

C 言語検定必勝プリント 明日のために 3 級編

No.02 文字を数字で扱う char 型

昨日は int と float 型を覚えた。今日は char 型の変数を考えてみる。その前にそもそもコンピュータは文字が分かるのかを考えると……

答えは「**コンピュータに文字は分からない**」となります。

コンピュータを使ってこの文章書いていますし、メールも読める。コンピュータは文字が分かるに決まってるじゃんという方もいらっしゃると思いますが実は本当にコンピュータは字が分からないのです。日本語だけじゃありません、アルファベットも数字も全く分からないのです。

じゃあどうやっているかというと、人間がひとつひとつ文字に番号をつけて組み合わせているだけなのです。その文字に割り振った番号を「文字コード」といいます。実は char 型の変数には数字しか入りません。文字コード番号が入るのです。

なぜ int 型とかと一緒にしないのかというと int 型は CPU の形式（アーキテクチャって言います）によって大きさが変化しますが、文字数は変わらないため、アーキテクチャが変わっても変数の大きさを変える必要がありません。

たとえば数字は 0～9 までの 10 文字ですし、アルファベットも a～z（小文字）、A～Z（大文字）で $26 \times 2 = 52$ 文字です。日本語はちと多くて 3 千文字くらいですがこれも数がそんなに増減するものでもないです。また、もともとアルファベットと数字を表現できれば十分だったので 1 バイト（=256 通り）もあれば文字を表現できます。

そのため int 型で文字を表現するには無駄が多すぎるため、文字専用の変数の型である char 型を用意したのです。

面倒くさそうですが実はこれ結構便利だったりします。ちなみに文字の足し算をしてみるとこんなことができます

文字コードの足し算

'A' + 1 ⇒ 'B'
'A' + 32 ⇒ 'a'

今は「ふーん」って感じですがこれが後になって響いてきます。最重要ポイントね。

日本語の表示は 1 バイト（256 通り）では文字が表示できないので文字 2 つ分（2 バイト 65536 通り：正確にはちょっと違う）で 1 文字を表現するようにしました。これを 2 バイト文字と呼びます。

要領が同じ場合、扱える文字数は日本語だと英数字の半分になってしまうのです。

また、文字は複数組み合わせる場合があり、これを文字列と言います。

文字列を利用しない場合、char 型の変数だけでやろうとすると文字の数だけ変数が必要になってしまいます。

試しに実際やってみましょうか。

例) Hellow を char 型変数に入れてみる

char a1='H', a2='e', a3='l', a4='l', a5='o', a6='w' こりは大変・・・

こんな状態では大変ですので、配列という仕組みを使って文字列を扱うことになっています。

本当に分からないんです。

※文字コード：別紙参照

前回やりましたね

だってコンピュータはアメリカ生まれですもん。

それと、命令する場合日本語よりも英語のほうがシンプルなため。

ちなみに、聞いた話では外国では英語もしゃべれない日本人がなぜプログラミングできるか分からない人も多いそうです。

この辺検定に良く出ます。

（でも次に必ず出るとは限りません）

※toupper/tolower 関数の仕組みです。

半角文字：1 バイト 英数文字
全角文字：2 バイト 日本語とか

ここは C を学ぶ上で大切な部分です。

特別付録 ASCII 文字コード表（アルファベットだけね）

文字	コード			文字	コード			文字	コード			文字	コード		
	10 進	8 進	16 進		10 進	8 進	16 進		10 進	8 進	16 進		10 進	8 進	16 進
NUL	0	00	0x00	SP	32	040	0x20	@	64	0100	0x40	`	96	0140	0x60
SOH	1	01	0x01	!	33	041	0x21	A	65	0101	0x41	a	97	0141	0x61
STX	2	02	0x02	"	34	042	0x22	B	66	0102	0x42	b	98	0142	0x62
ETX	3	03	0x03	#	35	043	0x23	C	67	0103	0x43	c	99	0143	0x63
EOT	4	04	0x04	\$	36	044	0x24	D	68	0104	0x44	d	100	0144	0x64
ENQ	5	05	0x05	%	37	045	0x25	E	69	0105	0x45	e	101	0145	0x65
ACK	6	06	0x06	&	38	046	0x26	F	70	0106	0x46	f	102	0146	0x66
BEL	7	07	0x07	'	39	047	0x27	G	71	0107	0x47	g	103	0147	0x67
BS	8	010	0x08	(40	050	0x28	H	72	0110	0x48	h	104	0150	0x68
HT	9	011	0x09)	41	051	0x29	I	73	0111	0x49	i	105	0151	0x69
NL*	10	012	0x0a	*	42	052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
VT	11	013	0x0b	+	43	053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
NP	12	014	0x0c	,	44	054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
CR	13	015	0x0d	-	45	055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
SO	14	016	0x0e	.	46	056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
SI	15	017	0x0f	/	47	057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
DLE	16	020	0x10	0	48	060	0x30	P	80	0120	0x50	p	112	0160	0x70
DC1	17	021	0x11	1	49	061	0x31	Q	81	0121	0x51	q	113	0161	0x71
DC2	18	022	0x12	2	50	062	0x32	R	82	0122	0x52	r	114	0162	0x72
DC3	19	023	0x13	3	51	063	0x33	S	83	0123	0x53	s	115	0163	0x73
DC4	20	024	0x14	4	52	064	0x34	T	84	0124	0x54	t	116	0164	0x74
NAK	21	025	0x15	5	53	065	0x35	U	85	0125	0x55	u	117	0165	0x75
SYN	22	026	0x16	6	54	066	0x36	V	86	0126	0x56	v	118	0166	0x76
ETB	23	027	0x17	7	55	067	0x37	W	87	0127	0x57	w	119	0167	0x77
CAN	24	030	0x18	8	56	070	0x38	X	88	0130	0x58	x	120	0170	0x78
EM	25	031	0x19	9	57	071	0x39	Y	89	0131	0x59	y	121	0171	0x79
SUB	26	032	0x1a	:	58	072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
ESC	27	033	0x1b	;	59	073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
FS	28	034	0x1c	<	60	074	0x3c	¥	92	0134	0x5c		124	0174	0x7c
GS	29	035	0x1d	=	61	075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
RS	30	036	0x1e	>	62	076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
US	31	037	0x1f	?	63	077	0x3f	_	95	0137	0x5f	DEL	127	0177	0x7f

C 言語検定必勝プリント確認問題 明日のために 3 級編

No.02 文字を数字で扱う char 型

問 1 次の設問を読んで、括弧の中に当てはまる語句を a,b,h は解答群より c~g は解答群へ数字を記入しなさい。

コンピュータ上で文字を扱うためには (a) を利用します。これは文字を理解できないコンピュータ上で文字を扱う場合、人間が文字に番号を割り振り、コンピュータの持つデータとその文字についての番号「(a)」によりコンピュータ上で文字を扱えるようにしたものです。

C の変数の型に文字を扱うための char 型があります。実はこれ、文字自体がこの変数の中に代入されるのではなく文字コード番号という (b) が代入されます。たとえば 'A' というアルファベットの文字コードが 10 進数で 65 だった場合次のような計算ができます。

'A' + 1 ⇒ 'B' ('B'の文字コードは 66)

このように文字コードという数字に対しての計算が可能です。また特定の文字が大文字であるかどうかはコードの大小で判別できます、例えば 'A' の文字コードが 65、'a' の文字コードが 97 だった場合、文字コードが 65 から (c) の間の場合大文字、97 から (d) の場合小文字となります。

またこの仕組みを使えば、大文字のコードに (e) を足せば小文字へ変換できますし、小文字のコードから (e) を引けば大文字へ簡単に変換できます。

今まで話してきた内容はあくまで英語圏（英数文字）での話です。char 型の変数は大きさが実は 1 バイト ((f) 通り) です。そのため文字を割り振ろうとした場合、英数文字だけであれば 0~9、A~Z、a~z で (g) 個、さらにいくつかの記号をあわせても十分対応できるのですが、日本語の場合それだけでは全く足りません。そのため char 型の変数を 2 つ組み合わせて表現します。そのため日本語の文字コードは (h) 文字と呼ばれます。

【解答群 1 a,b,h】

ア) 電気コード

イ) 文字コード

ウ) ミンクコート

エ) 記号

オ) 数字

カ) ルーン

キ) 2 バイト

a		b		H	
---	--	---	--	---	--

c		d		e	
f		g			

問 2 次の処理結果を解答欄に書きなさい

1) printf(" A+C-60 = %c¥n", 'A' + 'C' -60);

1)	
----	--

2) printf(" 'a' - 32 = %c¥n", 'a' - 32);

2)	
----	--

C 言語検定必勝プリント確認解答 明日のために 3 級編

No.02 文字を数字で扱う char 型

問 1 次の設問を読んで、括弧の中に当てはまる語句を a,b,h は解答群より c~g は解答群へ数字を記入しなさい。

コンピュータ上で文字を扱うためには (a) を利用します。これは文字を理解できないコンピュータ上で文字を扱う場合、人間が文字に番号を割り振り、コンピュータの持つデータとその文字についての番号「(a)」によりコンピュータ上で文字を扱えるようにしたものです。

C の変数の型に文字を扱うための char 型があります。実はこれ、文字自体がこの変数の中に代入されるのではなく文字コード番号という (b) が代入されます。たとえば 'A' というアルファベットの文字コードが 10 進数で 65 だった場合次のような計算ができます。

'A' + 1 ⇒ 'B' ('B'の文字コードは 66)

このように文字コードという数字に対しての計算が可能です。また特定の文字が大文字であるかどうかはコードの大小で判別できます、例えば 'A' の文字コードが 65、'a' の文字コードが 97 だった場合、文字コードが 65 から (c) の間の場合大文字、97 から (d) の場合小文字となります。

またこの仕組みを使えば、大文字のコードに (e) を足せば小文字へ変換できますし、小文字のコードから (e) を引けば大文字へ簡単に変換できます。

今まで話してきた内容はあくまで英語圏（英数文字）での話です。char 型の変数は大きさが実は 1 バイト ((f) 通り) です。そのため文字を割り振ろうとした場合、英数文字だけであれば 0~9、A~Z、a~z で (g) 個、さらにいくつかの記号をあわせても十分対応できるのですが、日本語の場合それだけでは全く足りません。そのため char 型の変数を 2 つ組み合わせて表現します。そのため日本語の文字コードは (h) 文字と呼ばれます。

【解答群 1 a,b,f】

ア) 電気コード

イ) 文字コード

ウ) ミンクコート

エ) 記号

オ) 数字

カ) ルーン

キ) 2 バイト

a	イ	b	オ	h	キ
---	---	---	---	---	---

c	90	d	122	e	32
f	256	g	62		

問 2 次の処理結果を解答欄に書きなさい

3) printf(" A+C-60 = %c¥n", 'A' + 'C' -60);

1)	H
----	---

4) printf(" 'a' - 32 = %c¥n", 'a' - 32);

2)	A
----	---