

# プログラミング基礎演習I (第3・4回)

伊原彰紀

Akinori Ihara

akinori-i@is.naist.jp

(伊原: B307室)

# 前回の復習

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

```
main()
```

```
{
```

```
    /* Display a greeting message */
```

```
    printf("Hello NAIST!!\n");
```

```
}
```

**/\* \*/ で囲まれた部分がコメント。  
実行には影響しない。**

# 前回の復習

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

```
main()
```

```
{
```

```
    int year;
```

```
    year = 2014;
```

```
    printf("This year is %d\n", year);
```

```
}
```

整数を扱う型宣言

代入演算

整数を表示する.

%dで表示する  
変数を指定

# 前回の復習

```
#include <stdio.h>
main()
{
    float height;
    height = 170.5;
    printf("I am %f cm tall\n", height);
}
```

小数を扱う型宣言

代入演算

小数（実数）を表示する.

# データ型と宣言

	型名	バイト長	とりうる値の範囲
文字型	char	1	-128～127
	unsigned char	1	0～255
整数型	int	4	-2147483648～2147483647
	long	4	-2147483648～2147483647
	short	2	-32768～32767
実数型	float	4	$1.18 \times 10^{-38} \sim 3.40 \times 10^{38}$
	double	8	$2.23 \times 10^{-308} \sim 1.80 \times 10^{308}$

- 各整数型 (int/long/short) には unsigned 版 もある.
- 実数型には unsigned はない.
- 実数型は浮動小数点型とも呼ばれる.

# データ型と宣言

	型名	バイト長	とりうる値の範囲
文字型	char	1	-128~127
	unsigned char	1	0~255

- 文字型のデータはプログラム内では '(1重引用符)でくる。  
例: 'A'        'z'        '+'
- 文字はアスキーコードで表される。  
例: 'A'は65    'B'は66        'a'は97

# 変数の初期化

- 変数の宣言時に初期化することができる。

```
int i, j;  
i = 1;  
j = 2;
```



```
int i=1, j=2;
```

```
float i=1.0, j=2.2;
```

```
char i='a', j='+';
```

# 例題2-1: さまざまな型の変数を使ってみる.

```
#include <stdio.h>

main()
{
    int i = 10;
    double x = 123.45
    char z = 'A';

    printf("i=%d, x=%lf, z=%c\n", i, x, z);
}
```

整数を表示

倍精度実数  
(double)を表示

文字を表示



## 例題2-2: char型の変数の値を試みる.

```
#include <stdio.h>

main()
{
    char z = 'A';

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

文字を表示

整数を表示

# 定数

整数定数		例
	10進数.	2004, -100
	8進数: 0が頭につく文字列.	010 = 8
	16進数: 0xが先頭につく文字列. 10から15までの数字は英字のa(A)からf(F)に対応させる.	0x10 = 16 0xff = 255
浮動小数点定数		例
	整数部、小数点、小数部、指数部からなる.	1.0, -12.34 1.5e-3 (= $1.5 \times 10^{-3}$ )
文字列定数		例
	二重引用符で囲まれた文字の列.	"Nara"

## 例題2-3: 10進数と16進数

```
#include <stdio.h>

main()
{
    int i, j;
    i = 10;
    j = 0x10;

    printf("i=%d, j=%d\n", i, j);
}
```

## 例題2-4: 16進数で表示してみると

```
#include <stdio.h>

main()
{
    int i, j;
    i = 10;
    j = 0x10;

    printf("i=%x, j=%x\n", i, j);
}
```

整数を16進  
数で表示

## 例題2-5：整数型に実数を代入すると

```
#include <stdio.h>

main()
{
    int i;
    i = 12.345;

    printf("i=%d\n", i);
}
```

## 例題2-6: 型変換と代入演算子の順序

```
#include <stdio.h>

main()
{
    int    i ;
    float  x, y;

    x = i = y = 3.65 ;
    printf ( " x=%f    i=%d    y=%f  \n", x, i, y);
}
```

### 実行結果

```
x=3.000000    i=3    y=3.65000
```

# 例題2-6: 型変換と代入演算子の順序

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

```
main()
```

```
{
```

```
    int    i ;
```

```
    float  x, y;
```

```
    x = i = y = 3.65 ;
```

```
    printf ( " x=%f    i=%d    y=%f  \n", x, i, y);
```

```
}
```

後ろから順に代入される。

実行結果

x=3.000000 i=3 y=3.65000

# 式と算術演算子

- 算術演算子

- 和差積:  $a + b$   $a - b$   $a * b$

- 商:  $a / b$

- 整数演算の商は小数点以下切り捨て

- 剰余:  $a \% b$

- 単項演算子

- インクリメント(1増加):  $a++$   $++a$

- デクリメント(1減少):  $a--$   $--a$

- 括弧:  $()$



# 例題2-7: ++演算子を使ってみよう

```
#include <stdio.h>

main()
{
    int    i = 5;
    i=i+1;
    printf ("i=%d \n", i);

    i++;
    printf ("i=%d \n", i);
}
```

# a++と++aの違い

- `a = b++;` `b`の値が`a`に代入され, その後`b`に1を加える
- `a = ++b;` `b`に1を加え, その結果が`a`に代入される

## 例題8

```
#include <stdio.h>

main()
{
    int    a, b = 5;
    a=b++;
    printf ("a=%d \n", a);
    a=++b;
    printf ("a=%d \n", a);
}
```

# 演算子の優先順位

- 例えば、「\*」、「/」、「%」は「+」、「-」より優先される

`int i = 10+3*5;` の時、`i`の値は25

- C言語には非常に多くの演算子があるため、すべてを覚えるのは大変・・・
- 自信がない時は、必ず括弧「()」を使う  
()の中が必ず優先順位が上

`int i = 10+(3*5);` の時も、`i`の値は25

# 算術演算子(代入演算子)

演算子	意味	使用法	他の表記
<b>=</b>	右辺から左辺への代入	<code>a=b;</code>	
<b>+=</b>	左辺の変数と右辺の変数を加算し、左辺に代入	<code>a+=b;</code>	<code>a=a+b;</code>
<b>-=</b>	左辺の変数から右辺の変数を減算し、左辺に代入	<code>a-=b;</code>	<code>a=a-b;</code>
<b>*=</b>	左辺の変数と右辺の変数を乗算し、左辺に代入	<code>a*=b;</code>	<code>a=a*b;</code>
<b>/=</b>	左辺の変数を右辺の変数で除算し、左辺に代入	<code>a/=b;</code>	<code>a=a/b;</code>
<b>%=</b>	左辺の変数を右辺の変数で割った剰余を、左辺に代入	<code>a%=b;</code>	<code>a=a%b;</code>

# キーボードからの入力方法

int 型の変数 kにデータを入力するには

```
int k;  
scanf("%d", &k);
```

%? は、printf(); と同様

# キーボードからの入力方法

char 型の変数 kにデータを入力するには

```
char k;  
scanf("%c", &k);
```

## 例題2-8: データを読み込む

```
#include <stdio.h>

main()
{
    int apple1, apple2;
    apple1 = 15;
    printf("りんごが%d個あります. \n", apple1);
    printf("さらに何個加えますか?");
    scanf("%d", &apple2);
    printf("りんごを%d個加えます. \n", apple2);
    apple1 += apple2;
    printf("合計で%d個です. \n", apple1);
}
```

変数 apple2の値を  
キーボードから得る

# 演習問題

演習2-1 キーボードから整数を2個, 順に読み込んで, それらの四則演算と剰余を出力するプログラムを作成せよ.

表示例

Enter two numbers

5

3

キーボードから入力

add 8

sub 2

mul 15

div 1

mod 2



# 演習問題

演習2-2 センチメートルとインチの相互変換をするプログラムを作成せよ.

なお, 1インチは2.54cmである.

表示例

Input number

38.1

キーボードから入力

38.1 cm = 15 inch

38.1 inch = 96.774 cm

優先順位	演算子の種類		演算子
<div>高</div> <div>↑</div> <div>↓</div> <div>低</div>	式		( ) [ ] -> .
	単項演算子		! ~ ++ -- - (型) * & sizeof
	二項演算子	乗除	* / %
		加減	+ -
		シフト	<< >>
		比較	< <= > >=
		等価	== !=
		ビットAND	&
		ビットXOR	^
		ビットOR	
		論理AND	&&
		論理OR	
	条件演算子		? :
	代入演算子		= += -= *= /= %= >>= <<= &= ^=  =
	カンマ演算子		,

### 演習2－3:

int型の変数xとyに, それぞれ5と7を代入し, その和と積を表示せよ.

### 演習2－4:

キーボードから入力された1個の整数値の2乗, 3乗を求めて表示せよ.

### 演習2－5:

キーボードから正の整数値を読みこみ, その数の下一桁の数字を表示せよ. たとえば, 2016が入力されたら6を出力すること.

(ヒント: 剰余「%」を使う)

### 演習2－6:

円の半径を入力とし, 円の面積を計算せよ. (円周率=3.14とする)

### 演習2－7:

入力された実数値の小数点以下の値を表示せよ. (例: 12.34 → .34)

(ヒント: 実数から整数への型変換を利用する)

### 演習2－8:

入力された実数値の小数点以下を四捨五入したものを表示せよ.

### 演習2－9:

キーボードから入力された大文字のアルファベット(A～Zのどれか1文字)を小文字に変換して表示せよ。(例:Aが入力されたらaと出力する)

(ヒント:A, B, C, ...のアスキーコードは65,66,67... 一方, a,b,c,...のアスキーコードは97,98,99...

### 演習2－10:

1本108円の缶コーヒー1本と, 1本128円の牛乳パック2本を購入し, 千円札で払った場合のお釣りを計算して表示せよ.

ただし, 8%の消費税を追加し, お釣りの額は整数とする.

(消費税を四捨五入するかどうかは自由とする.)