1. 2 進数の基本

1. 位取り記数法

数字を並べて数値を表現する方法を，位取り記数法（くらいどりきすうほう）という。例えば，数字 1，2，3 を並べて，数値 123 を表現する方法である。日常使用している方法である。

日常使用する 10 進数の場合は，0～9 の 10 種類の数字を並べて，時には小数点（.）を使用して，数値を表現する。例えば，10 進数の

123.45

は，位取り記数法で表現された数値であるが，これは次の意味である。

$$
\begin{aligned}
123.45 &= 100 + 20 + 3 + 0.4 + 0.05 \\
&= 1\times 100 + 2\times 10 + 3\times 1 + 4\times\frac{1}{10} + 5\times\frac{1}{100} \\
&= 1\times 10^{2} + 2\times 10^{1} + 3\times 10^{0} + 4\times 10^{-1} + 5\times 10^{-2}
\end{aligned}
$$

ここで，百の位（くらい）には10^{2}，十の位には10^{1}，一の位には10^{0}，小数第一位には10^{-1}，小数第二位には10^{-2}という 10 の累乗（るいじょう）がついている。10 の累乗とは，10^{n}の形の数のことである。10^{n} に対して，10 を底（てい），n を指数という。

すなわち，10 進数とは，0～9 の 10 種類の数字を並べたものであり，その意味は，「各位の数字（0～9）」と「10 の累乗」との積の和である。ここで，基本になっている数の 10 を「基数（きすう）」，各桁に掛けられる 10 の累乗を各桁の重（おも）みという。

上記の数値では，重みは 10^{2}，10^{1}，10^{0}，10^{-1}，10^{-2} である。

● 注意

小数とは，小数点のついた数値のことであり，整数部・小数点・小数部からなる。

１２３４.５６７

① 整数部は「１２３４」，整数部の桁数は 4 桁

② 小数部は「５６７」，小数部の桁数（小数点以下の桁数）は 3 桁

③ 次のように表現する。

一の位は 4，十の位は 3，百の位は 2，千の位は 1

$\frac{1}{10}$ の位（小数第一位（だいいちい））は 5，$\frac{1}{100}$ の位（小数第二位（だいにい））は 6，$\frac{1}{1000}$ の位（小数第三位（だいさんい））は 7

● 注意 2

10 進数は小学校 4 年生で学習し，1 つ上の位は 10 倍，1 つ下の位は 1／10 倍になると習う。上の位に進むと，位は 10 倍ずつ増える。

一（10^0），十（10^1），百（10^2），千（10^3），万（10^4），十万（10^5），百万（10^6），千万（10^7），一億（10^8），十億（10^9），百億（10^{10}），千億（10^{11}），一兆（10^{12}），十兆（10^{13}），百兆（10^{14}），千兆（10^{15}），一京（10^{16}），十京（10^{17}），百京（10^{18}），千京（10^{19}）

６７３９５６４１４５４３２を読んで，漢字で書きなさい（４年生の問題）

右から４桁ごとに区切って，千，万，億，兆。

６｜７３９５｜６４１４｜５４３２

六兆 七千三百九十五億 六千四百十四万 五千四百三十二

2. p進数

pを2以上の自然数とする。10進数の場合と同様に，p進数は，p種類の数字

0，1，…，$p-1$

を並べてできる数値である。（pが10より大きい場合は，10以上の数字が必要になるので，そのときは，英字$A, B, \cdots$を使用する。）

例えば，p進数の

① $abc.de$

とは，次の数を意味する。

② $a \times p^2 + b \times p^1 + c \times p^0 + d \times p^{-1} + e \times p^{-2}$

（ここで，a, b, c, d, eは，0，1，…，$p-1$のいずれか）

逆に，②の形で表される10進数は，p進数では①のように表す。

p進数ではpが基数となり，各桁につく重みはpの累乗である。また，p進数であることを明示するために，①を

$$(abc.de)_p$$

のように表す。

（例１）（2進数の場合）

① 使用できる数字は，0，1の2種類のみ。

② 2進数$(101.01)_2$を10進数に直すと

$$(101.01)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$
$$= 1 \times 4 + 0 \times 2 + 1 \times 1 + 0 \times \frac{1}{2} + 1 \times \frac{1}{4}$$
$$= 5 + \frac{1}{4} = 5.25$$

（例２）（3進数の場合）

① 使用できる数字は，0，1，2の3種類のみ。

② 3進数$(112.02)_3$を10進数に直すと

$$(112.02)_3 = 1\times3^2 + 1\times3^1 + 2\times3^0 + 0\times3^{-1} + 2\times3^{-2}$$
$$= 1\times9 + 1\times3 + 2\times1 + 0\times\frac{1}{3} + 2\times\frac{1}{9}$$
$$= 14 + \frac{2}{9} = \frac{128}{9}$$

3. 基本

コンピュータは2進数しか理解できないが，コンピュータの世界では人間にもわかりやすいように，8進数・10進数・16進数も使用される。16進数では，A～Fの6個の英字が使用される。

● **対応関係**

10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

● **2進数と16進数の各桁の重み**

桁	2進数	16進数
1	$2^0 = 1$	$16^0 = 1$ $(= 2^0)$
2	$2^1 = 2$	$16^1 = 16$ $(= 2^4)$
3	$2^2 = 4$	$16^2 = 256$ $(= 2^8)$
4	$2^3 = 8$	$16^3 = 4096$ $(= 2^{12})$
5	$2^4 = 16$	$16^4 = 65536$ $(= 2^{16})$
6	$2^5 = 32$	⋮
7	$2^6 = 64$	
8	$2^7 = 128$	
9	$2^8 = 256$	
10	$2^9 = 512$	
11	$2^{10} = 1024$	
12	⋮	

（注）2進数3桁 ＝ 8進数1桁，2進数4桁 ＝ 16進数1桁

● 10進数（基数＝10）

$582 = 5 \times 100 + 8 \times 10 + 2 \times 1 = 5 \times 10^2 + 8 \times 10^1 + 2 \times 10^0$

各桁の重み：下位から 10^0，10^1，10^2，…

● 2 進数（基数＝2）

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (13)_{10}$$

$2^3 = 8$, $2^2 = 4$, $2^1 = 2$, $2^0 = 1$　←　各桁の重み（2 倍ずつ）

● 8 進数（基数＝8）

$$(572)_8 = 5 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = (378)_{10}$$

$8^2 = 64$, $8^1 = 8$, $8^0 = 1$　←　各桁の重み（8 倍ずつ）

● 16 進数（基数＝16）

0，1，2，…，9，A，B，C，D，E，F

A	B	C	D	E	F	
10	11	12	13	14	15	←　10 進数

$$(3E6)_{16} = 3 \times 16^2 + 14 \times 16^1 + 6 \times 16^0 = (998)_{10}$$　←　E は 14

$16^2 = 256$, $16^1 = 16$, $16^0 = 1$　←　各桁の重み（16 倍ずつ）

4. 基数変換　（*p* 進数 ⇒ 10 進数）

p 進数は，下位から見ると，重みは 1 から始まり *p* 倍ずつ増えていく。

p 進数の重み：$p^0 = 1$，p^1，p^2，p^3，p^4，…

① 2 進数の重み：1，2，4，8，16，32，64，…（2 倍ずつ増える）

② 8 進数の重み：1，8，64，512，…（8 倍ずつ増える）

③ 16 進数の重み：1，16，256，4096，…（16 倍ずつ増える）

p 進数を 10 進数に変換するには，各桁の数字と重みを掛けて，合計すればよい。

● **2 進数 → 10 進数**　$(10111001)_2 = (185)_{10}$

128	64	32	16	8	4	2	1		… 重み（2 倍ずつ）
×	×	×	×	×	×	×	×		
1	0	1	1	1	0	0	1		… 2 進数 8 桁
128	+ 0	+ 32	+ 16	+ 8	+ 0	+ 0	+ 1	= 185	… 10 進数

● **8 進数 → 10 進数**　$(257)_8 = (175)_{10}$

64	8	1		… 重み（8 倍ずつ）
×	×	×		
2	5	7		… 8 進数 3 桁
128 +	40	+ 7	= 175	… 10 進数

● **16 進数 → 10 進数**　　$(2B7)_{16} = (695)_{10}$

256	16	1	… 重み（16 倍ずつ）
×	×	×	
2	B	7	… 16 進数 3 桁　$(B)_{16} = (11)_{10}$
512 +	176	+ 7 = 695	… 10 進数

5. 除法の原理

13 を 2 で割ると，商は 6 で，余りは 1 であり，次の等式が成立する。

$$13 = 2 \times 6 + 1$$

逆に，

$$13 = 2 \times q + r \quad (q, r \text{は整数で，} 0 \le r \le 1)$$

が成立すれば，$q = 6$，$r = 1$ となる。このことは，次の除法の原理で保証されている。

● **除法の原理**

自然数 p を定めたとき，任意の整数 a に対して，

$$a = pq + r,\quad 0 \le r \le p - 1$$

をみたす整数 q, r がそれぞれただ 1 つ存在する。

q を「a を p で割ったときの商」，r を「a を p で割ったときの余り」という。

6. 基数変換　(10 進数 ⇒ p 進数)

10 進数を p 進数で表すには，10 を p で割り，その商を p で割り，その商を p で割り…という計算を繰り返し，商が 0 になるまで p で割り続ければよい。各余りが p 進数の各桁の数字になる。

例えば，10 進数の 13 を 2 進数で表してみよう。

① 13 を 2 で割る。

$13 = 2 \times 6 + r_1$　（商は 6 で，余りは $r_1 = 1$）

② 商の 6 を 2 で割る。

$6 = 2 \times 3 + r_2$　（商は 3 で，余りは $r_2 = 0$）

③ 商の 3 を 2 で割る。

$3 = 2 \times 1 + r_3$　（商は 1 で，余りは $r_3 = 1$）

④ 商の 1 を 2 で割る。

$1 = 2 \times 0 + r_4$　（商は 0 で，余りは $r_4 = 1$）

従って，

$$(13)_{10} = (r_4 r_3 r_2 r_1)_2 = (1\ 1\ 0\ 1)_2$$

となるが，これを確認してみよう。

まず，④から $1 = r_4$ だから，③の式は

$$3 = 2 \times r_4 + r_3$$

これを②の式に代入すると

$$6 = 2 \times (2 \times r_4 + r_3) + r_2 = 2^2 \times r_4 + 2^1 \times r_3 + r_2$$

さらに，これを①の式に代入すると

$$13 = 2 \times (2^2 \times r_4 + 2^1 \times r_3 + r_2) + r_1 = 2^3 \times r_4 + 2^2 \times r_3 + 2^1 \times r_2 + r_1$$

従って，

$$13 = r_4 \times 2^3 + r_3 \times 2^2 + r_2 \times 2^1 + r_1 \times 2^0$$

となるので，

$$(13)_{10} = (r_4\, r_3\, r_2\, r_1)_2$$

上記は 2 進数の場合だが，一般の p 進数でも同じである。

● **10 進数 → 2 進数**　　$(45)_{10} = (101101)_2$

			余り
2	) 45	…	1
2	) 22	…	0
2	) 11	…	1
2	) 5	…	1
2	) 2	…	0
2	) 1	…	1
	0		

※ 商が 0 になるまで商を 2 で割り続け，余りを逆に並べる。

● **10 進数 → 8 進数**　　$(378)_{10} = (572)_8$

```
      47          5          0
8 ) 378     8 ) 47     8 ) 5
    32          40         0
    58          7          5
    56
     2
```

※ 商が 0 になるまで商を 8 で割り続け，余りを逆に並べる。

378 ÷ 8 = 47 余り 2
47 ÷ 8 = 5 余り 7
5 ÷ 8 = 0 余り 5

● 10 進数 → 16 進数　　$(451)_{10} = (1C3)_{16}$

```
      28          1          0
16 ) 451    16 ) 28    16 ) 1
     32           16         0
     131          12         1
     128
       3
```

※ 商が 0 になるまで商を 16 で割り続け，余りを逆に並べる。
余りは 16 進数で表示する。

451 ÷ 16 = 28 余り 3
28 ÷ 16 = 1 余り 12（＝C）
1 ÷ 16 = 0 余り 1

7. 最大の 2 進数

10 進数の 10000 は，0 が 4 つあるので10^4 である。逆に，10^4 は，10 進数では 10000（0 が 4 つ）である。このことは，他の基数でも同様である。

例えば，2 進数の 10000 は，10 進数では2^4 である。逆に，10 進数の2^4 は，2 進数では 10000（0 が 4 つ）である。

従って，10 進数の 256（$= 2^8 = 16^2$）などは，2 進数では 100000000（0 が 8 つ），16 進数では 100（0 が 2 つ）である。

2 進数の 10000 → 0 が 4 つ → $(10000)_2 = (2^4)_{10}$
8 進数の 10000 → 0 が 4 つ → $(10000)_8 = (8^4)_{10}$
16 進数の 10000 → 0 が 4 つ → $(10000)_{16} = (16^4)_{10}$

4 桁の最大の 2 進数は 1111 であるが，これは 10 進数ではいくらだろうか。普通に計算しても良いが，1111 の次の 2 進数は 10000（0 が 4 つ）であるので，$2^4 - 1$ に一致する。

$$(1111)_2 = (2^4 - 1)_{10} = (15)_{10}$$

8. 2 進数，8 進数，16 進数の相互変換

例えば，2 進数を 16 進数に変換する場合，2 進数を 10 進数に変換して，10 進数を 16 進数に変換しても良いが面倒である。2 進数，8 進数，16 進数の間の相互の変換には，もっと簡単な方法がある。次の表を見てみよう。

2 進数	000	001	010	011	100	101	110	111
10 進数	0	1	2	3	4	5	6	7
8 進数	0	1	2	3	4	5	6	7

2 進数	0000	0001	…	1001	1010	1011	1100	1101	1110	1111
10 進数	0	1	…	9	10	11	12	13	14	15
16 進数	0	1	…	9	A	B	C	D	E	F

これから，次のことが分かる。

- 2 進数 3 桁（000～111）は，8 進数 1 桁（0～7）に対応している。
- 2 進数 4 桁（0000～1111）は，16 進数 1 桁（0～F）に対応している。

さて，数 a に対して，a の 2 倍（$a \times 2$）は $a + a$ の意味である。$a \times 3$，$a \times 4$，… も同様である。これは，2 進数でも 16 進数でも同じである。

$$a \times 2 = a + a$$
$$a \times 3 = a + a + a$$
$$a \times 4 = a + a + a + a$$

10 進数 11 を 10 倍したら，0 が 1 つ増えて 110 になる。100（$=10^2$）倍したら，0 が 2 つ増えて 1100 になる。他の基数も全く同様である。

つまり，2 進数 11 を 2 倍したら 110，4（$=2^2$）倍したら 1100，8（$=2^3$）倍したら 11000 になる。これは，次の等式から明らかである。

$$(11)_2 \times 2 = (1 \times 2^1 + 1 \times 2^0) \times 2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (110)_2$$

ここで，2 進数を 8 倍したら 0 が 3 つ，16 倍したら 0 が 4 つ増えることに注意しよう。

$$(11)_2 \times 8 = (11)_2 \times 2^3 = (11000)_2$$

$$(11)_2 \times 16 = (11)_2 \times 2^4 = (110000)_2$$

このことから，2 進数を右から 3 桁ごとに区切れば，8 を基数とする 8 進数の表現になる。例えば，2 進数 $(abcdefg)_2$ については，次のようになる（a ～ g は，0 または 1 である）。

$$\begin{aligned}(abcdefg)_2 &= (a\,|\,bcd\,|\,efg)_2 \\ &= (a\,|\,000\,|\,000)_2 + (bcd\,|\,000)_2 + (efg)_2 \\ &= (a)_2 \times 8^2 + (bcd)_2 \times 8 + (efg)_2\end{aligned}$$

ここで，$(a)_2$，$(bcd)_2$，$(efg)_2$ のいずれも 8 進数の 1 桁になるので，$(a)_2 = (X)_8$，$(bcd)_2 = (Y)_8$，$(efg)_2 = (Z)_8$ とおくと，次が成立する。

$$(abcdefg)_2 = (X\,|\,Y\,|\,Z)_8$$

従って，2 進数は右から 3 桁ごとに区切れば，容易に 8 進数に変換できる。

① 2 進数 1101011 を 8 進数に変換したい
② 右から 3 桁ごとに区切って，3 桁の 2 進数を作る
　001 | 101 | 011
③ 3 桁の 2 進数をそれぞれ 8 進数に直す（10 進数に直すと考えても同じ）
　001 → 1，101 → 5，011 → 3
④ 従って，次のように変換される

$(1101011)_2 = (153)_8$

8 進数を 2 進数に変換するときも，同様に考えればよい。

① 8 進数 153 を 2 進数に変換したい

② 8 進数の各桁を 3 桁の 2 進数に直す（各桁を 10 進数と考えても同じ）

1 → 001，5 → 101，3 → 011

③ 従って，次のように変換される

$(153)_8 = (001101011)_2 = (1101011)_2$

以上のように，2 進数と 8 進数の相互変換では

2 進数 3 桁　⇔　8 進数 1 桁

と考えればよい。

2 進数と 16 進数の相互変換も同様であり，

2 進数 4 桁　⇔　16 進数 1 桁

と考えればよい。

<table>
<tr><td rowspan="2">2 進数
と
8 進数</td><td>2 進数 → 8 進数</td><td>8 進数 → 2 進数</td></tr>
<tr><td>右から 3 桁ごとに区切って 8 進数に直す
１０１|０１１ … 2 進数
↓　　↓
５　　３ … 8 進数 $(53)_8$</td><td>8 進数の各桁を 3 桁の 2 進数に直す
５　２　４ … 8 進数
↓　↓　↓
101 010 100 … 2 進数</td></tr>
<tr><td rowspan="2">2 進数
と
16 進数</td><td>2 進数 → 16 進数</td><td>16 進数 → 2 進数</td></tr>
<tr><td>右から 4 桁ごとに区切って 16 進数に直す
１１０１|０１１０ … 2 進数
↓　　↓
13（＝D）　６ … 16 進数（D6）$_{16}$</td><td>16 進数の各桁を 4 桁の 2 進数に直す
２　８　C … 16 進数
↓　↓　↓　(C = 12)
0010　1000　1100 … 2 進数</td></tr>
<tr><td>8 進数
と
16 進数</td><td colspan="2">いったん 2 進数に変換すればよい。8 進数 ⇔ 2 進数 ⇔ 16 進数</td></tr>
</table>

9. 10 の補数

コンピュータには，引き算を行う減算回路がない。2 進数の引き算は，補数を利用した加算で行っている。補数を使えば，加算回路と補数を作る回路だけで減算ができ，加算回路と減算回路の両方を用意する必要がないのである。

引き算を足し算で行う補数加算について，まず 10 進数でその考え方を説明する。

10 進数には，n 桁の数 A に 10^n を加えても A は変わらない，という性質がある。10^n は，$n+1$ 桁の最小の整数である。

例えば，１桁の数７に１０を加えても，１桁目の７は変わらない。

７ ＋ １０ ＝ １７

１桁目が同じであれば「＝」と表現すると，次の等式が成り立つ。

７ ＝ ７ ＋ １０

2 桁の数３５の場合は，１００を加えても３５は変わらない。ここで，１桁目と 2 桁目が同じであれば「＝」と表現すると，

３５ ＝ ３５ ＋ １００

いま，次の引き算を計算したいとする。

７ － ４

結果は１桁の数であるので，これに１０を加えても１桁目は変わらない。

７ － ４ ＝ ７ － ４ ＋ １０ ＝ ７ ＋（１０－４）＝ ７ ＋ ６

これは，引き算が足し算で計算できることを意味する。つまり，

７ － ４ ＝ ７ ＋ ６ ＝ １３ ＝ ~~1~~3（答）

ここで，

１０ － ４ ＝ ６

を，４の１０の補数という。この補数は，

４ ＋ ○ ＝ １０

を満たす○のことである。

では，2 桁の場合を考えてみる。

７５ － ４

結果は 2 桁以下の数なので，１００を加えても変わらない。

７５ － ４ ＝ ７５ － ４ ＋ １００ ＝ ７５ ＋（１００－４）

＝ ７５ ＋ ９６ ＝ １７１ ＝ ~~1~~７１（答）

ここで，

１００ － ４ ＝ ６　すなわち　４＋ ○ ＝ １００

となる数○を，４の１０の補数という。

以上のように，補数を加えれば，引き算ができることがわかる。補数は何桁で考えるかで，その値は異なる。

● ７ － ４ ＝ ７ ＋（４の１０の補数）＝ ７ ＋ ６ ＝ １３ ＝ ３

● ７５ － ４ ＝ ７ ＋（４の１０の補数）＝ ７５ ＋ ９６ ＝ １７１ ＝７１

10. １の補数と２の補数

以下，２進数の場合を考える。まず，「１の補数」と「２の補数」を形式的に覚えよう。補数計算では，桁数（ビット数）を固定して考える。以下，４ビット（４桁）で説明するが，こ

れは４ビットしか記憶できないメモリでの演算と考えればよい。

● 補数（4 ビットの場合）

(1)　4 ビットの 2 進数 A について，

① 次をみたす 4 ビットの 2 進数 B を，A の 1 の補数という。

A ＋ B ＝ １１１１　（4 ビットの最大の 2 進数で，10 進数で 2^4-1）

② 次をみたす 4 ビットの 2 進数 B を，A の 2 の補数という。

A ＋ B ＝ １００００　（5 ビットの最初の 2 進数で，10 進数で 2^4）

③ 4 ビットの 2 進数００００の 2 の補数は，００００自身と定める。

(2)　1 の補数の求め方

各ビットを反転させる（ビットの０と１を逆にする）

(3)　2 の補数の求め方

１の補数を求めて，１を加える

例えば，A ＝ ０１０１，B ＝ １０１０ のとき

A ＋ B ＝ １１１１

であるから，A の 1 の補数は B であり，B の 1 の補数は A である。

同様に，A ＝ ０１０１，B ＝ １０１１ のとき

A ＋ B ＝ １００００

であるから，A の 2 の補数は B であり，B の 2 の補数は A である。

従って，同じ補数を２回とると元に戻る。

A の 1 の補数の 1 の補数は A である

A の 2 の補数の 2 の補数は A である

１の補数は，各ビットの０と１を逆にすれば求めることができる。（ビットの０と１を逆にすることを，ビットを反転するという。）

A ＝ ０１０１ → 各ビットを反転 → １０１０　（A の 1 の補数）

２の補数は，１の補数に１を加えればよい。

１０１０ ＋ １ ＝ １０１１（A の 2 の補数）

ただし，０００００の場合，その１の補数は１１１１であるから，これに１を加えると１００００（5 桁）になってしまう。この場合は，５ビット目の１を切り捨て，０００００の２の補数は０００００自身と考える。

つまり，０００００の場合のみ，

０００００ ＋ B ＝ １００００

をみたす４桁の２進数Ｂは存在しない。そこで，０００００の２の補数は０００００自身と定義するわけである。

● **補数の求め方**（4 ビット）

```
    0   1   0   1 … 4 ビットの 2 進数 A
    ↓   ↓   ↓   ↓   （各ビットを反転）
    1   0   1   0 … A の 1 の補数
+                1   （1 を加える）
-------------------
    1   0   1   1 … A の 2 の補数
```

以上は，他のビット数の場合も同様である。例えば，8 ビットの 2 進数 A では，次の通り。

A ＋（A の 1 の補数）＝　１１１１１１１１（8 ビットであり，10 進数では 2^8-1）

A ＋（A の 2 の補数）＝ １００００００００（9 ビットであり，10 進数では 2^8）

● **補数の求め方**（8 ビット）

```
    0   0   0   0   1   1   0   1 … 8 ビットの 2 進数 A
    ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓   （各ビットを反転）
    1   1   1   1   0   0   1   0 … A の 1 の補数
+                               1   （1 を加える）
-----------------------------------
    1   1   1   1   0   0   1   1 … A の 2 の補数
```

11. ２の補数をすばやく求める方法

２の補数の求め方には次の 2 つがあるが，すばやく求めるには(2)の方が良い。

(1)　各ビットを反転して１の補数を求めて，１を加える。

(2)　最下位から最初の１までのビットは不変とし，後続の上位全ビットを反転させる。

では，(2)を説明しよう。

（例１）１０１０の２の補数を求める

①　2 桁目で初めて 1 が登場するので，そこまではビットは不変にする。

１０<u>１０</u>　（下線部のビットは変えない）

②　残りの上位ビットを反転する。

０１<u>１０</u>　（これが求める２の補数）

③　確かに次が成立する。

１０１０ ＋ ０１１０ ＝ １００００

なぜ，上記の方法で２の補数が求まるかは，次の縦書き計算で考えればわかる。

```
        1   0   1   0
+       ※   ※   ※   ※
-----------------------
```

１　０　０　０　０

（例２）２の補数を求める

○　１０１１００　→（3桁目まで不変）→　２の補数は　０１０１００

○　１０１１０１　→（1桁目のみ不変）→　２の補数は　０１００１１

12. ２の補数を利用した２進数の減算

2進数の引き算は，10進数に変換して計算できるが，コンピュータ内部では，次のように補数を加算して計算している。

● 2進数の引き算（コンピュータ内部）
n 桁の2進数A，Bについて
A － B ＝ A ＋（Bの2の補数）（(*n*＋1)桁目は捨てる）

前述の10進数の場合と同様である。4桁の場合は

● 4桁の2進数に，5桁の１００００（10進数では2^4）を加えてもその4桁は変わらない

例えば，

１１０１ ＋ １００００ ＝ １１１０１（下位からの4桁が等しければ「＝」とする）

そこで，次の4ビットの引き算を考える。

１１００ － １０１０

結果は4桁なので，これに１００００を加えてもその4桁は変わらない。従って，

１１００ － １０１０ ＝ １１００ － １０１０ ＋ １００００
＝ １１００ ＋（１００００－１０１０）
＝ １１００ ＋（１０１０の２の補数）
＝ １１００ ＋ ０１１０
＝ １００１０
＝ ~~1~~００１０　（5桁になったときは5桁目を捨てる）

すなわち

１１００－１０１０ ＝ ００１０

● 問題

１１０１０１００ － １０００ を補数加算で計算せよ。

8桁－4桁のような場合は，多い方の桁数に統一して計算する。すなわち

１１０１０１００ － １０００ ＝ １１０１０１００ － ００００１０００

＝ １１０１０１００ ＋ （００００１０００の２の補数）

＝ １１０１０１００ ＋ １１１１１０００

この足し算を縦書きで計算すると

```
        1  1  0  1  0  1  0  0
+       1  1  1  1  1  0  0  0
-------------------------------
   ~~1~~  1  1  0  0  1  1  0  0   (9桁目は捨てる)
```

従って

１１０１０１００ － １０００ ＝ １１００１１００

13. 負の数の考え方

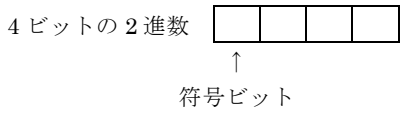
コンピュータは，すべての情報を１０１１０１１のようなビット列（2 進数）で記憶する。10 進数の 5 も 16 進数の AF も，ビット列で記憶する。一方，10 進数の世界には，負の数もある。では，負の数は，どんなビット列で記憶するのだろうか。

結論を言えば，現在のほとんどのコンピュータでは，次の方式を採用している。

負の整数は，その絶対値を表す 2 進数の 2 の補数で表す

これを 4 ビットで説明しよう。コンピュータでは，最上位ビット（先頭のビット）を「符号ビット」として使用し，次のように定める。

- 符号ビットが 0 のときは，0 または正の整数を表す
- 符号ビットが 1 のときは，負の整数を表す



そして，符号ビットが 0 のときは，残りの 3 ビットを 10 進数で表して，それに「＋」の符号を付ける。（符号ビットが 0 なので，4 ビットをそのまま 10 進数に表すと考えても同じである。）

▼ 0 または正の整数を表す 2 進数

2 進数	００００	０００１	００１０	００１１	０１００	０１０１	０１１０	０１１１
10 進数	＋0	＋1	＋2	＋3	＋4	＋5	＋6	＋7

符号ビットが 1 のときは，その 2 の補数を 10 進数で表して，「－」を付ける。

▼ 負の整数を表す 2 進数

2 進数	１０００	１００１	１０１０	１０１１	１１００	１１０１	１１１０	１１１１
10 進数	－8	－7	－6	－5	－4	－3	－2	－1

上記をまとめると，次のようになる。

① 2進数 ０１０１ を10進数で表すときは，符号ビットが0なので

０１０１ →（そのまま10進数に変換）→ ５ → ＋５

② ＋５を2進数で表すときは，符号が＋なので

＋５ →（５をそのまま2進数に変換）→ ０１０１

③ 2進数 １００１ を10進数で表すときは，符号ビットが1なので

１００１ →（2の補数）０１１１ → ７ → －７

④ －７を2進数で表すときは，絶対値7を2進数で表し，その2の補数を求める

－７ → ７ → ０１１１ →（2の補数）１００１

実際のコンピュータの内部では，固定した桁数を単位にして2進数が処理される。通常の2進数の世界では，１００１は10進数の9を意味するが，固定桁数が4桁のコンピュータでは，１００１は10進数の－7を意味することになる。

上の例で分かるように，＋7を表す2進数０１１１と，－7を表す2進数１００１は，互いに2の補数関係にある。

０１１１ ＋ １００１ ＝ １００００（10進数で$2^4=16$）

	A	B	A ＋ B
4桁の2進数	０１１１	１００１	１００００
通常の10進数	7	9	$2^4=16$
4桁のコンピュータ	＋7	－7	0

通常の10進数で言えば，足して16になる2つの数が，4桁のコンピュータの内部では，数値は同じで正と負の関係にある。従って，上の③は次のように求めることもできる。

③ 2進数１００１ を10進数で表すときは，符号ビットが1なので

１００１ →（そのまま10進数に変換）９ → ９ ＋ □ ＝ １６ になる□を求める

→ □ ＝ ７ → －７

以上の話は，他のビット数でも同様である。例えば，固定桁数が8桁のコンピュータでは，10進数－１３の2進数表現は，次のようになる。

－１３ → １３ →（8ビットの2進数で表現）００００１１０１

→（2の補数）１１１１００１１

従って，$(-13)_{10}=(11110011)_2$ となる。

14. 表現できる整数の範囲

n ビットの 2 進数で表現できる 10 進数の整数の範囲は，必ず覚えておくこと。

プログラミングの世界では，5 や 9 など，符号（＋－）がない整数を「符号なし整数」，－5 や＋9 など，符号が付いている整数を「符号付き整数」という。

n ビットの 2 進数は 2^n 個ある。符号ビットを考えなければ，n ビットの 2 進数は，0 以上の符号なし整数を表し，

最小値： 00 … 00（0 が n 個） ← 0

最大値： 11 … 11（1 が n 個） ← $2^n - 1$

従って，n ビットの 2 進数の表す整数の範囲は 0 ～ $2^n - 1$ である。

負の数を 2 の補数で表した場合，「先頭が 0 の 2 進数」と「先頭が 1 の 2 進数」は同数であり，それらの個数はともに，$2^n \div 2 = 2^{n-1}$ である。

先頭が 1 の 2 進数（負の整数）には 0 が含まれないので，その範囲は -2^{n-1} ～ -1 である。

先頭が 0 の 2 進数（0 または正の整数）には 0 が含まれるので，その範囲は，

0，1 ～ $2^{n-1} - 1$

結局，整数の範囲は -2^{n-1} ～ $2^{n-1} - 1$ であるが，8 ビットの 2 進数では次のようになる。

正の整数： 0000 0001 ～ 0111 1111 （1 ～ $2^7 - 1$）

ゼロ　　： 0000 0000

負の整数： 1000 0000 ～ 1111 1111 （-2^7 ～ -1）

● n ビットの 2 進数で表現できる整数の範囲

(1) 符号なしの整数の場合

0 ～ $2^n - 1$

(2) 符号付き整数の場合

-2^{n-1} ～ $2^{n-1} - 1$

（問題）8 ビットの 2 進数で表現される符号付き整数の範囲を求めよ。

＜解＞　上記の式に当てはめるだけである。ただし，忘れても，次のように簡単にできる。

8 ビットの 2 進数の個数は，$2^8 = 256$ である。これを 2 で割ると，

$256 \div 2 = 128$

よって，負の整数の範囲は -128 ～ -1 である。正の整数の範囲も 1 ～ 128 にしたいところだが，これだと 2 進数の個数は $128 \times 2 + 1$ になってしまう（0 があるので）。よって，正の整数の範囲は 1 ～ 127 であり，結局，求める範囲は -128 ～ 127 である。

つまり，符号付き整数の範囲は

－（半分の個数）～（半分の個数－1）