Decimal Representation of Numeric Data.

Use digits 0, 1, ..., 9

eg. 1234_{10} $\quad$ 6547_{10}

$A =$ "$a_3 a_2 a_1 a_0$" $= a_3 \times 10^3 + a_2 \times 10^2 + a_1 \times 10^1 + a_0 \times 10^0$

Defn: $(\text{anything})^0 = 1$.

$10^3 = 1000$, $10^2 = 100$, $10^1 = 10$, $10^0 = 1$, $\cancel{10^{-1} = \frac{1}{10}}$

~~a_0, a~~

$$\begin{array}{r} {\scriptstyle 0\,0\,1\,0} \\ 1234 \\ +\,6547 \\ \hline 7781 \end{array}$$

$$\begin{array}{r} {\scriptstyle 1\,0\,0\,0\,0} \\ 6547 \\ +\,8312 \\ \hline 4859 \end{array} \longrightarrow \text{Overflow.}$$

* Overflow is not detected by high level languages.
* Programmer must take measures to ensure it cannot happen.

Subtract? $A - B = A + (-B)$

$\downarrow$

Generate $-B$ from B.

base 10: 10's complement.

$A = 1234$

1. Form the 9's complement of each digit. (subtract each digit from 9).
2. Add 1 to the resulting 4-digit number

$$\begin{array}{lr} A \to & 8765 \\ + & 0001 \\ \hline -A: & 8766 \end{array} \qquad \underline{\text{Verify:}} \qquad \begin{array}{r} 1234 \\ +\ 8766 \\ \hline 0000 \end{array}$$

Multiplication by power of 10.

$A = 0035$, $10 \times A = 0350$.

$100 \times A = 3500$

$1000 \times A = 5000$

$10000 \times A = 0000$

"Left shift".

Division by power of 10:

$A = 1234$ $\quad A \div 10^0 = A \div 1 = 1234$.

$A \div 10^1 = 0123$

$A \div 10^2 = 0012$

$A \div 10^3 \quad 0001$

$A \div 10^4 \quad 0000$

"Right Shift".

Binary Representation of Numerical Data.

Fixed decimals $\longrightarrow$ Fixed binary places.

	4-digit decimal	8-bit binary.
Min	0_{10}	0_2
Max	9999_{10}	$\underbrace{11111111}_{255.}{}_2$
Digits	$0, 1, \ldots, 9$	$0, 1.$
	$a_3 10^3 + a_2 10^2 + a_1 10^1 + a_0 10^0$	$a_7 2^7 + a_6 2^6 + a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0$
Recall.	$10^0 = 1$	$2^0 = 1.$

BIT = Binary digIT.

Examples:

$0 = 00000000_2.$

$1_{10} = 00000001_2.$

$2_{10} = 00000010_2$ *

$3_{10} = 00000011_2.$

$4_{10} = 00000100_2$ *

$5_{10} = 00000101_2.$

$6_{10} = 00000110_2.$

$7 = 00000111_2$

$8 = 00001000_2$ *

$9 = 00001001_2$

$10 = 00001010_2$

$11 = 00001011_2$

$12 = 00001100_2$

Represent Negative Number

Decimal : 10's complement ; 9's comp + 1.

Binary : 2's comp. : 1's comp + 1.

Eg. $75_{10} - 12_{10}$.

$75_{10} = 64 + 11$.

$= 64 + 8 + 2 + 1$.

$= 01001011_2$

-12 : start from $12_{10} = 00001100_2$.

1's comp 11110011_2.

plus 1 $\quad 1$

$11110100 \rightarrow \underline{\underline{-12}}$.

$$\begin{array}{r} 01001011 \\ 11110100 \\ \hline 00111111_2 \end{array} \rightarrow 1+2+4+8+16+32.$$

$= 64 - 1$.

$= 63_{10}$.

Multiplication by Powers of 2 :

$1_{10} = 1_2$; $2_{10} = 10_2$; $4 = \underbrace{100_2}_{2^2}$; $8 = \overbrace{1000_2}^{2^3}$.

Consider 7×2^0. $111_2 \times 1 = 111_2$.

7×2^1

$$\begin{array}{r} 111 \\ +\ 111 \\ \hline 1110 \\ 1110 \\ \hline 011100 \\ +\ 011100 \\ \hline 111000 \end{array}$$

$7 \times 4 = 7 \times 2 \times 2 = 011100$

$7 \times 8 = 7 \times 4 \times 2 = 111000$

"Left shift" corresponds to multiplication by a power of 2.

Similarly, "Right shift" corresponds to division by a power of 2.