

# cp-3. 計算

(C プログラミング入門)

URL: <https://www.kkaneko.jp/cc/adp/index.html>

金子邦彦



# 内容

例題 1. 自由落下距離

四則演算

例題 2. 三角形の面積

浮動小数の変数, 入力文, 出力文, 代入文

例題 3.  $\sin$  関数による三角形の面積

ライブラリ関数

- プログラムを使って、自分の思い通りの計算ができるようになる
  - 四則演算
  - ライブラリ関数（三角関数、対数・指数関数など）
- 見やすいプログラムを書くために、ブロック単位での字下げを行う

# 例題 1 . 自由落下距離



- 自由落下距離を求めるプログラムを作る
  - 地上で物を落とし始めた後の自由落下距離を求める
  - 重力加速度  $g$  は  $9.8$  とする
  - 自由落下距離を求めるために, プログラム中に, 計算式  $y = (9.8 / 2.0) * x * x$  を書く

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf\n", &start_x );
    printf( "step_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf\n", &step_x );
    fp = fopen( "z:\data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = ( 9.8 / 2.0 ) * x * x;
        printf( "x= %f, y= %f\n", x, y );
        fprintf( fp, "x=, %f, y=, %f\n", x, y );
    }
    fprintf( stderr, "file created\n" );
    fclose( fp );
    return 0;
}
```

自由落下距離の  
計算を行っている部分

# 実行結果例



```
start_x = 0
step_x = 0.1
x= 0.000000, y= 0.000000
x= 0.100000, y= 0.049000
x= 0.200000, y= 0.196000
x= 0.300000, y= 0.441000
x= 0.400000, y= 0.784000
x= 0.500000, y= 1.225000
x= 0.600000, y= 1.764000
x= 0.700000, y= 2.401000
x= 0.800000, y= 3.136000
x= 0.900000, y= 3.969000
x= 1.000000, y= 4.900000
x= 1.100000, y= 5.929000
x= 1.200000, y= 7.056000
x= 1.300000, y= 8.281000
x= 1.400000, y= 9.604000
(以下続<)
```

# 四則演算のための演算子



- + 和
- - 差
- \* 積
- / 商

## 例題 2 . 三角形の面積

- 底辺と高さを読み込んで、面積を計算するプログラムを作る

例) 底辺が 2 . 5 , 高さが 5 のとき,

面積 : 6 . 2 5

- 底辺, 高さ, 面積を扱うために, 浮動小数の変数を 3 つ使う



```
#include <stdio.h>
#pragma warning(disable:4996)
int main()
{
    double teihen;
    double takasa;
    double menseki;
    printf("teihen=");
    scanf("%lf", &teihen);
    printf("takasa=");
    scanf("%lf", &takasa);
    menseki = teihen*takasa*0.5;
    printf("menseki=%f¥n", menseki);
    return 0;
}
```

入力部分

計算部分

出力部分

# 実行結果例

x=2.5

y=5

z = 6.250000

# プログラム実行順



`printf("teihen=");` メッセージ「teihen=」を表示

`scanf("%lf", &teihen);` 浮動小数データを読み込み

`printf("takasa=");` メッセージ「takasa=」を表示

`scanf("%lf", &takasa);` 浮動小数データを読み込み

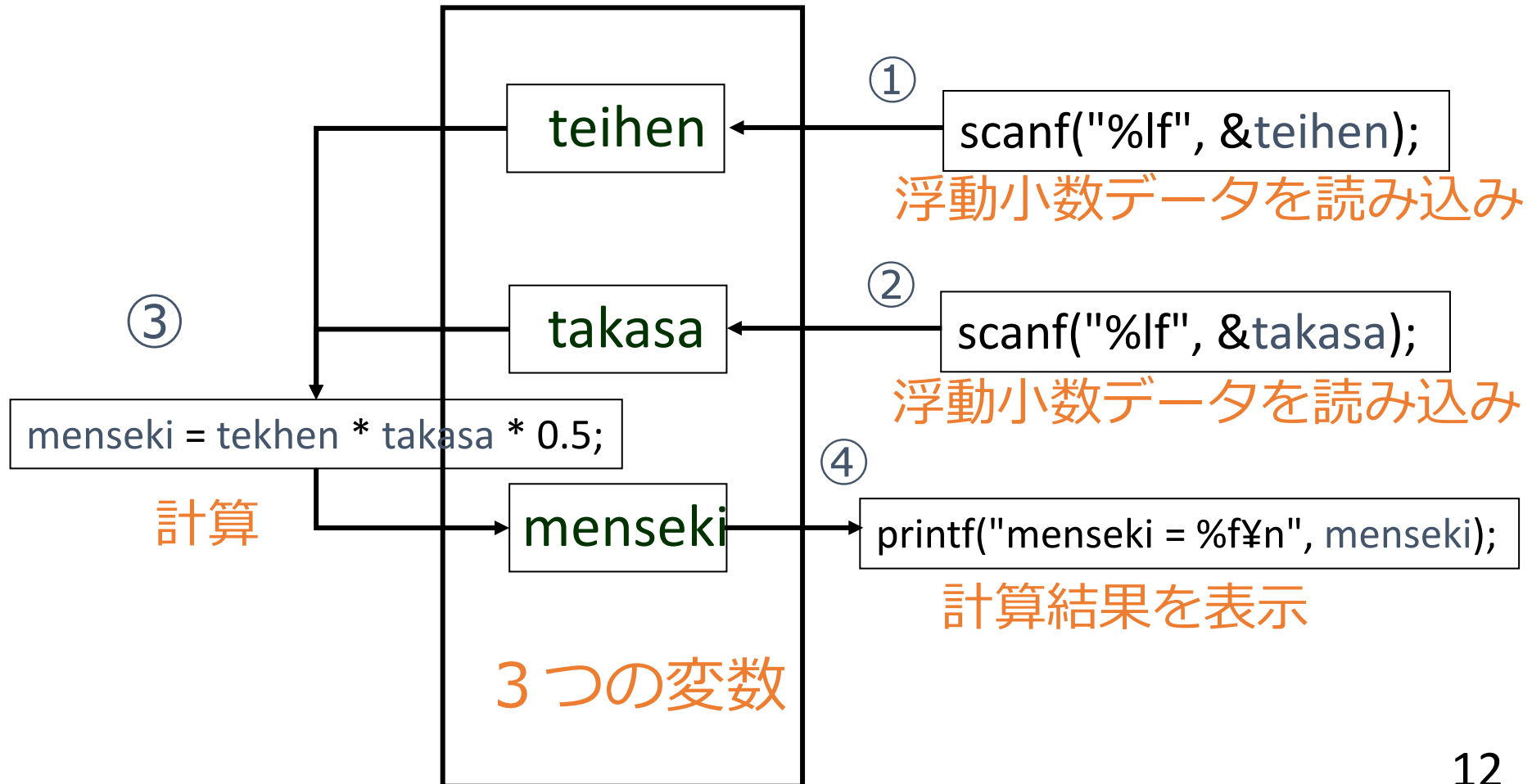
`menseki = teihen * takasa * 0.5;` 計算

`printf("menseki = %f¥n", menseki);` 計算結果を表示

`return 0;` 終わり

# プログラムとデータ

## メモリ



# 変数宣言

- 変数は、データを入れるための容器
- 変数宣言とは、変数を使うために、名前と型を書いて、変数の使用をコンピュータに伝えること

```
double teihen;
```

} 浮動小数データで、変数名は「teihen」

```
double takasa;
```

} 浮動小数データで、変数名は「takasa」

```
double menseki;
```

} 浮動小数データで、変数名は「menseki」

「double」とは、浮動小数データという意味。

```
menseki = teihen*takasa*0.5;
```

- 計算結果 (teihen\*takasa\*0.5) を, 変数 menseki に格納する (このことを, 代入という)
- 「=」は, 変数に計算結果等を格納するという意味. 「両辺が等しい」という意味ではない

# 入力, 出力とは

- 入力
  - データの読み込み  
(読み込まれたデータは変数に格納される)
- 出力
  - メッセージの表示
  - データの表示  
(変数に格納されたデータが表示される)

# 入力文



```
scanf("%lf", &teihen);
```

書式

&

読み込むべき変数名

- 入力文とは、データを読み込むための文
- 書式と読み込むべき変数名を書く
  - 書式： 浮動小数データを読み込む場合、書式は「%lf」と書くことになっている
  - 変数名： 変数名の前には「&」を付けること



# いろいろな入力

```
double x;  
scanf( "%lf¥n", &x );
```

浮動小数の変数  
x への入力

```
double a;  
double b;  
scanf( "%lf¥n", &a );  
scanf( "%lf¥n", &b );
```

浮動小数の変数  
a と b への入力

# 出力文



```
printf("menseki=%f¥n", menseki);
```

書式

表示すべき変数名

- 出力文とは、データとメッセージを表示するための文
- 書式と表示すべき変数名を書く
  - 書式： 浮動小数データを表示する場合、書式は「%f」と書くことになっている
  - 変数名： 変数名の前には「&」を付けない (scanf とは違う)

# いろいろな出力

```
printf( "x= ?" );
```

メッセージ 「x= ?」 の  
表示

```
printf( "x= %f" );  
printf( "y= %f" );
```

「x= 10.0000  
y= 20.0000」  
のように、メッセージ  
と変数の中身を並べて  
表示

¥n



次の行に進め（**改行**）という指示

**printf** 文などの中で用いる

(例)

```
printf("menseki=%f¥n", menseki);
```

# 浮動小数データの使い方

- 変数宣言 :

```
double teihen;
```

```
    double takasa;
```

```
    double menseki;
```

- 書式 :

**%lf** – scanf (入力) での書式

**%f** – printf (出力) での書式

## 例題 3 . sin 関数による三角形の面積



- 三角形の2辺の長さ **a, b** とその挟角 **theta** を読み込んで, 面積 **S** を計算するプログラムを作る
  - 面積を求めるために, sin関数を使う
  - 円周率 $\pi=3.14159$  とする

$$S = \frac{1}{2} ab \sin \theta$$

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double a;
    double b;
    double theta;
    double S;
    printf("a=");
    scanf("%lf", &a);
    printf("b=");
    scanf("%lf", &b);
    printf("theta=");
    scanf("%lf", &theta);
    S = 0.5 * a * b * sin( theta * 3.14159 / 180.0 );
    printf("S = %f¥n ", S );
    return 0;
}
```

入力部分

計算部分

出力部分

# ライブラリ関数



- 指数, 対数, 平方根
  - exp 指数関数 (eを底とする指数 z の累乗, eのz乗)
  - log 対数関数 (底をeとする自然対数の計算)
  - sqrt 平方根
- 三角関数
  - acos 逆コサイン
  - asin 逆サイン
  - atan 逆タンジェント
  - cos コサイン
  - sin サイン
  - tan タンジェント
- その他
  - fabs 絶対値
  - fmod(x,y) 浮動小数データの剰余
  - pow(x,y) べき乗 (xのy乗)



# いろいろな計算

```
y = sin( x );
```

$\sin x$  を計算し,  $y$  に格納

```
y = sqrt( x );
```

$\sqrt{x}$  を計算し,  $y$  に格納

```
d = sqrt( ( x * x ) + ( y * y ) );
```

$\sqrt{x^2 + y^2}$  を計算し,  $d$  に格納

# ライブラリ関数の利用



- 計算に関するライブラリ関数を利用するには,

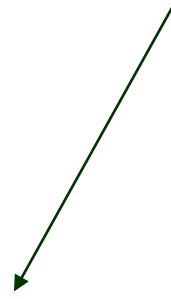
を, プログラムの先頭部分に書くこと

```
#include <math.h>
```

# 三角関数では「ラジアン」を使う

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double degree;
    double s;
    printf("degree=");
    scanf("%lf", &degree);
    s = sin( degree * 3.14159 / 180.0 );
    printf("sin(%f) = %f¥n ", degree, s );
    return 0;
}
```

「度」から「ラジアン」への変換



- 180.0 の「.0」には意味がある（浮動小数での計算を行うべきであることをコンピュータに教えている）

## 課題 1 . Heron の公式

- 三角形の 3 辺の長さ  $a, b, c$  を読み込んで, 面積  $S$  を計算するプログラムを作りなさい.
  - Heronの公式を用いること

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{但し } s = (a+b+c)/2$$

## 課題 2. 四則演算



- 2つの数を読み込んで、和、差、積、商、剰余を計算するプログラムを作りなさい