

# Training 3

## C の基本

株式会社イーシーエス 出版事業推進委員会

## Lesson1 算術演算



### Point◆◇算術演算をマスターしよう!!

C言語のプログラムにおいて、加減乗除の数値計算を行うことを算術演算といいます。コンピュータに計算させるための演算子をここでしっかりと練習しておきましょう。

#### 【問題1】 次の演算結果を答えなさい。

①  $\square = 34 + 57$

②  $\square = 164 + 48$

③  $\square = -15 - 27$

④  $\square = 265 - 98$

⑤  $\square = -31 * 25$

⑥  $\square = 153 * 34$

⑦  $\square = 345 / 23$

⑧  $\square = 2272 / 284$

⑨  $\square = 100 \% 34$

⑩  $\square = 3150 \% 153$

#### 【問題2】 次の演算結果を答えなさい。

①  $\square = ( 63 + 18 ) \% 7 * 21$

②  $\square = 63 + 18 \% 7 * 21$

③  $\square = 316 - 174 \% 53 / 5 + 28 * 13$

④  $\square = ( 63 + 32 ) / 5 * ( ( 78 - 69 ) \% 6 )$

## Lesson2 代入演算



### Point◆◇代入演算を理解しよう!!

代入演算はプログラム内の演算を簡略に且つ見やすくする為のものです。  
算術演算の場合よりもシンプルなプログラムを書くことができます。

#### 【問題1】 次の x の演算結果を答えなさい。

① `int x = 87;`  
`x += 82;`

② `int x = 69;`  
`x += 157;`

③ `int x = -97;`  
`x -= 63;`

④ `int x = 54;`  
`x -= 18;`

⑤ `int x = 78;`  
`x *= 3;`

⑥ `int x = -26;`  
`x *= 10;`

⑦ `int x = -162;`  
`x /= 9;`

⑧ `int x = 176;`  
`x /= 11;`

⑨ `int x = 66;`  
`x %= 4;`

⑩ `int x = 495;`  
`x %= 54;`

## Lesson3 標準入出力「入力変換指定子」



### Poi nt◆◇変換形式に注意しよう!!

scanf 関数は、標準入力バッファから指定された変換形式にしたがって値を取得します。  
使用頻度が高いので、しっかりと覚えましょう。

※変換指定子については巻末資料参照

**【問題1】** 次の問題の□を埋めて scanf 関数を完成させなさい。

① int 型 10 進数表記で入力(符号あり)  
int x;  
scanf("□", &x);

② int 型 10 進数表記で入力(符号なし)  
unsigned int x;  
scanf("□", &x);

③ int 型 8 進数表記で入力  
int x;  
scanf("□", &x);

④ int 型 16 進数表記で入力  
int x;  
scanf("□", &x);

⑤ float 型で入力  
float x;  
scanf("□", &x);

⑥ double 型で入力  
double x;  
scanf("□", &x);

⑦ char 型で 1 文字を入力  
char x;  
scanf("□", &x);

⑧ char 型で文字列を入力  
char x[10];  
scanf("□", x);

**【問題2】** 次の問題の□を埋めて入力値に対応する scanf 関数を完成させなさい。

① 入力値 : 32, 15, 3c  
int dec, oct, hex;  
scanf("□", &dec, &oct, &hex);

② 入力値 : 2009/4/1  
int yy, mm, dd;  
scanf("□", &yy, &mm, &dd);

※ dec は 10 進数入力(符号あり)  
oct は 8 進数入力  
hex は 16 進数入力とする

※ yy mm dd は 10 進数入力(符号あり)とする

## Lesson4 標準入出力「出力変換指定子」



### Point ◆◇変換形式に注意しよう!!

printf 関数は、数値や文字列を指定された変換形式にしたがって変換し出力バッファに出力します。使用頻度が高いので、しっかりと覚えましょう。

### 【問題1】 次の出力結果になるように printf 関数を完成させなさい。

- ① 出力結果: 32  
`int x = 32;`  
`printf("□□", x);`
- ② 出力結果: A  
`char x = 'A';`  
`printf("□□", x);`
- ③ 出力結果: TEST  
`char x[ ] = "TEST";`  
`printf("□□", x);`
- ④ 出力結果: 1. 230000  
`float x = 1. 230000f;`  
`printf("□□", x);`
- ⑤ 出力結果: FF  
`int x = 255;`  
`printf("□□", x);`
- ⑥ 出力結果: 1. 953125e+004  
`float x = 19531. 25f;`  
`printf("□□", x);`
- ⑦ 出力結果: □□□443  
`int x = 443;`  
`printf("□□□", x);`
- ⑧ 出力結果: 0073  
`int x = 73;`  
`printf("□□□", x);`
- ⑨ 出力結果: 164□□□  
`int x = 164;`  
`printf("□□□□", x);`
- ⑩ 出力結果: 24. 5000  
`float x = 24. 5f;`  
`printf("□□□□", x);`

**【問題2】** 次の□に printf 関数の出力結果を書きなさい。

① int x = 34;  
printf("%d", x);

出力結果 :

② char x = 84;  
printf("%c", x);

出力結果 :

③ char x[ ] = "abc";  
printf("%5s", x);

出力結果 :

④ int x = 108;  
printf("%x", x);

出力結果 :

⑤ int x = 47;  
printf("%-5o", x);

出力結果 :

⑥ double x = 52.104;  
printf("%0.4lf", x);

出力結果 :

⑦ int x = 67;  
printf("%04d", x);

出力結果 :

⑧ double x = 142.65;  
printf("%e", x);

出力結果 :

⑨ double x = 0.00145;  
printf("%E", x);

出力結果 :

⑩ int x = 2009;  
int y = 4;  
int z = 1;  
printf("%d/%02d/%02d", x, y, z);

出力結果 :

# 解答

## Training3 C の基本

### Lesson1 算術演算

問題1	①91	②212	③-42	④167
	⑤-775	⑥5202	⑦15	⑧8
	⑨32	⑩90		

問題2	①84	②147	③677	④57
-----	-----	------	------	-----

### Lesson2 代入演算

問題1	①169	②226	③-160	④36
	⑤234	⑥-260	⑦-18	⑧16
	⑨2	⑩9		

### Lesson3 標準入出力「入力変換指定子」

問題1	①%d	②%u	③%o	④%x
	⑤%f	⑥%lf	⑦%c	⑧%s

問題2	② %d, %o, %x	②%d/%d/%d
-----	--------------	-----------

### Lesson4 標準入出力「出力変換指定子」

問題1	①%d	②%c	③%s	④%f
	⑤%X	⑥%e	⑦%6d	⑧%04d
	⑨%-6d	⑩%.4f または%.4f		

問題2	①34	②T
	③□□abc	④6c
	⑤57□□	⑥52.1040
	⑦0067	
	⑧1.426500e+002	
	⑨1.450000E-003	
	⑩2009/04/01	

#### 【解説】問題1

⑤255を16進数にすると、FFとなります。そのためxを16進数で出力させることが求められています。さらにFFは大文字なので、出力変換指定子は%Xとなります。

⑦出力結果をみると6個分のスペースに右詰めで出力されています。printf関数では%の直後に正の整数を入れることで、その分だけスペースを使い、右詰めで出力させることができます。そのため%6dとすることで、6個分のスペースで、右詰めで10進数出力することができます。

⑨出力結果をみると、6個分のスペースに左詰めで出力されています。printf関数では%の直後に負の整数を入れることで、その絶対値の分だけスペースを使い、左詰めで出力させることができます。そのため%-6dとすることで、6個分のスペースで、左詰めで10進数出力することができます。

⑩出力結果をみると7個分のスペースに小数点以下4桁で出力されています。printf関数では最小フィールド幅と精度を設定します。整数の絶

対値の分だけスペースを使い、小数の分だけ小数点以下の桁を出力させることができます。そのため%.4fとすることで、7個分のスペースで、小数点以下を4桁にして、float型で出力することができます。

※ただし、スペースを確保しても、出力する数値の桁数が、確保したスペースをオーバーしている場合、数値の桁数分だけ出力されます。

## 巻末資料

### 変換指定子

変換指定子とは書式指定の最後に書くアルファベット記号です。  
データの表示形式を指定します。

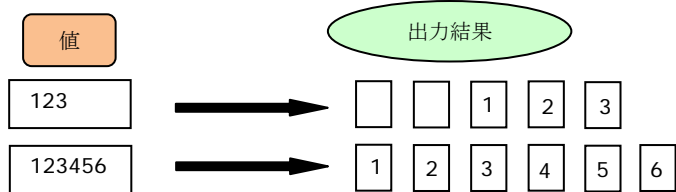
<p><b>整数の 10 進数表示(%d)</b> 整数値を 10 進数で表示できる。</p> <p>値</p> <p>出力結果</p> <p>int_val = 123 → 123 int_val = -123 → -123</p>	<p><b>整数の 16 進数表示(%x,%X)</b> 整数値を 16 進数で表示できる。 (16 進数を大文字で表示するか、小文字で表示するか選択可能)</p> <p>値</p> <p>出力結果</p> <p>int_val = 123 → 7b int_val = -123 → FFFFFFF85</p>
<p><b>整数を符号なし整数として 10 進数表示(%u)</b> 整数値を符号なし整数として、10 進数で表示できる。 (int 型(符号あり整数型)に負数を代入して表示すると、符号ビットを無視して符号なし整数として表示してしまう。)</p> <p>値</p> <p>出力結果</p> <p>int_val = 123 → 123 int_val = -123 → 4294967173</p>	<p><b>実数を 10 進小数形式で表示(%f)</b> 実数を 10 進小数形式で表示できる。</p> <p>値</p> <p>出力結果</p> <p>int_val = 123.123 → 123.123000 int_val = -123.123 → -123.123000</p>
<p><b>実数を指数表記で表示(%e,%E)</b> 実数を 10 を基数とした指数表記で表示できる。(大文字表示か小文字表示か選択可能)</p> <p>値</p> <p>出力結果</p> <p>int_val = 123.123 → 1.231230e+002 int_val = -123.123 → -1.231230e+002</p>	
<p><b>文字表示(%c)</b> 整数値を文字コードとして、文字に変換して表示する。</p> <p>値</p> <p>出力結果</p> <p>char_val = 0x61 または 97 → a int_val = 0x61 または 97 → a</p>	<p><b>文字列表示(%s)</b> 配列に格納された整数値を文字コードとして、NULL 文字までを文字に変換して表示する。 引数は、文字列の先頭のポインタとする。</p>



### 表示する最小のフィールド幅

表示する最小のフィールド幅を設定できます。最小フィールド幅に満たないとき空白が入ります。また最小フィールド幅より長いときは表示する長さを優先します。

「%5d」とした場合、半角で5文字分の領域を表示するときに確保します。



### 精度(小数点以下の桁数)

小数点以下の桁数を指定できます。精度を指定するには「. (ピリオド)」の後に桁数を書きます。

