

Training 5

変数とデータ型

株式会社イーシーエス 出版事業推進委員会

Lesson1 データ型とデータの大きさ



Point ◆◇データ型の範囲をしっかりと理解しよう!!

C言語ではさまざまなデータ型を使います。
データ型によって扱える値の範囲や使用するバイト数が決まっています。

【問題1】 □を埋めて表を完成させなさい。

データ型	使用バイト数	扱える値の範囲
① signed char	□バイト	□ ~ □
② unsigned char	□バイト	□ ~ □
③ signed short int	2バイト	□ ~ □
④ unsigned short int	□バイト	□ ~ □
⑤ signed int	4バイト	□ ~ □
⑥ unsigned int	4バイト	□ ~ □
⑦ float	□バイト	1.175E-□ ~ 3.402E+□
⑧ double	□バイト	2.225E-□ ~ 1.797E+□
⑨ long double	8バイト	2.225E-□ ~ 1.797E+□

【問題2】 下記代入プログラムを実行した場合の変数 x の値を 10 進数で答えなさい。

- | | |
|--------------------------------|---------------------------------|
| ① signed char x;
x = -30; | ② signed char x;
x = -200; |
| ③ signed char x;
x = 129; | ④ signed char x;
x = 258; |
| ⑤ unsigned char x;
x = 140; | ⑥ unsigned char x;
x = -30; |
| ⑦ unsigned char x;
x = 300; | ⑧ unsigned char x;
x = -270; |

【問題3】 下記代入プログラムを実行した場合の変数 x の値を10進数で答えなさい。

① signed char x;
x = 0x7f;

② signed char x;
x = 0x80;

③ signed char x;
x = 0xff;

④ unsigned char x;
x = 0x7f;

⑤ unsigned char x;
x = 0x80;

⑥ unsigned char x;
x = 0xff;

【問題4】 次のプログラムを実行した場合に表示される実行結果を答えなさい。

```
main.c
#include<stdio.h>

void main(void)
{
    int in_a;
    char ch_b;

    in_a = 0x80;
    ch_b = 0x80;

    if(in_a == 0x80)
    {
        printf("a");
    }

    if(ch_b == 0x80)
    {
        printf("b");
    }
}
```

Lesson2 キャスト演算



Poi nt ◆◇データ型変換を理解しよう!!

キャスト演算は演算で使われる変数のデータ型を一時的に変換することができます。
大きい型から小さい型への変換はデータが失われる可能性がありますので注意が必要です。

【問題1】 以下の空欄あてはまる言葉を解答群から選びなさい。

データ型には演算の際に型の優先順位が存在します。

優先順位は > > > > > です。

より小さいデータ型は一時的に に変換され、演算、処理されます。

解答群

ア. double	イ. int	ウ. unsigned int
エ. float	オ. Long	カ. unsigned long

【問題2】 変数 x、y、z を以下のように宣言したとき、次の式の演算結果を答えなさい。

【変数の宣言】

```
int x = 5;  
int y = 4;  
float z;
```

① $z = x / y;$

② $z = (\text{float})x / y;$

③ $z = (\text{float})(x / y);$

④ $x = x / y;$

⑤ $x = x / y * 100;$

⑥ $x = (\text{float})x / y * 100;$

解答

Training5 変数とデータ型

Lesson1 データ型とデータの大きさ

問題 1	以下の表
------	------

	使用バイト数	扱える値の範囲	
①	1 バイト	-128	~ 127
②	1 バイト	0	~ 255
③	2 バイト	-32768	~ 32767
④	2 バイト	0	~ 65535
⑤	4 バイト	-2, 147, 483, 648	~ 2, 147, 483, 647
⑥	4 バイト	0	~ 4, 294, 967, 295
⑦	4 バイト	1. 175E-38	~ 3. 402E+38
⑧	8 バイト	2. 225E-308	~ 1. 797E+308
⑨	8 バイト	2. 225E-308	~ 1. 797E+308

問題 2	①-30	②56	③-127	④2
	⑤140	⑥226	⑦44	⑧242

問題 3	①127	②-128	③-1	④127
	⑤128	⑥255		

問題 4	a
------	---

【解説】問題 2 ①

最初に変数には値が `const int` 型で格納されます。

-30 を 2 進数で表します(30 の 2 の補数をとる)

30 : 11110

-30: 100010(ビット反転させて、1 を足す)

-30 を `int` 型(4 バイト)の 2 進数で表します。

(符号ビットは 1 のため、前に 1 が入る)

111111111111111111111111111100010

最初に、この値が `x` に格納されます。

格納された値を変数の型に変換し、その値が変数の値となります。

`x` は `char` 型(1 バイト)のため上位 24 ビットが切り捨てられます。

11100010

この 11100010 が `x` の値となります。

では、この値を 10 進数にしましょう。

`x` は符号ありなので、先頭は符号ビットとなります。

11100010 は符号ビットが立っているため、負の数となります。そのため 2 の補数をとって絶対値を求めます。

11100010

00011101(ビット反転)

00011110(1 を足す)

00011110 は 10 進数に直すと 30 となります。

よって `x` の値は -30 となります。

-30 は `signed char` の範囲内のため、すぐに -30 という答えが出るかもしれませんが、変数に値を代入するときは上記の手順を踏んでいることを覚えていれば、範囲外であっても値を求めることができます。

②-200 を `int` 型の 2 進数で表すと

111111111111111111111111111100111000 となります。

この値を `char` 型に直すと

00111000 となります。

符号ビットは 0 なので、このまま 10 進数に直すと 56 となります。

③129 を `int` 型の 2 進数で表すと

00000000000000000000000010000001 となります。

この値を `char` 型に直すと

10000001 となります。

符号ビットは 1 なので、2 の補数をとって、絶対値を求めると

01111111 となり、この値を 10 進数に直すと 127 となるので、答えは -127 となります。

⑥-30 を `int` 型の 2 進数で表すと

111111111111111111111111111100010 となります。

この値を `unsigned char` 型に直すと

11100010 となります。

先頭ビットは 1 ですが、この型は符号なしのため、先頭は符号ビットではありませんので、このまま 10 進数に直すと 226 となります。

⑦300 を `int` 型の 2 進数で表すと

000000000000000000000000100101100 となります。

この値を `unsigned char` 型に直すと

00101100 となります。

このまま 10 進数に直すと 44 となります。

⑧-270 を `int` 型の 2 進数で表すと

111111111111111111111111101110010 となります。

この値を `unsigned char` 型に直すと

11110010 となります。

このまま 10 進数に直すと 242 となります。

【解説】問題 4

この問題で注意すべきことは、変数と `0x80` を比較しているとき、`int` 型で比較を行っているということです。

`int` 型 `in_a` と `0x80` の比較ではどちらも `0x80` なので真となり、`a` が出力されます。

`char` 型 `ch_b` の `0x80` を `int` 型に直すと符号ビットが 1 なので `0xfffff80` となり、`0x80` ではないので、偽となり、`b` は出力されません。

Lesson2 キャスト演算

問題 1	①ア	②エ	③カ	④オ
	⑤ウ	⑥イ		

問題 2	①1.0	②1.25	③1.0	④1
	⑤100	⑥125		

【解説】問題 2

<キャストの優先順位>

`double > float > unsigned long > long > unsigned int > int`

データ型の違うものどうしの計算はサイズの大きいデータ型に合わせられます。

そして、その計算結果は代入される先のデータ型に合わせられます。

※`int` 以下のデータ型はいったん `int` 型に変換され計算されますので注意しましょう。