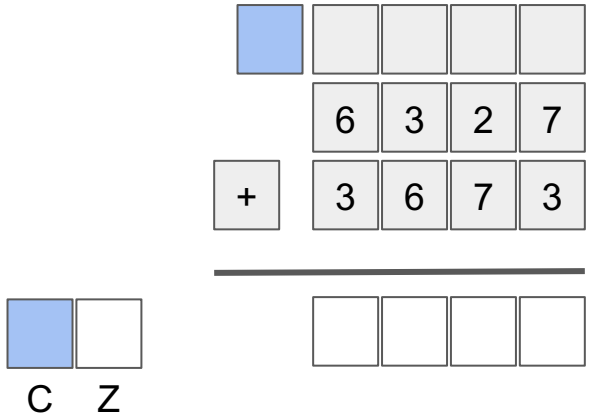
- Base 10: our native base.
- Glyphs: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- The algorithms to evaluate various functions are the same, regardless of base
- On a computer, we are limited to a certain number of digits.
- We can summarize our results: 0 == FALSE, 1 == TRUE
  - For unsigned operations:
    - the final value is Zero (Z)
    - the calculation resulted in final carry (C)
  - For signed values
    - the final value is Negative (S)
    - the calculation resulted in an overflow (V)

Addition: (Before)

- First, introduce some status values:
  - Zero, Carry, (Sign, Overflow)
- Assume a word size of 4
- Notice the notation of "to carry" a value

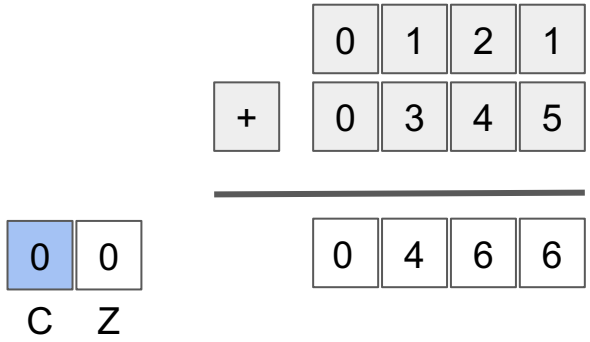


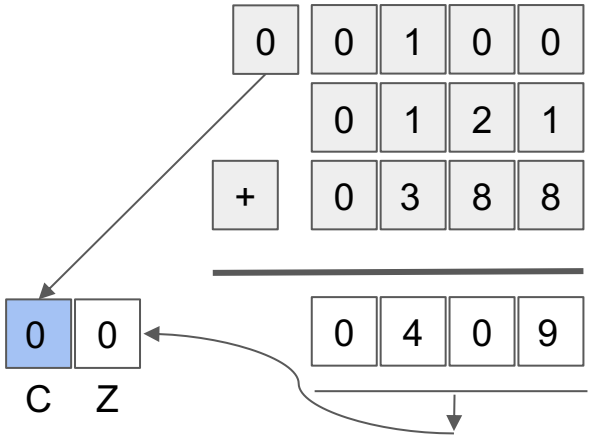


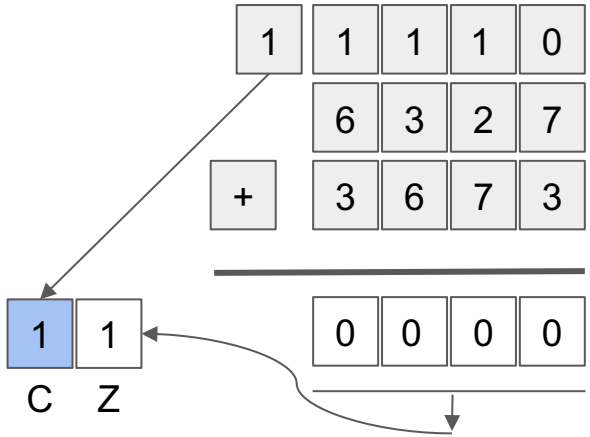


Addition: (After)

- First, introduce some status values:
  - Zero, Carry, (Sign, Overflow)
- Assume a word size of 4
- Notice the notation of "to carry" a value

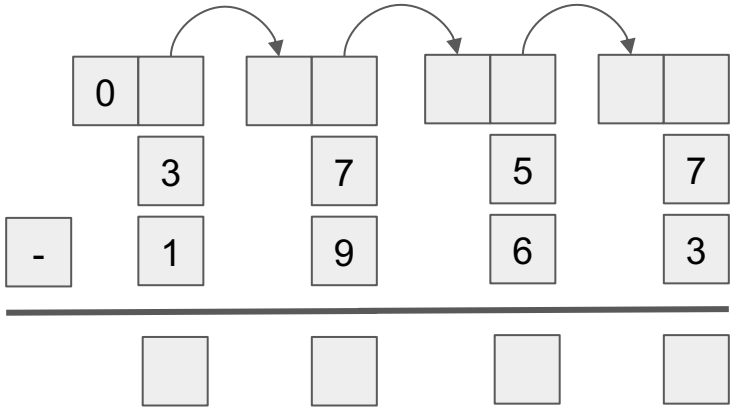






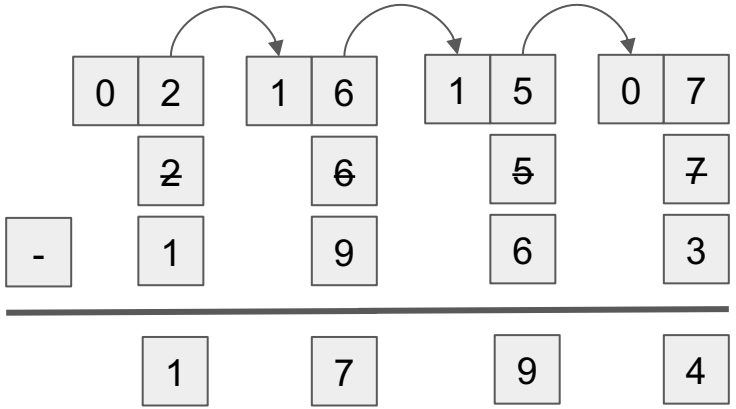
Subtraction (before)

- 3757 - 1963 = 1794
- Traditional Method:
  - Notice the notation of "to borrow" a value
- Other Methods: (common core)
  - Left → Right (Mental Math)
  - Singapore (No Borrow)
  - Counting Up (Giving Change)
- Via Method of Complements



Subtraction (after)

- 3757 - 1963 = 1794
- Traditional Method:
  - Notice the notation of "to borrow" a value
- Other Methods: (common core)
  - Left → Right (Mental Math)
  - Singapore (No Borrow)
  - Counting Up (Giving Change)
- Via Method of Complements



Method of Complements

- A technique to encode both positive and negative numbers
  - uses the same algorithm to perform addition, subtraction performed by the addition of complements

- Complement: *a thing that completes or brings to perfection:* $X + Y = 10^n$ (1 0…0)
- Radix 10:
  - *10's complement*
    - 7 + x = 10 : x is the 10s complements of 7 x = 3
    - 46 + y = 100 : y is the 10s complements of 46 y = 54
  - 9's complement
    - 7 + a = 9 : a is the 9s complements of 7 a = 2
    - 46 + b = 99 : b is the 9s complements of 46 b = 53

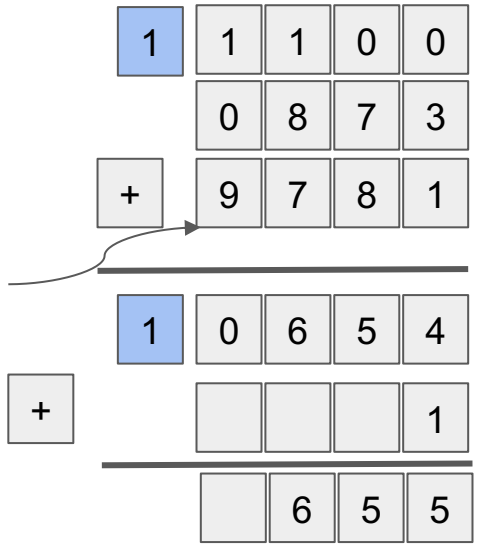
- The math:

2nd Grade	10's complement	9's complement	
45	45	45	33
- 11	+ 89	+ 88	+ 1
34	~~1~~ 34	~~1~~ 33	34

Algorithm: Subtraction via 9's Complements

- Example: 873 - 218 ⇒ 0873 – 0218
  - 4-digit register:
  - 10^5 = 10,000

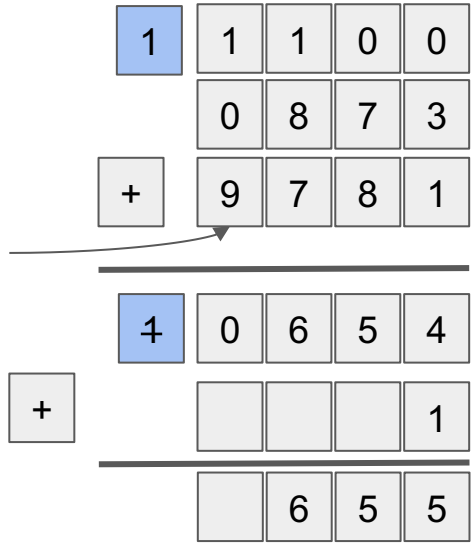
1. Take the nines complement of the subtrahend (0218)
   - With respect to 1 0000
2. Add the complement to the minuend (0873)
3. Drop the leading "1” (the Carry)
4. Add 1



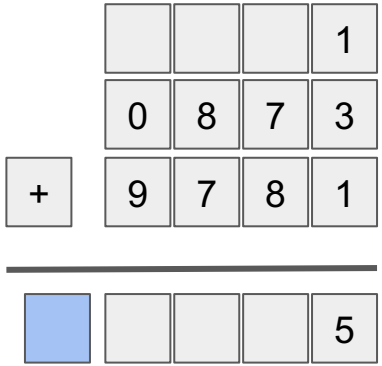
Algorithm: Subtraction via 9's Complements

- Example: 873 - 218 ⇒ 0873 – 0218

1. Take the nines complement of the subtrahend (0218)
2. Add the complement to the minuend (0873)
3. Drop the leading "1"
4. Add 1

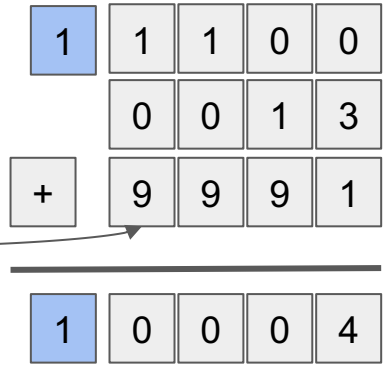


- Optimization:
  introduce initial carry in



Algorithm: Subtraction via 10's Complements

- Example: 13 - 9 ⇒ 0013 – 0009



1. Take the 10s complement of the subtrahend (0009)
2. Add the complement to the minuend
3. Drop the leading "1".

- ~~Optimization~~: Addition of adding one is baked in!