

プログラミング入門

第3回講義

Cプログラムの初歩 — はじめてのCプログラミング — (その2)

◆ マークのあるサンプルプログラムは
`/home/course/prog0/public_html/2006/lec/source/`
下に置いてありますから、各自自分のディレクトリに
コピーして、コンパイル・実行してみてください

日本語のテキストファイルを作成する際に気をつけたいといけない文字コードの話が
Lec11-23,24にあります
ので、読んでください

今日の主な項目

- $a = a+1$ とはということ？
- `int`型の派生と`int`型以外の型(教科書3章)
- 型変換(同上)
- 入出力(教科書2.3節)



$a = a + 1$ とはどういうこと？

- 数学の世界では $a = a + 1$ はおかしい
→ 移項すると0と1が等しくなる??

- Cでは「 $=$ 」の記号は「代入」を表す

- つまり、 $a = a + 1$ は、

「変数 a の値に1を加えてそれを新しい a の値にする」ことを意味している。



$a = a + 1$ の使用例

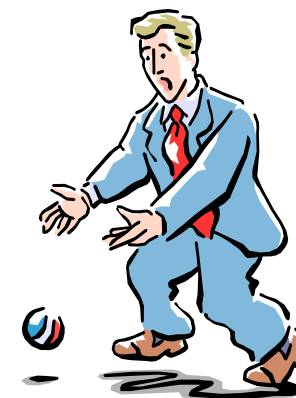
- どんな時に $a = a + 1$ が役に立つだろう？
- 回数を数える場合などに有効
- 以下の例は、数を入力し0が入力されるまでに
入力された回数を調べるもの

```
int a = 0, b;  
while (1) {  
    scanf("%d", &b);  
    if (b == 0) break;  
    a = a + 1;  
}
```



代入の時に注意する点

- 代入とは左辺の値を右辺の値と同じにする(書き換える)こと
- つまり以下の二つの式は意味が異なる
 - $a = 10$ (aに10を代入する)
 - $10 = a$ (10にaの値を代入する: **エラー**)
- $a + b = 20$ も **エラー** になる
 - 代入が出来ない!



int型ではない整数型(p.64,65)

- int型以外に以下のような整数型がある
 - short int
(又は省略して**short**)型
 - long int
(又は省略して**long**)型
- 会津大環境では右の範囲を表現出来る
- printfではどの整数型も「**%d**」を使用して表示可

| | |
|--------|--------------------------------|
| short型 | -32768 ~ 32767 |
| long型 | -2147483648 ~ 2147483647 |
| int型 | -2147483648 ~ 2147483647 |

整数型以外の数の型： 浮動小数点型(p.65)

- 身の回りにある数でint型でうまく表現出来ないものもある

- 例えば

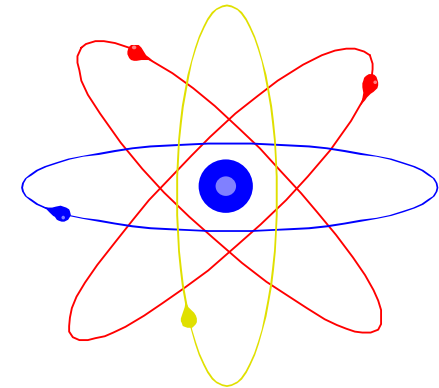
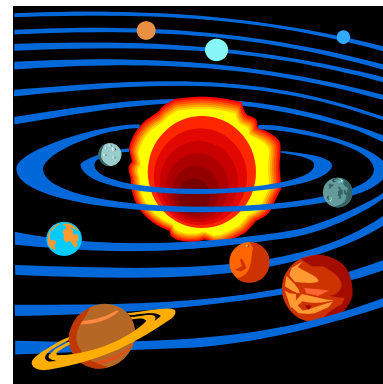
- 身長 (173.5 cm)

- 円周率 (3.141.....)

- 光年 (9兆4600億キロメートル) : 大きなもの

- 電子の重さ (9.1×10^{-31} キログラム) : 小さなもの

- 単精度・倍精度2種類の型



浮動小数点変数の宣言と形式

■ 2種類の浮動小数点型

■ 単精度浮動小数点型: **float**

```
float a;
```

■ 倍精度浮動小数点型: **double**

```
double b;
```

• 2種類の表現形式

– 小数点形式: 3.14

```
a = 3.14;
```

– 指数形式: 4.3e-5 (4.3x10⁻⁵のこと)

```
b = 4.3e-5;
```

(eは大文字でも小文字でも良い)

– (問題) それでは3.6x10²³はどうなりますか？



printfでの浮動小数点数の表示方法(p.75)

- 浮動小数点数を表示するには
%f、%e、%E、%g、%Gを使用する(**%d**は使用出来ない！)

float a = 0.00125 の時

| 書式 | 説明 | 表示 |
|-------------------------------|-----------------|--------------|
| <code>printf("%f", a);</code> | 小数点で表示 | 0.001250 |
| <code>printf("%e", a);</code> | 指数で表示 (e小文字) | 1.250000e-03 |
| <code>printf("%E", a);</code> | 指数で表示 (E大文字) | 1.250000E-03 |
| <code>printf("%g", a);</code> | 適切な形式で表示 (e小文字) | 0.00125 |
| <code>printf("%G", a);</code> | 適切な形式で表示 (E大文字) | 0.00125 |

| | | |
|-------------------------------|---|------------|
| <code>printf("%d", a);</code> | 間違って%dを使用すると、誤った数字が表示されるので注意！ (コンピュータによって値が違う) | 1073741824 |
|-------------------------------|---|------------|

各型の表現出来る範囲(p.65,66)

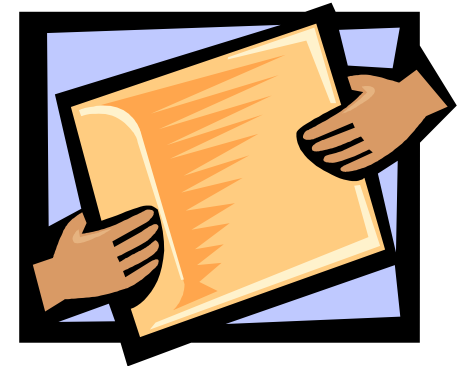
- 範囲: **short < long < float < double**
(会津大WSではint は longと同じ)

| | |
|---------------|---|
| short | -32768~32767 |
| long | -2147483648~2147483647 |
| float | $\pm 3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$ |
| double | $\pm 1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$ |

lec03-8の答え 3.6e23

型変換(p.95)

- 異なる型間でデータの変換を行うこと。
- 整数型 → 浮動小数点型: **そのまま**
3 → 3.0
- 浮動小数点型 → 整数型: **切り捨て**
3.15 → 3
- 以下の場合に**型変換**が行われる
 - 混合演算
 - 代入
 - 明示的な型変換



混合演算(p.95)

- 型が混ざった演算のことを「**混合演算**」と言う
(例えば `int a; float b;` の時の `a + b` の演算)
- 混合演算の結果は、**より大きな範囲を持つ型に自動的に変換**される

| a \ b | short | long | float | double |
|--------|--------|--------|--------|--------|
| short | short | long | float | double |
| long | long | long | float | double |
| float | float | float | float | double |
| double | double | double | double | double |

代入(p.97)

- 型が違うものを代入すると、右辺の値が代入される側(左辺)の型に変換される。

```
float a;  
int i = 3;  
a = i;
```

→aの値は3.0になる

```
float a = 3.14;  
int i;  
i = a;
```

→iの値は3になる

(整数型への型変換は切り捨てになる)



演算による型変換時の注意：割り算

- 代入すると左辺の型になるというので、以下のような計算をした。でも結果は2.5とならず、2.0になる。何故だろう？

```
int i = 10;  
int j = 4;  
double a;  
a = i / j;
```

- これはi、jが整数型なので、**i/jの計算が整数型で行われてしまう**からである。(つまりaへ代入する時にはすでに結果が2になっている)

/home/course/prog0/public_html/2006/lec/source/lec03-1a.c

◆ キャスト(p.98)

- 正しい割り算結果を得るために、**割り算の前に**整数*i*、*j*を浮動小数点型(double又はfloat)に変換する必要がある。
- このように**明示的に型を変換**することを「**キャスト**」と呼ぶ。
- キャストは変数や定数の前に変換後の型を**カッコ()**でくくってつけることで示す。つまり「**(変換したい型)変数**」となる

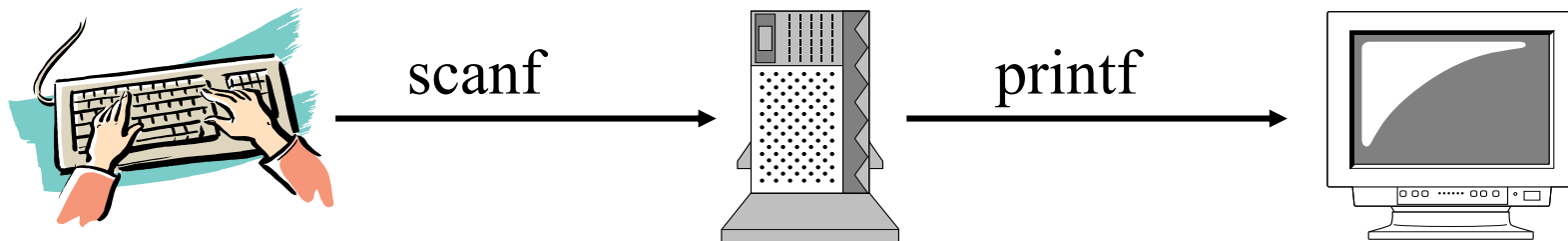
```
int i = 10;  
int j = 4;  
double a;  
a = (double) i / (double) j;
```

これで正しい計算が出来る

/home/course/prog0/public_html/2006/lec/source/lec03-1b.c

scanf(p.48)

- scanfを使うとキーボードから入力することができる
- 大雑把に言うと、scanfは丁度printfの反対のような動作をする。



scanfの使用方法

- scanfの基本形は以下の通りである。

```
scanf ("%d", &i);
```

10進数の入力

変数iにデータが入力される
「&」は今はおまじない

- このscanf文は「10進整数を一つ読んで、整数型変数iに代入する」ことを示している。

◆ scanfの使用例

- 例えば、年齢を入力するプログラム(の一部)は以下のようなになる

```
int age;  
printf("How old are you ?\n");  
scanf("%d", &age);
```

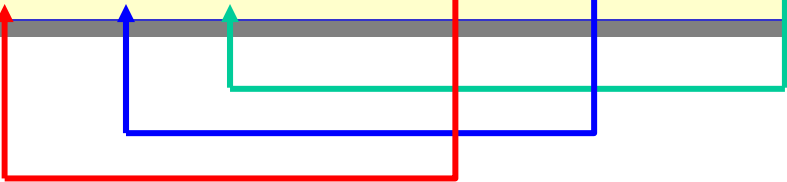
- これによって変数ageにはキーボードから入力された数値(この場合年齢)が代入される。

`/home/course/prog0/public_html/2006/lec/source/lec03-2.c`

複数データの入力(p.105)

- 一度に複数データを入力する場合は以下のようにする。

```
scanf ("%d%d%d", &i, &j, &k);
```



- printfと同様に、3つ並んだ「%d」は順に変数i,j,kに対応
- 「%d %d %d」のように空白をあけても構わない。
- データは**10 12 32**のように1つ以上の空白をあけて入力する。あるいは以下のように改行で区切っても良い。

10

12

32

int型以外の入力

- short、long、float、doubleなど、int型以外のデータの入力は以下のものを使用する。
(%e, %g等はLec03-9参照のこと)

short(10進): %hd

long(10進): %ld

float : %f, %e, %E, %g, %G

double : %lf, %le, %lE, %lg, %lG

例: `scanf("%lf", &var_dbl);`

scanfの注意(1)

- 「“ ”」内には%d等の書式指定および空白以外は書かない。特に改行(**\n**)を書くと、トラブルの元になる。



```
scanf ("%d\n", &i);
```

- 入力する変数には必ず「&」をつける。この意味は後期 Programming-1 で習う。それまでは「おまじない」だと思っていて差し支えない。(興味のある人は自習してみると良い)



```
scanf ("%d", i);
```

scanfの注意(2)

- printfはfloat/doubleどちらも「%f」で表示出来る。



```
double dd = 3.14;  
printf ("%f\n", dd);
```

- ところが、scanfは型に合った形式の指示をしないとうまく読み込めない！（この場合は%lfにすべき）



```
double dd = 3.14;  
scanf ("%f", &dd);
```

scanf/printfの書式のまとめ

| 変数の型 | printf | scanf |
|--------|----------------|---------------------|
| short | %d | %hd |
| int | %d | %d |
| long | %d | %ld |
| float | %f,%e,%E,%g,%G | %f,%e,%E,%g,%G |
| double | %f,%e,%E,%g,%G | %lf,%le,%lE,%lg,%lG |

(注: 整数の16進数表示を指定する「%x(入力では%x、%hx、%lx)」という書式もある)

◇ プログラム例 (平均を求める)

```
#include <stdio.h>
main()
{
    int i,j,k;
    float average;

    printf("3人分の体重(整数kg)を空白で分けて入力してください : ");
    scanf("%d%d%d",&i,&j,&k);

    average = ((float)(i + j + k))/3;

    printf("3人の体重はそれぞれ %dkg %dkg %dkgで、\n",i,j,k);
    printf("その平均は %fkg となりました\n",average);
}
```

キャスト!

```
std1dc1{s1000000}1: ./a.out
3人分の体重(整数kg)を空白で分けて入力してください : 57 72 49
3人の体重はそれぞれ 57kg 72kg 49kgで、
その平均は 59.333332kg となりました
std1dc1{s1000000}2:
```

/home/course/prog0/public_html/2006/lec/source/lec03-3.c