

第2章 データ型

■ 問題 2 - 1

データ型に関する次の記述について、正しいものには○を、誤っているものには×を解答欄に記入しなさい。ただし、下線以外の記述は正しいものとする。

- (1) unsigned char 型は、負の値を表現することができる。
- (2) unsigned short 型の変数は、0～+65,535の値を表現することができる。
- (3) char 型の変数は、-128～+127の値を表現することができる。
- (4) signed int 型は、0 を含む正又は負のいずれかの値を表現する。
- (5) char 型は、文字データを格納するものであるが、四則演算に使用できない。
- (6) char 型は1バイトの領域を割り当てるため、数の表現範囲は short 型と同じである。

〔解答欄〕

(1)		(2)		(3)		(4)		(5)		(6)	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--

■ 問題 2 - 2

定数データの記述方法に関する次の記述について、正しいものには○を、誤っているものには×を解答欄に記入しなさい。ただし、下線以外の記述は正しいものとする。

- (1) 文字配列 str の宣言時の初期化において、「str[] = "STRING"」と記述すれば、「str[] = {'S', 'T', 'R', 'I', 'N', 'G', '¥0'}」と記述したとき、str には同じ内容が格納される。
- (2) 整数定数表現では、「0120」、「80」、「0x50」は同じ値の整数定数を表している。
- (3) 整数定数表現では、「0102」、「72」、「0x42」は同じ値の整数定数を表している。
- (4) 文字配列では、文字列定数の終端を示す文字として、「¥0」が使用される。
- (5) char 型変数 moji に対する代入において、「moji = 'K'」としても、「moji = 71」としても、moji には同じ内容が格納される。

- (6) 文字配列 `str` の宣言時の初期化において、「`str[3] = {'T'}`」と記述すれば、「`str[3] = "T"`」と記述したとき、`str` には同じ内容が格納される。

【解答欄】

(1)		(2)		(3)		(4)		(5)		(6)	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--

■ 問題 2 - 3

変数名及び変数の宣言方法に関する次の記述について、正しいものには○を、誤っているものには×を解答欄に記入しなさい。ただし、下線以外の記述は正しいものとする。

- (1) “+” (プラス), “-” (マイナス), “*” (アスタリスク), “/” (スラッシュ) などの演算記号は、英字と組み合わせれば変数名として使用できる。
- (2) 変数名として “short” は使用できるが、“shortdata” は使用できない。
- (3) 変数名、定数名などの識別子は、31 文字までは識別されるがそれを超える文字列は認識されない。
- (4) 変数の宣言において、カンマで区切って 1 行にまとめ、一度に複数の変数の型を指定することはできない。
- (5) 変数は、宣言と同時に初期値を与えることはできない。
- (6) 数字リテラルの末尾に型接尾子 (“F” または “f”) を付加すると、実数になる。

【解答欄】

(1)		(2)		(3)		(4)		(5)		(6)	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--

■ 問題 2 - 4

データ型とその表現範囲に関する次の表の(1)~(4)にあてはまる適切な字句を解答群から選び、解答欄に記号で記入しなさい。

なお、char型は1バイト、int型は2バイトのメモリ領域が割り当てられるものとし、char型については、符号付きchar型、int型については、符号付きint型がサポートされている処理系であることを前提とする。

データ型	扱える数値の範囲
char	(1) ~ 127
unsigned char	0 ~ (2)
short(signed short)	-32768 ~ 32767
unsigned short	0 ~ 65535
int(signed int)	-32768 ~ (3)
unsigned int	(4) ~ 65535
long(signed long)	-2147483648 ~ 2147483647
unsigned long	0 ~ 4294967295
float	$\pm 3.4 \times 10^{-38}$ ~ 3.4×10^{38}
double	$\pm 1.7 \times 10^{-308}$ ~ 1.7×10^{308}

【解答群】

ア -128 イ -127 ウ 0 エ 255 オ 256
カ 32767 キ 65535

【解答欄】

(1)		(2)		(3)		(4)	
-----	--	-----	--	-----	--	-----	--

■ 問題 2 - 5

変数名及び処理内容に関する次の表を読んで、プログラムの(1)~(5)にあてはまる適切な字句を解答群から選び、解答欄に記号で記入しなさい。

変数名	処理内容
cntw	宣言時に初期値として-1を与える。
chrw	文字's'を代入する。
saidai	long型整数 100000 を代入する。
saisyo	long型整数-1 を代入する。
str_a	str_bの内容を代入する。
str_b	宣言時に初期値として文字列“wxyz”を与える。

<プログラム>

```
#include <string.h>    /* strcpy 関数を使用するための宣言 */

int main(void)
{
    int cntw = (1);
    long saidai, saisyo;
    char chrw, str_a[10], str_b[10] = (2);

    chrw = (3);
    saidai = (4);
    saisyo = (5);
    strcpy( (6) ); /* str_a に str_b の内容を複写 */

    return 0;
}
```

【解答群】

ア	'wxyz'	イ	'S'
ウ	"wxyz"	エ	"S"
オ	{wxyz}	カ	{S}
キ	-1	ク	-1F
ケ	-1L	コ	100000F
サ	100000L	シ	str_b, str_a
ス	str_a, str_b		

【解答欄】

(1)		(2)		(3)		(4)		(5)		(6)	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--

第2章 データ型 解答

■ 問題 2 - 1

(1)	×	(2)	○	(3)	○	(4)	○	(5)	×	(6)	×
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

- (1) unsigned char型は、正の値は表現することができるが、負の値を表現することはできない。
- (2) unsigned short型は符号ビットをもたない代わりに、16ビットをすべて数値で表すのに使用する。そのため、unsigned short型の変数が表現できる範囲は、0及び正の整数のみであるが、正の整数の範囲がsigned short型に比べ2倍表現できることから、範囲は0～+65,535である。
- (3) char型の変数は、-128から+127の値を表現する。
- (4) signed int型の変数は、0を含む正又は負の両方の値を表現する。なお、signedは一般的に省略し、「+」記号も省略することが多い。
- (5) 文字データはASCIIコードなどの文字コードで文字(char)型変数に格納される。文字コード自体は整数なので、四則演算を行うことができる。
- (6) char型は、1バイトの領域を割り当て、数の表現範囲は-128～+127である。一方、short(signed short)型は、2バイトの領域を割り当て、数の表現範囲は-32,768～+32,767である。

■ 問題 2 - 2

(1)	○	(2)	○	(3)	×	(4)	○	(5)	×	(6)	○
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

- (1) ダブルクォーテーション「"」で囲って文字列データの初期化を行うと、末尾に「\0」が自動で付加される(当然ながら、要素数が省略されている場合には、この「\0」が1文字と認識されて配列全体の長さが決まる)。中カッコ「{ }」で囲み、要素を「,」でそれぞれ区切って初期化を行う場合、末尾に「\0」を記述する必要がある。
- (2) 0120(8進数の120)を10進数に変換すると $1 \times 64 + 2 \times 8 = 80$ で、0x50(16進数の50)を10進数に変換すると、 $5 \times 16 + 0 = 80$ である。
- (3) 0102(8進数の102)を10進数に変換すると $1 \times 64 + 2 = 66$ で、0x42(16進数の42)を10進数に変換すると $4 \times 16 + 2 = 66$ である。
- (4) 文字列の終端を示す値として、「\0」が使用される。
- (5) 文字「K」はASCIIコードで「75」である。「moji = 'K」と「moji = 75」は同じ内容を表す。
- (6) 文字列データはダブルクォーテーション「"」または中カッコ「{ }」で囲って初期化することができる。ダブルクォーテーションで初期化する場合は、末尾に「\0」が自動的に付けられる。また、シングルクォーテーション「'」で囲って初期化する場合は、指定した要素数(ここでは3)に不足する分には0が格納される。(少なくともchar型において)0と'\0'は同一内容なので、「str[3] = {'T', '\0'}」と同じになる。

■ 問題 2 - 3

(1)	×	(2)	×	(3)	○	(4)	×	(5)	×	(6)	○
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

- (1) 演算記号は、同時に区切り子（デリミタ）とみなされる。したがって、英字と組み合わせても変数名としては使用できない。
- (2) 変数名の大文字と小文字は区別される。したがって、先頭文字が大文字の“Short”は、予約語“short”とは区別されるので、変数名“Shortdata”は使用できる。
- (3) 変数名として 31 文字を超える部分は認識されないが使用することはできる。
- (4) 例えば、“int var1, val2, val3;” のように記述することができる。
- (5) 例えば、“int var = 0;” のように記述することができる。
- (6) 変数名の末尾に“F”を付加しても、float（実数）型を指定したものとはならないことに注意する。

■ 問題 2 - 4

(1)	ア	(2)	エ	(3)	カ	(4)	ウ
-----	---	-----	---	-----	---	-----	---

- (1) char型は 1 バイト（8 ビット）のメモリ領域が割り当てられるので、-128～127 までの数値を扱うことができる。
- (2) unsigned char型は負の数値を扱うことができない代わりに、正の整数を 2 倍扱うことができ、0～255 の数値を表現できる。
- (3) int型は 2 バイト（16 ビット）のメモリ領域が割り当てられるので、-32,768～32,767 までの数値を表現することができる。
- (4) unsigned int型は符号部をもたないため、負の数値を表現することができない。

■ 問題 2 - 5

(1)	キ	(2)	ウ	(3)	イ	(4)	サ	(5)	ケ	(6)	ス
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

- (1) int型は、signed int型であるため、負の整数をそのまま設定する。
- (2) 文字配列に直接代入はできないが、宣言時には“{'a', 'b', ...}”のように文字をシングルクォーテーションで囲むか、“"ab..."”のように、文字列をダブルクォーテーションで囲むという形式で初期値を与えることができる。
- (3) 1 文字の場合はシングルクォーテーション“'’で囲む。
- (4), (5) 変数saidai, saisyoはlong型なので、代入文の右辺もlong型定数にする。
- (6) strcpy関数の第 1 引数には複写先の文字型配列を、第 2 引数には複写元の文字型配列を指定する。