11. 文字型

C 言語でよく使う変数の型には以下のようなものがある。

型宣言名	データの型	扱える値	サイズ
(long) int	倍長整数型	-2147483648 〜 2147483647	4 バイト
float	実数型	$\pm 3.4 \times 10^{-38}$〜 $\pm 3.4 \times 10^{38}$ (有効桁は 7 桁)	4 バイト
double	倍精度実数型	$\pm 1.7 \times 10^{-308}$〜 $\pm 1.7 \times 10^{308}$ (有効桁は 15 桁)	8 バイト
char	文字型	-128 〜 127	1 バイト

例：

```
char ch;      /* 宣言 */
ch = 'A';
```

で ch という文字型変数に文字 A を代入することになる。
このあと試しに、

```
printf("%c = %d\n", ch, ch);
```

を実行すると「A = 65」と出力されるが、これは文字 A はコンピュータの内部では 65(二進数で 1000001) という値で表されていることを示している。
多くの場合、コンピュータで扱う文字は、ASCII(アスキー) コードという 0 から 127 までの整数値に対応づけられている。

コード	文字	コード	文字	コード	文字	コード	文字
0~ 8	（説明略）	53	5	78	N	103	g
9	水平タブ	54	6	79	O	104	h
10	改行	55	7	80	P	105	i
11~ 31	（説明略）	56	8	81	Q	106	j
32	空白	57	9	82	R	107	k
33	!	58	:	83	S	108	l
34	“	59	;	84	T	109	m
35	#	60	<	85	U	110	n
36	$	61	=	86	V	111	o
37	%	62	>	87	W	112	p
38	&	63	?	88	X	113	q
39	‘	64	@	89	Y	114	r
40	(	65	A	90	Z	115	s
41	)	66	B	91	[	116	t
42	*	67	C	92	¥	117	u
43	+	68	D	93	]	118	v
44	,	69	E	94	^	119	w
45	-	70	F	95	_	120	x
46	.	71	G	96	`	121	y
47	/	72	H	97	a	122	z
48	0	73	I	98	b	123	{
49	1	74	J	99	c	124	|
50	2	75	K	100	d	125	}
51	3	76	L	101	e	126	~
52	4	77	M	102	f	127	削除

この対応において重要なことは、

文字	0	1	2	3	4	5	6	7	8	9
ASCII コード	48	49	50	51	52	53	54	55	56	57

文字	A	B	C	D	…	Y	Z
ASCII コード	65	66	67	68	…	89	90

文字	a	b	c	d	…	y	z
ASCII コード	97	98	99	100	…	121	122

のように 0〜9、A〜Z、a〜z の間は ASCII コードが連続して対応づけられていることである。

ch の値を出力するには、 `printf("%c", ch);` でもよいが、

```
putchar(ch);
```

の方が簡便である。なお、`putchar('G');` とすると G がそのまま出力される。

[例題 1] 次のプログラムを実行すると何が出力されるか考えなさい。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    int i;
 5    char ch = 'A';
 6
 7    for (i=0; i<=25; i++) {
 8      putchar(ch + i);
 9      putchar(' ');
10    }
11  }
```

1 文字単位の入力

キーボードから 1 文字を整数型変数[1]ch に読み込むには `scanf(.....);` でなく、

```
ch = getchar();
```

を使う。

[例題 2] キーボードからの入力を . (ピリオド) が押されるまで 1 文字ずつ読み込んで、そのまま出力するプログラムをつくりなさい。

[1]当初、char 型は正数のみを取り扱っていた。そこで、負の値を取ることがある getchar の型は、int 型とした。このような歴史的な理由により、ch は int 型で宣言したほうがよい。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    char ch;
 5
 6    ch = getchar();
 7    while (ch != '.') {
 8                  ┌─────┐
 9        putchar ( │     │ );          /* そのまま出力 */
10                  └─────┘
11        ch = getchar();
12    }
13  }
```

上のプログラムは次のように簡潔に書ける。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    char ch;
 5
 6    while ((ch=getchar()) != '.')
 7                  ┌─────┐
 8        putchar ( │     │ );
 9                  └─────┘
10  }
```

(注) これまでの授業で、

```
    if (n != 0)
         { 文1; 文2; ... }
    while (i >= 0)
         { 文1; 文2; ... }
    for (i=0; i<=10; i++)
            { 文1; 文2; .. }
```

のように文のかたまりを { } でくくってきたが、文が 1 つしかない場合は { } を省略してもよい。

[例題 3] キーボードからの入力を改行が押されるまで 1 文字ずつ読み込んで、それがアルファベットの小文字なら大文字に変換して、それ以外ならそのまま出力するプログラムをつくりなさい。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    char  ch;
 5
 6    while ((ch=getchar()) != '\n') {      /* 授業のコンピュータでは、\は円記号 */
 7                                  ┌──────┐
 8      if ('a' <= ch && ch <= │      │) /*入力文字がアルファベットの小文字かどうか*/
 9                                  └──────┘
10        putchar(ch-'a'+'A');               /* 小文字を大文字に変換して出力 */
11      else
12        putchar (ch);                      /* そのまま出力 */
13    }
14  }
```

[例題 4] キーボードから「f4」のような < 文字 >< 0 から 9 までの数字 > のコンビネーションを読み込み、最初の < 文字 > を < 数字 > 回繰り返して、この場合なら ffff と出力するプログラムを作りなさい。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    char ch1, ch2;
 5    int i;
 6
 7    ch1=getchar();                    /* 最初の文字の入力 */
 8    ch2=getchar();                    /* 次の文字の入力 */
 9                  ┌────────┐
10    for (i=1; i<=│        │ ; i++)    /* ch2 を整数に変換し、その回数だけ */
11                  └────────┘
12      putchar(ch1);                   /* ch1 を出力 */
13  }
```

[例題 5] キーボードからの数字入力を 1 文字ずつ読み込んで、変数 i にその値を代入するプログラムをつくりなさい。ただし、数字入力は . (ピリオド) で終わるものとする。(例えば、キーボードで「941.」と打つと整数型変数 i に 941 が代入されるということ)

```
 1  #include <stdio.h>
 2  main()
 3  {
 4    char ch;
 5    int  i=0;
 6
 7    while ((ch=getchar()) != '.') {
 8
 9      if ('0' <= ch && ch <= |       |)     /* 数字かどうか一応チェックする */
10
11                |       |                  /* 前に読み込んだ分を 1 桁くり上げ、*/
12        i = i * |       | + (ch - '0');     /* 今読んだ分を整数に直して加える */
13                |       |
14    }
15    printf("input integer = %d", i);
16  }
```

[例題 6] キーボードで「218+35」と打つと、計算して答を出力するようなプログラムをつくりなさい。