

URL:
http://www.yuasa.kuis.kyoto-u.ac.jp/~yuasa/index_J.html

「計算機科学概論」講義資料 (パスワードが必要)

ユーザー名: programming

パスワード: nyuumon

第1回 (第2章) PostScript 版 PDF 版

第2回 (第3章) PostScript 版 PDF 版

第3回 (第4章) PostScript 版 PDF 版

第4回 (第5章) PostScript 版 PDF 版

第2章 C言語の基礎

2.1 関数 (function)

数学の問題

x を整数とすると、関数 $f(x)$ を次のように定義する。

$$f(x) = 2x^2 + 3x + 1$$

このとき $f(4)$ の値を求めよ。

f は関数名

x はパラメータ (parameter)

$2x^2 + 3x + 1$ は関数の本体

$f(4)$ の 4 は実引数 (argument)

$2 \times 4^2 + 3 \times 4 + 1 = 45$ が $f(4)$ の値

C言語では

```
int f(int x)
{
    return 2*x*x + 3*x + 1;
}
```

int は整数 (integer) の意味

波括弧 (中括弧) で囲まれた部分が関数の本体

return 以降の式が関数の値を計算する

1

2

画面に “Ans.45” と書かせる関数

```
main()
{
    int value;

    value = f(4);
    printf("Ans.:%d\n", value);
}
```

本体の構成:

1. 変数宣言 . value は整数変数
2. 関数呼出しと代入
 $f(4)$ の値を求めて value の値とする。
つまり、関数 f を実引数 4 で呼び出して、 f の返す値を value に代入する。
3. ライブラリ関数の呼出し
ライブラリ関数 printf を呼び出して

Ans. (value の値)

を表示する。

コンパイルと実行

```
% cat saisho.c
int f(int x)
{
    return 2*x*x + 3*x + 1;
}

main()
{
    int value;

    value = f(4);
    printf("Ans.:%d\n", value);
}
% cc -o saisho saisho.c
% saisho
Ans.45
%
```

実行が始まると、まず main が呼び出される。

プログラムの関数が main 1 つだけでもかまわない。

```
% cat hello.c
main()
{
    printf("Hello, world.\n");
}
%
```

3

4

値が重要でない関数

```
void sisoku(int a, int b)
{
    int wa, sa, seki, sho, amari;

    wa    = a + b;
    sa    = a - b;
    seki  = a * b;
    sho   = a / b;
    amari = a % b;

    printf("a = %d, b = %d\n", a, b);
    printf("a + b = %d\n", wa);
    printf("a - b = %d\n", sa);
    printf("a * b = %d\n", seki);
    printf("a / b = %d ... %d\n",
           sho, amari);
}
```

sisoku(11, 3); と呼び出すと

```
a = 11, b = 3
a + b = 14
a - b = 8
a * b = 33
a / b = 3 ... 2
```

5

定数

● 整数定数

```
1234    10 進数
02322   8 進数
2322(8) = 2 × 83 + 3 × 82 + 2 × 8 + 2
         = 1234(10)
0x4d2   16 進数
4d2(16) = 4 × 162 + 13 × 16 + 2
         = 1234(10)
```

● 浮動小数点 (floating-point number) 定数

実数を表現する .

3.14, 0.5, 2.71828

1.234e2 は 123.4 と同じ ($1.234 \times 10^2 = 123.4$)

1e-3 は 0.001 と同じ ($1 \times 10^{-3} = 0.001$)

● 文字定数

```
'a'   小文字の a
'\n'  改行文字
'\t'  タブ
'\'   エスケープ文字
'\'   シングルクォート
```

7

2.2 字句構造 (Cプログラムの構成要素)

1. 識別子 (identifier)

```
f      x      main  value printf
```

2. キーワード (keyword)

```
int    return
```

3. 定数 (constant)

```
2      3      1      4
```

4. 文字列リテラル (string literal)

```
"Ans.%d\n"
```

5. 演算子 (operator)

```
*      +      =
```

6. 区切り記号 (separator)

```
{      ;      }      ,      (      )
```

7. 空白 (white space)

```
int f(int x)
{
    return 2*x*x + 3*x + 1;
}
main()
{
    int value;

    value = f(4);
    printf("Ans.%d\n", value);
}
```

6

2.3 変数と型

変数 (variable) ... データを格納するための一種の箱

変数の宣言 ... 名前のついた箱を用意する

```
int yonjou(int x)
{
    int y;

    y = x*x;
    return y*y;
}
```

変数には型 (type) という属性がついている .

C言語の基本的な型

```
char   文字型
int    整数型
float  単精度の浮動小数点数型
double 倍精度の浮動小数点数型
```

printf の主要な変換文字 (conversion character)

%d 整数の 10 進 (decimal) 表示

%o 整数の 8 進 (octal) 表示

%x 整数の 16 進 (hexadecimal) 表示

%c 文字 (character) の表示

%s 文字列 (string) の表示

%f 倍精度浮動小数点数の表示

8

2.4 コマンド引数

sisoku を呼び出すプログラム

```
% cat sisoku.c
main(int argc, char *argv[])
{
    int a, b;

    sscanf(argv[1], "%d", &a);
    sscanf(argv[2], "%d", &b);

    sisoku(a, b);
}
% cc -o sisoku sisoku.c
% sisoku 11 3
a = 11, b = 3
....
```

argc はコマンド引数の個数

argv[n] はコマンド引数の個々の文字列

```
argv[0]  "sisoku"
argv[1]  "11"
argv[2]  "3"
```

間違ったコマンド引数に対処する .

```
% cat sisoku.c
main(int argc, char *argv[])
{
    int a, b;

    if (argc != 3)
        printf("Usage: sisoku n1 n2\n");
    else {
        sscanf(argv[1], "%d", &a);
        sscanf(argv[2], "%d", &b);
        if (a <= 0)
            printf("%d not positive\n", a);
        else if (b <= 0)
            printf("%d not positive\n", b);
        else
            sisoku(a, b);
    }
}
% sisoku 11
Usage: sisoku n1 n2
% sisoku -11 -3
-11 not positive
% sisoku 11 -3
-3 not positive
% sisoku 11 3
a = 11, b = 3
a + b = 14
....
```

9

10

2.5 文 (statement)

これまでに登場した文

1. 式文

《式》;

```
value = f(4);
printf("Usage: sisoku n1 n2\n");
```

2. if 文

```
if (《式》) 《文》 else 《文》
if (《式》) 《文》

if (a <= 0)
    printf("%d not positive\n", a);
else if (b <= 0)
    printf("%d not positive\n", b);
else
    sisoku(a, b);
```

C 言語では、条件は整数で与える .

整数の 0 が偽 (false) , それ以外は真 (true)

3. 複合 (compound) 文

```
{ 《宣言》 ... 《宣言》 《文》 ... 《文》 }
```

4. return 文

```
return 《式》;
```

2.6 演算子 (operator)

単項 (unary) 演算子

前置 (prefix) 演算子

後置 (postfix) 演算子

2 項 (binary) 演算子

算術演算子

$-e$ 正負の符号反転

e_1+e_2 足し算

e_1-e_2 引き算

e_1*e_2 かけ算

e_1/e_2 割算の商

$e_1\%e_2$ 割算の余り . オペランドは整数 .

代入演算子

$e_1=e_2$ e_2 を e_1 に代入

$e_1+=e_2$ e_1+e_2 を e_1 に代入

$e_1-=e_2$ e_1-e_2 を e_1 に代入

...

注意

$x = y = b + 10$ は

$x = (y = b + 10)$ と等価

$a -= b$ は $a-b$ を a に代入

$a =- b$ は $-b$ を a に代入

11

12

インクリメント (increment) 演算子

`e++` e を 1 増やす
`++e` e を 1 増やす
`e--` e を 1 減らす
`--e` e を 1 減らす

x の値が 5 のとき

`y = x++` で y は 5
`y = ++x` で y は 6

関係演算子

`e1<e2` $e_1 < e_2$ かどうか
`e1>e2` $e_1 > e_2$ かどうか
`e1<=e2` $e_1 \leq e_2$ かどうか
`e1>=e2` $e_1 \geq e_2$ かどうか
`e1==e2` $e_1 = e_2$ かどうか
`e1!=e2` $e_1 \neq e_2$ かどうか

条件が成り立てば 1, 成り立たなければ 0.

論理演算子

`!e` 論理否定. e が真なら偽, 偽なら真.
`e1&&e2` 論理積. 両方真のときのみ真
`e1||e2` 論理和. 少なくとも片方が真のときのみ真

注意

`10 <= x < 20` は
`(10 <= x) < 20` の意味で常に真.
`10 <= x && x < 20` と書くこと.
 ‘<=’ を ‘=<’ と書かないように.

`if (a == b) 《文》` を
`if (a = b) 《文》` と書いてはいけない.

13

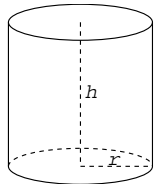
14

2.7 数学ライブラリ

関数名	機能
<code>ceil(x)</code>	x 以上の最小の整数 $\lceil x \rceil$
<code>floor(x)</code>	x 以下の最大の整数 $\lfloor x \rfloor$
<code>fmod(x,y)</code>	x/y の剰余
<code>fabs(x)</code>	絶対値 $ x $
<code>sqrt(x)</code>	平方根 $\sqrt{x}, x \geq 0$
<code>exp(x)</code>	指数関数 e^x
<code>log(x)</code>	自然対数 $\ln x, x > 0$
<code>log10(x)</code>	常用対数 $\log_{10} x, x > 0$
<code>pow(x,y)</code>	べき乗 x^y
<code>sin(x)</code>	正弦 $\sin x$, x の単位はラジアン
<code>cos(x)</code>	余弦 $\cos x$, x の単位はラジアン
<code>tan(x)</code>	正接 $\tan x$, x の単位はラジアン
<code>asin(x)</code>	正弦の逆関数 $\arcsin x, -1 \leq x \leq 1$
<code>acos(x)</code>	余弦の逆関数 $\arccos x, -1 \leq x \leq 1$
<code>atan(x)</code>	正接の逆関数 $\arctan x$
<code>atan2(y,x)</code>	(y/x) の正接の逆関数 $\arctan(y/x)$
<code>sinh(x)</code>	双曲線正弦 $\sinh x$
<code>cosh(x)</code>	双曲線余弦 $\cosh x$
<code>tanh(x)</code>	双曲線正接 $\tanh x$

数学ライブラリを使うときは,

```
#include <math.h>
をプログラムの先頭に書いておく.
```



高さが h , 体積が V の円柱の底面の半径

$r = \sqrt{\frac{V}{\pi h}}$ を求める

```
#include <math.h>

double column(double h, double V)
{
    return sqrt(V / (M_PI * h));
}
```

リンク時にライブラリもリンクする.

```
% cc -o column column.c
ld: Undefined symbol
    _sqrt
% cc -o column column.c -lm
%
```

15

16