

基礎理論

データの単位

ビット... 0か1が入る箱

1ビット...  x1
 ← 1か0が入る

バイト... 8ビットの集まり

1バイト = 8ビット  x8

バイトは8個のビットの集まり、2進数では1または0が表せるので、

1バイト = 2⁸通り (256通り) の表現ができる。

1ビット : [0] もしくは [1] 2通り

2ビット : 2²通り

3ビット : 2³通り

⋮

8ビット : 2⁸通り

データの単位

テラ	T	1,000,000,000,000	10 ¹²
ギガ	G	1,000,000,000	10 ⁹
メガ	M	1,000,000	10 ⁶
キロ	K	1,000	10 ³
基準 バイト	byte	1	1
ミリ	m	0.001	10 ⁻³
マイクロ	μ	0.000001	10 ⁻⁶
ナノ	n	0.000000001	10 ⁻⁹
ピコ	p	0.000000000001	10 ⁻¹²

1GB = 10億 バイト
↳ = 80億 ビット

2進数・16進数

2進数: 0と1だけを表記され、1まづ行ったら1つ桁が上がる。

10進数	0	1	2	3	4	5	6	7	...
2進数	0	1	10	11	100	101	110	111	...

16進数: 0~9までは10進数と同様に表記、10~15までをA~Fで表記

10進数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19...
16進数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13...

例: (701)₁₀ を 2進数と16進数で表せ

2進数: 1010111101

16進数: 2BD

2) 701 ... 1	16) 701 ... 13 → D
2) 350 ... 0	16) 43 ... 11 → B
2) 175 ... 1	2
2) 87 ... 1	
2) 43 ... 1	
2) 21 ... 1	
2) 10 ... 0	
2) 5 ... 1	
2) 2 ... 0	
2) 1 ... 1	

~進数を表すとき、

2進数の値 10^x (10)₂ ✓

10進数の値 10^x (10)₁₀ ✓

「ただ書くだけ」何進数か
区別できないから()を付けた書く

小数

- ・有限小数 ... 小数部の桁数に限りがある小数
- ・無限小数 ... 小数部の桁数に限りがないう小数 (円周率 など ...)
- ・循環小数 ... 同じ文字が無限に繰り返される小数

10進数と同様に、2進数、16進数にも小数が存在する。
2進数小数、16進数小数

▷ 基数

基数とは桁上かつの基準となる数

～進数から～進数へ変換することを基数変換という

2進数の基数に2、

10進数の基数は10、

16進数の基数に16

▷ 基数変換

出題例: 「2進数を10進数に変換」だけでなく「2進数から16進数」や「10進数から16進数」など、さまざまなパターンが存在する

2進数から16進数や、16進数から2進数に変換する際は、
一度、10進数に変換して解く。

2進数 $\longleftrightarrow$ 10進数 $\longleftrightarrow$ 16進数

・2進数 $\rightarrow$ 10進数に変換

2進数 ... 1 0 0 1 . 0 0 1 例) $\frac{1}{8} = 0.125$

桁の重み ... 2^3 2^2 2^1 1 $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$

$8 + 0 + 0 + 1$. $0 + 0 + \frac{1}{8}$ $\rightarrow (9.125)_{10}$

↑ 小数のときは分数となる

・16進数 $\rightarrow$ 10進数に変換

16進数 ... (2)₁₆ (B)₁₆ (D)₁₆ . (C)₁₆ (8)₁₆ $\frac{12}{16} = 0.75$

10進数 ... (2)₁₀ (11)₁₀ (13)₁₀ . (12)₁₀ (8)₁₀ $\frac{8}{256} = 0.03125$

桁の重み ... 256 16 16 1 $\frac{1}{16}$ $\frac{1}{256}$

$512 + 176 + 13$. $12/16 + 8/256$

$(701.78125)_{10}$

10進数 → 2進数に変換

例: $(18)_{10}$ を 2進数に変換

10進数を2進数に変換するときは、ひたすら2で割る
商が「1」になるまで割ったら余を下から並べる

$$\begin{array}{r} 2 \overline{) 18} \dots 0 \\ 2 \overline{) 9} \dots 1 \\ 2 \overline{) 4} \dots 0 \\ 2 \overline{) 2} \dots 0 \end{array}$$

↑ 10010

$$(18)_{10} \rightarrow (100110)_2$$

小数部の計算 10進数小数を2進数小数に変換

例: $(0.625)_{10}$ を 2進数小数に変換

・解き方

10進数の小数部だけをとり出し、ひたすら2を掛ける。積のうち整数部は取り出して残りの小数部は再び掛ける。

$$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \end{array} \rightarrow \begin{array}{r} 0.25 \\ \times 2 \\ \hline 0.50 \end{array} \rightarrow \begin{array}{r} 0.5 \\ \times 2 \\ \hline 1.0 \end{array}$$

整数は取り出した順に並べる

$$0.101$$

注意: 2進数小数なので 0.~と続くように書く

10進数小数から2進数小数に変換した際、無限小数になることがある

10進数 → 16進数に変換

解法、要領は2進数の時と同じ

(701)₁₀ を16進数に変換

$$\begin{array}{r}
 16 \overline{) 701} \dots 13 \rightarrow D \\
 16 \overline{) 43} \dots 11 \rightarrow B \\
 \hline
 \phantom{16 \overline{) 43}} 2
 \end{array}
 \Rightarrow \underline{2BD}$$

整数を10進数に変換して、2進数の時と同様に下から余りを並べる

10進数小数 → 16進数小数に変換

例: (0.78125)₁₀ を16進数に変換

$$\begin{array}{r}
 0.78125 \\
 \times 16 \\
 \hline
 5468750 \\
 78125 \\
 \hline
 12.50000
 \end{array}
 \quad
 \begin{array}{r}
 0.5 \\
 \times 16 \\
 \hline
 30 \\
 5 \\
 \hline
 8.0
 \end{array}$$

(12)₁₀ → (C)₁₆
(8)₁₀ → (8)₁₆

16進数小数なので (12)₁₀ は (C)₁₆ になる

0.C8

2進数と16進数の基数変換の特徴

2進数の有限小数は、10進数にしても必ず有限小数になる

10進数の有限小数は、2進数になると無限小数になることもある

負の2進数

2進数で負の整数の表現

絶対値表現

ビット列の先頭の1ビットで正負を表し、残りのビットで値を表す方法。

先頭の1ビットが 0 = 正 先頭の1ビットが 1 = 負

先頭1ビットが正負を表すために必ずビットを符号ビットとし

例: 10進数「2」の正負を8ビットの2進数で表現

$$\begin{array}{rcl}
 10進数 & & \\
 +2 & \longrightarrow & 0000010 \\
 -2 & \longrightarrow & 1000010
 \end{array}$$

0は正
1は負の数

下位桁は同じ
(2)₁₀ → (10)₂

・10の補数表現

すべてのビットを反転させて、正負を表現する

例: $(2)_{10}$ の正負を8ビットの2進数で表現

10進数	→	2進数
+2	→	0000010
-2	→	1111101

↓ 反転
先頭1ビットは符号ビット

・2の補数表現

すべてのビットを反転した後、1を足して正負を表す

例: $(2)_{10}$ の正負を8ビットの2進数で表現

10進数	→	2進数
+2	→	0000010
-2	→	1111101

↓ ビットを反転
↓ 1を足す
先頭1ビットは符号ビット

2進数の四則演算

・2進数の足し算、引き算

2進数の足し算は足して2になったら繰り上がる

例: $10 + 1 = 11$ $1 + 1 + 1 = 11$

$1 + 1 = 10$ ← 10進数では $1+1=2$ なので、2進数では 10 となる

問題、次の2進数を計算せよ

10100011

+ 10010111

100111010 ← $1+1=10$ で桁が繰り上がる

↑ ↑ 繰り上がりで $1+1+1=11$

↑ $1+0+1=10$

$1+0+0=1$

ひたすら繰り上がる。

2進数の式を10進数に変換して計算

それを2進数に再度変換してもOK

・2進数の引き算

例: $(10)_2 - (1)_2 = (1)_2$

$(2)_{10} - (1)_{10} = (1)_{10}$

桁借り
$$\begin{array}{r} 10 \\ - 1 \\ \hline 1 \end{array}$$

2進数の引き算は前の桁を借りて計算する

・2進数の引き算 (別の方法)

例: $(76)_{10} - (9)_{10}$ の計算

まず式の値を2進数に変換 << 前提として、8ビットの2進数、先頭ビットが符号

$$\begin{array}{r} (76)_{10} \quad - \quad (9)_{10} \\ (01001100)_2 \quad - \quad (00001001)_2 \end{array}$$

次に $(9)_{10}$ をマイナスにする << 引く値をマイナスに2引き値にする

??では2の補数表現を使う

$$\begin{array}{ccc} 00001001 & \rightarrow & 11110110 & \rightarrow & 11110111 \\ & \text{1の補数表現} & & \text{2の補数表現} & \\ & \text{反転させて} & & \text{更に1を足して} & \end{array}$$

2の補数が求まったら先の $(76)_{10}$ と足す

$$\begin{array}{r} 01001100 \quad (76)_{10} \\ + 11110111 \quad (-9)_{10} \\ \hline 01000011 \end{array}$$

前提として、8ビットの2進数で表現おとしたので、
101000011 9桁目に来ている1は消お。
よって答えは 01000011 となる。

出題の際、引く値を絶対値表現で見えるのか、1の補数、2の補数で表すのかは、書いてある。

・シフト演算

ビットを左右にずらして掛け算や割り算を行う計算方法

論理シフト: 正負を考慮しない (符号ビットを考慮しない)

算術シフト: 正負を考慮する (符号ビットを考慮する)

符号ビットを考慮して計算するのか、そうでないかは問題文によって指定される
問題文に応じて、論理シフトか算術シフトか考える。

・論理シフト

シフト演算には左シフトと右シフトがある。

左シフト: 掛け算、ビット列を n 個分だけ左にずらすと、元の数の 2^n 倍になる

右シフト: 割り算、ビット列を n 個分だけ右にずらすと、元の数の $(\frac{1}{2})^n$ 倍

左シフト 例: $(2)_{10}$ を 2倍 前提 8ビット

2進数 00000010 10進数 2

000000100 $\rightarrow$ 0000100 = 4

↑空いたビットには0を入れる

今回、8ビットが前提なので、それを越えた桁は消す

右シフト 例: $(4)_{10}$ を $\frac{1}{2}$ 倍 前提 8ビット

2進数 00000100 10進数 4

00000010 $\leftarrow$ 越えたビットは消す

空いているビットには0

$(0000010)_2 = (2)_{10}$

・オーバーフロー

シフト演算で前提として定めたビット数を越えて、そのビットは消される。

2の消されたビットが、10進数の計算結果と異なる場合、処理に失敗している。

このことをオーバーフローという。

・算術シフト

算術シフトは論理シフトと同じ要領で解く、ただし、正負を考慮する。

先頭の符号ビットは固定したままビットをずらす。

左シフトの例 値を4倍する

2進数 (2の補数表現) 11110100 10進数 -12

11010000 $\rightarrow$ -48

↑符号ビット

↑空きのビットには0

4倍なので2つずらす

越えたビットは消す

右シフト 値を $\frac{1}{4}$ 倍する

2進数 (2の補数表現) 11110100 10進数 -12

11110100 $\rightarrow$ -3

↑符号ビット以外を n 倍分をずらす

↑1を入れる

符号ビットはそのまま

例題: 10進数の演算式 $7 \div 32$ の結果を2進数で表せ

まずは7を2進数に

$$\begin{array}{r} 2) 7 \dots 1 \\ 2) 3 \dots 1 \\ 1 \end{array} \quad (7)_{10} = (111)_2$$

$$\begin{array}{r} 111.000000 \\ 000.00111 \end{array}$$

5つずらす

答え 0.00111_2

次に32が2の何乗なのか求めたい

$$\begin{array}{r} 2) 32 = 2^5 \\ 2) 16 \\ 2) 8 \\ 2) 4 \\ 2) 2 \end{array}$$

$(\frac{1}{2})^5$
割り算なので分数

トシフト演算と足し算を組み合わせた問題

例: 10進数の演算式 3×9 の結果を2進数で表せ

9倍は表現できないので、一担8を使う
8は2の冪乗で表せる。

$$\begin{array}{r} 2) 3 \dots 1 \\ 1 \end{array} \quad (11)_2$$

$$\begin{array}{r} 11 \\ 11000 \\ ++ \\ 11 \\ \hline 11011 \end{array}$$

3乗なので3つずらせる

8倍 (2^3)₁₀

9倍

8倍 (2^3)₁₀

倍数は何度も同じ数を足し続けること。
なので8倍した数に、もう一度同じ数を足すことで
9倍になる!

$$\begin{array}{r} 11011 \\ \times \quad \times \quad \times \quad \times \quad \times \\ 168421 \end{array}$$

$$168 + 2 + 1 = 27$$

3×9と等しい!

$$11011$$

細かく倍にしていくときは、この冪乗に足りない分を足す

過去問

数値を2進数で格納するレジスタがある。このレジスタに正の整数 x を設定した後、レジスタの値を2ビット左にシフトして x を加える操作を行って、レジスタの値は x の何倍になるか。このとき、オーバーフローは発生しないとする。

レジスタの値を2ビット左にシフトして x を加える。

2ビット左にシフトなので

$$x \times 2^2 + x \leftarrow \text{ここに} x \text{を加える}$$

x の値を14にしたら、5で割る、よって x の5倍

$$\text{答え: } x \times 2^2 + x$$

基礎理論

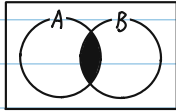
集合と論理演算

数学Aの復習

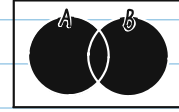
集合の関係を視覚的に表した物としてベン図がある

例:

$A \cap B$ 積集合



$A \cup B$ 和集合



$\bar{A}$ 補集合

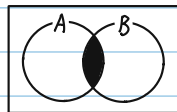


論理演算

2進数の1桁2のみを対象にした演算方式。四則演算の演算子は+、-、×、÷

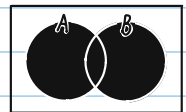
論理演算ではAND、OR、XOR、NOT

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



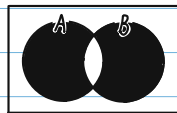
AND演算
論理積

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



OR演算
論理和

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



XOR演算
排他的論理和

A	NOT A
0	1
1	0



NOT演算
否定

AND ... 1と1以外は0、1と1で1

OR ... 1が含まれていたら1

XOR ... 2つの信号(0と0、1と1)では0、それ以外は1

NOT ... 信号を反転させる。

NANDとNOR 優能演算子

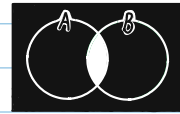
NAND ... ANDとNOTを組み合わせたもの。ANDの演算結果も反転

NOR ... NANDと同様に ORにNOTを組み合わせたもの。ORの結果を反転

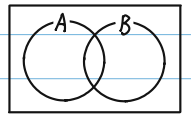
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



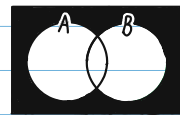
否定論理積



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



否定論理和

論理演算子の優先順位

括弧 > NOT > AND > OR

例: A OR B AND C を演算せよ

優先順位で"はまず" B AND C を演算し、その結果と OR する。

過去問

XとYの否定論理積 X NAND Y は、NOT(X AND Y)として定義される

X OR Y を NAND だけを使って表した論理式はどれか?

A. ((X NAND Y) NAND X) NAND Y

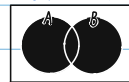
解くときはベン図を使って考える

1. (X NAND X) NAND (Y NAND Y)

ウ. (X NAND Y) NAND (X NAND Y)

X OR Y を NAND だけで

エ. X NAND (Y NAND (X NAND Y))



を作る

-Pのベン図

まず (X NAND Y) のベン図を描く

NANDは1と1で0になる



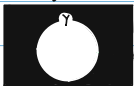
この結果からAはX

(X NAND Y) と X を NAND する

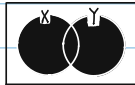
-1のペンで

まず $(X \text{ NAND } Y)$ これは単に NOT と同じなので



$(Y \text{ NAND } Y)$ も同様  これを NAND すると

14 1で0、それ以外は1となる



これは OR と同じなので

答えは 1

・X及びYはそれぞれ0又は1の値をとる変数である。 $X \square Y$ をXとYの論理演算としたとき、次の真理値表が与えられた。 $X \square Y$ の真理値表はどれか。

X	Y	X AND (X $\square$ Y)	X OR (X $\square$ Y)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

ここが共通していることは $X \text{ AND } (X \square Y)$ であり、
AND と OR を出して13と15のこと。

まず $X \text{ AND}$ したとき 0で OR したときは 1である

よってXが0、Yが0のときの $X \square Y$ は 1であることが分かる

X	Y	(X $\square$ Y)
0	0	1

次に、 $X \text{ AND}$ して0、 $X \text{ OR}$ して1であり、Xは0、

Yは1である。したがって式の中には、Yは出ないので、単に

$X \square Y$ は 1であるとなる。 $\ll X \square Y$ と AND、OR するのは、
Yは関係ない

X	Y	(X $\square$ Y)
0	0	1
0	1	1

Xの値が1で、 $X \square Y$ と AND したときは、0で、OR した
ときは、1が結果になる。

論理演算のルールに則り、211とANDは1と1で

しか1を出力しない。ORは0と0以外が1、よって $X \square Y$ は0

X	Y	(X $\square$ Y)
0	0	1
0	1	1
1	0	0

最後に AND をすると、1で OR をすると、1、そして、Xの値は1、
ANDは1と1で1を出力するので、 $X \square Y$ は1となる。

X	Y	(X $\square$ Y)
0	0	1
0	1	1
1	0	0
1	1	1