Mathematical Operations

Working with Numbers

Mathematical Operations

- Without the ability to perform basic math operations (i.e. addition and subtraction), having the ability to represent numbers is meaningless.
- Again we will work from what we know about base-10 addition.
- When adding 2 base-10 numbers addition is performed from right to left, least significant digit to most significant digit.

Addition

When two base-10 digits are added:

- if the sum is less than or equal to 9 then the sum is the result
  - 9 being the largest digit in base-10
- if the sum exceeds 9, then the sum is a two digit value
  - the low order digit of the sum is the result
  - the high order digit of the sum is a carry
    - the carry is added with the next pair of digits **or**
    - the number of digits increases in size by 1.

Addition - Binary

Binary addition rules are especially simple:

```
0 + 0          = 0
1 + 0          = 0 + 1 = 1
1 + 1          = 10₂ actually 0 with a carry of 1
1 + 1 + 1      = 11₂ actually 1 with a carry of 1
```

The first three assume a carry-in of 0, having a carry-in of 1 goes to the next rule down

Addition

We can generalize base-10 addition to any base-B (just need to know how to carry).

Addition rules for a base-B are as follows (where $0 <= A,C,D < B$):

$0 + A$	=	A (identity rule)
$C + A$	=	$A + C$ (commutative property)
$(A + C) + D$	=	$A + (C + D)$ (associative property)
$A + C$	=	$1D_B$ where $D = (A + C)$ **mod** B, with a carry of 1 for the 10_B column

Addition

Example: add two binary numbers:

$$
\begin{array}{r}
100110011101_2 \\
+\ \underline{111011001100}_2
\end{array}
$$

Addition

Example: add two binary numbers:

```
Carries:  111110011100

           100110011101_2
        +  111011001100_2
          1100001101001_2
```

Addition

Typically only carries of 1 are included:

$$\begin{array}{rr} \text{Carries:} & 11111111\phantom{{}_2} \\ & 100110011101_2 \\ + & \underline{111011001100}_2 \\ & 1100001101001_2 \end{array}$$

Addition

Your turn

$$\begin{array}{rr} & 01110_2 \\ + & \underline{10110}_2 \end{array}$$

$$\begin{array}{rr} & 102A_{16} \\ + & \underline{\quad 3E1}_{16} \end{array}$$

Addition

Your turn

```
  1111     (carries)
   01110₂
+  10110₂
  100100₂
```

$$\begin{array}{rr} & 1111 \\ & 01110_2 \\ + & \underline{10110_2} \\ & 100100_2 \end{array} \quad \text{(carries)}$$

$$\begin{array}{rr} & 1 \\ & 102A_{16} \\ + & \underline{3E1_{16}} \\ & 140B_{16} \end{array}$$

Subtraction

We can generalize base-10 subtraction to any base-B (just need to know how to carry).

Subtraction rules for a base-B are as follows (where $0 \le A,C,D,E < B$):

$A - 0$	=	A (identity rule)	
$A - A$	=	0	
$CD - A$ (where $A > D$)	=	$(C-1)E_B$	where $E = (D + B) - A$, subtract 1 from the carry column (i.e. column with digit C)

Subtraction - Binary

Binary addition rules are especially simple:

$1 - 0 = 1_2$

$0 - 0 = 0_2$

$1 - 1 = 0_2$

$10 - 1 = 1_2$ (01_2) actually 1 with a borrow of 10_2 or 2

Subtraction

Example: subtract two binary numbers

$$
\begin{array}{r}
101001011_2 \\
-\ \underline{100101110}_2
\end{array}
$$

Subtraction

Example: subtract two binary numbers

$$\begin{array}{r} 10\cancel{1}00\cancel{1}011_2 \\ -\ \underline{100101110}_2 \\ 000011101_2 \end{array}$$

Subtraction

Your turn

$$\begin{array}{r} 10110_2 \\ -\ \underline{01101}_2 \end{array}$$

$$\begin{array}{r} 102A_{16} \\ -\ \underline{3E1}_{16} \end{array}$$

Subtraction

Your turn

$$\begin{array}{r} 10110_2 \\ -\ \underline{01101}_2 \\ 01001_2 \end{array}$$

$$\begin{array}{r} 102A_{16} \\ -\ \underline{\ \ 3E1}_{16} \\ C49_{16} \end{array}$$