

# プログラミング練習 型

吉林

## ・そもそも型とは

メモリに配置された 0 と 1 の羅列のデータをどのようにコンピュータが解釈するかコンピュータに指示するもの

型	キーワード	使用ビット数
文字データ	char	8 ビット
符号付き整数	int	16 ビット
浮動小数点数	float	32 ビット
倍精度浮動小数点数	double	64 ビット
値なし	void	なし

[c\\_Intro/c\\_pro03.md at main · acoytaka/c\\_Intro · GitHub](#) より引用

## ・ビットとは

メモリにはビット (bit) と呼ばれる小さな箱が 8 の倍数で、縦にも横にも並んでいる。この箱には 0 または 1 を格納することが出来、メモリはデータを記録することが出来る。書き換えることも可能。

実際は以下の例 (16bit 幅) の様に 0 または 1 がびっしり格納されている。

← 16bit 幅 →															
0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0
:															
0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0

変数の宣言を行うとこのメモリ領域のある部分に名前が付けられ、宣言以降名前でもメモリにアクセスすることが可能になる。

## ・ char 型

文字データを格納するための型。1 つの文字を記憶するために 8bit(=1 バイト)のメモリを消費する型。文字は'A'の様にシングルコーテーションで囲って表現する。(char 型単体には複数文字列は格納できない)

文字もメモリには数値、0 と 1 で格納されるため、int 型等で解釈すれば数値としても利用できる。よって文字と数値の対応が決められている。C 言語では ASCII コードが採用されている。以下に ASCII コード (一部抜粋) を示す。

Char	ASCII	Decimal	Bits	Char	ASCII	Decimal	Bits	Char	ASCII	Decimal	Bits
0	48	0	000000	F	70	22	010110	d	100	44	101100
1	49	1	000001	G	71	23	010111	e	101	45	101101
2	50	2	000010	H	72	24	011000	f	102	46	101110
3	51	3	000011	I	73	25	011001	g	103	47	101111
4	52	4	000100	J	74	26	011010	h	104	48	110000
5	53	5	000101	K	75	27	011011	i	105	49	110001
6	54	6	000110	L	76	28	011100	j	106	50	110010
7	55	7	000111	M	77	29	011101	k	107	51	110011
8	56	8	001000	N	78	30	011110	l	108	52	110100
9	57	9	001001	O	79	31	011111	m	109	53	110101
:	58	10	001010	P	80	32	100000	n	110	54	110110
;	59	11	001011	Q	81	33	100001	o	111	55	110111
<	60	12	001100	R	82	34	100010	p	112	56	111000
=	61	13	001101	S	83	35	100011	q	113	57	111001
>	62	14	001110	T	84	36	100100	r	114	58	111010
?	63	15	001111	U	85	37	100101	s	115	59	111011
@	64	16	010000	V	86	38	100110	t	116	60	111100
A	65	17	010001	W	87	39	100111	u	117	61	111101
B	66	18	010010	'	96	40	101000	v	118	62	111110
C	67	19	010011	a	97	41	101001	w	119	63	111111
D	68	20	010100	b	98	42	101010				
E	69	21	010101	c	99	43	101011				

<https://www.researchgate.net/profile/DidiIstardi/publication/322538143/figure/tbl1/AS:667701353656321@1536203816209/Char-Data-conversion-ASCII-Decimal-and-Biner.png> より

## ・ int 型

数値の記憶に最もよく使われる型。16bit。正の数であれば、10 進数の数字をそのまま 2 進数としてメモリに格納する。逆に解釈するときはメモリに角煮字されている 0 と 1 の羅列 (2 進数) は 10 進数として解釈される。

10 進数の負の数を格納するときは「2 の補数」というテクニックでメモリに格納する。最上位ビットは符号ビットと呼ばれ、最上ビットが 0 であれば正の数、1 であれば負の数である。2 の補数に関しては以下を参照のこと。

[2 の補数とは？2 の補数の計算方法と表現範囲をわかりやすく解説！1 の補数との違いは？C 言語での補数計算プログラムもチェック | IT フリーランスエンジニア案件なら A-STAR \(エースター\) \(agency-star.co.jp\)](#)

## ・ float 型 / double 型

浮動小数点を持つ数値のためのデータ型、つまり小数が扱える。float 型は 32bit。Double 型は 64bit。これらの数値の格納方法は複雑なため、ここでは割愛する。というか俺もわからん。

## ・ void 型

「どんなデータであるか（サイズや扱われ方）」が定義されていない。関数の引数や関数の戻り値が「無い」ことを明示的に示すことができる。用途はこのくらいしかない。

## 練習問題

### ① ふたつの整数値を読み込んでその平均値を求める。

以下の実行結果を示すプログラムを書いてみよう。(参考書 List2-5) (A と B は入力)

---

ふたつの整数を入力せよ。  
整数 A:500  
整数 B:10  
それらの平均は 255 です。

---

### ② 実数値の四則演算

以下の実行結果を示すプログラムを書いてみよう。(参考書 List2-7)  
(X と Y は入力) (小数第二位まで表示)

---

ふたつの値を入力せよ。  
実数 X : 40.5  
実数 Y : 5.2  
X + Y = 45.70  
X - Y = 35.30  
X \* Y = 210.60  
X / Y = 7.79

---

※小数点第 **n** 位まで表示したいときには printf で **%.**n**f** と書く。

### ③ 整数値を読み込んで平均を実数値で表示

以下の実行結果を示すプログラムを書いてみよう。(参考書 List2-10)

ヒント：型変換（以下サイトの「明示的な型変換（キャスト）」）

(int で宣言、double で出力すること。A・B・C は入力)

[型変換\(C言語\) - 超初心者向けプログラミング入門 \(pc-note.net\)](http://pc-note.net)

---

ふたつの整数を入力せよ。

整数 A : 40

整数 B : 45

それらの平均は 42.50 です。

---

3-2 (List2-11)

---

3つの整数を入力せよ。

整数 A : 87

整数 B : 45

整数 C : 59

それらの合計は 191 です。

それらの平均は 63.7 です。

---