
プログラミング基礎演習

第1回

C言語編

【コンパイル，制御構文】

長谷川禎彦

この講義について

- 内容はソフトウェアの授業とはほとんどリンクしていません
 - 前半でC言語, 後半でPythonを行います
 - 本講義を履修する際は, 「ソフトウェア1, (2)」の履修が原則必須です.
-

評価

出席点(小課題)



中間レポート

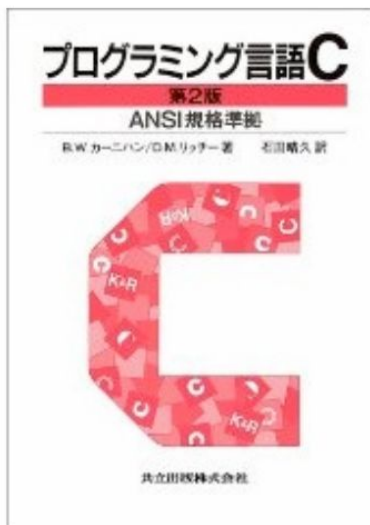


最終レポート

プログラミング基礎演習としては教科書は
指定しない

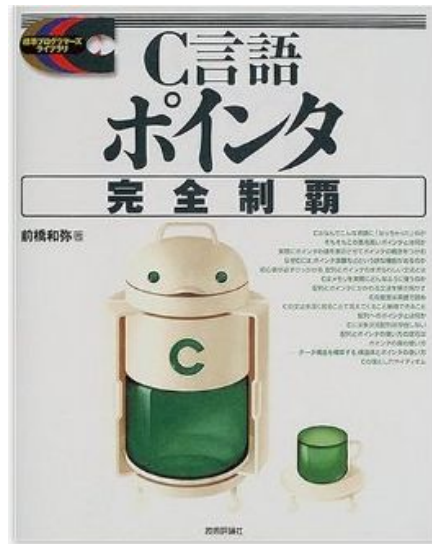
バイブル的本. でも
難しい.

初心者に
おすすめ!



プログラミング言語C ANSI規格準拠 (単行本)

[B.W. カーニハン](#) (著), [D.M. リッチー](#) (著), [石田 晴久](#) (翻訳)



C言語ポインタ完全制覇 (単行本)

前橋和弥 (著) 出版社: 技術評論社 (2001/01)



Cの絵本—C言語が好きになる9つの扉 (単行本)

アンク (著) 出版社: 翔泳社 (2002/03)

C言語の最重要点

- ポインタの理解

- ポインタの概念はほとんど全ての手続き型言語に存在.
Java, Ruby, Scalaなどは事実上ポインタしかない.
- ただし, ポインタとは言わずに「参照」という

- ここが理解できるなら, 簡単な教科書でも良い

出席

- 小課題を毎回提出
 - 締め切りは日曜の23:59
 - <https://goo.gl/QJ3LMP>
 - スライドのパスワード
 - program2017
-

課題提出

- <https://goo.gl/QJ3LMP>

Click here



小課題・レポート提出

小課題・レポートは「提出フォーム」より提出してください。「課題」選択で「小課題」か「レポート」を正しく選んでください。複数回提出した場合は、期限内の一番最後のものが採用されます。

- 提出フォーム
 - 提出は時間厳守です。遅れた場合は減点します。
 - 2017年2月19日締め切り小課題の提出者リスト（10分ごとに自動更新）
-

Windowsで課題を行うには

- Windowsにコンパイラを導入する
 - Cygwinというソフトウェアをインストールすると
、 gcc, emacs, gdb等が使える。
 - CygwinはWindows上にLinuxのような環境を構築するソフト
 - インストールの詳細は授業のホームページ
(<https://goo.gl/QJ3LMP>)に記述してある
 - Visual Studioなどが使える人はそれで行っても良い
-

Hello world

次のプログラムを実行してみよう

```
#include <stdio.h>
int main() {
    printf("Hello World\n");
    return 0;
}
```

mainの返値・引数

- 実行時パラメータなし (./a.out のような実行)
 - `int main()` または `int main(void)`: 正しい形式。
 - `main()`: こういう書き方も時々見かける。
- 実行時パラメータあり (./a.out 10 のような実行)
 - `int main(int argc, char *argv[])`
- `main`関数の返値が0: 正常終了

手順

- Emacsでソースコードを書くと後々便利(3年の課題でもよく使う)
 - もちろん他のエディターで書いても良い(e.g. vim)
 - kadai01.cという名前で**保存**
 - **コンパイル**する
 - ターミナルで
 - `$ gcc kadai01.c` (注意: \$は打たない。今後も同様)
 - というコマンドを打ってコンパイル
 - **実行**
 - ターミナルで
 - `$./a.out`
 - と打つと実行
-

補足

- 出力するファイル名を指定する
 - `$ gcc -o kadai01 kadai01.c`
 - とすると、出力ファイルがa.outではなく、kadai01となる
- ターミナルでのファイル操作コマンド
 - ファイル移動: `mv`
 - 例.リネーム: `mv name1 name2`
 - name1 : 元, name2 : 後
 - コピー: `cp`
 - ファイル一覧: `ls`

Emacsの簡単な操作法

- `ctrl+x ctrl+f`でファイル名を指定して開く
 - `ctrl+x`を押した後, `ctrl+f`を押すという意味
 - 存在しないファイル名の場合は新規作成
 - `ctrl+x ctrl+s` で保存
 - `ctrl+x ctrl+c` で終了
 - 間違ったときは`ctrl+g`で入力中のコマンドを消す
 - Undo: `Ctrl+x u`
-

Emacsの簡単な操作法

- emacsからコンパイル
- M-x compile と打つ
 - M-xはEscapeキーを押してからx
 - M-xを押した後, compileと書く
- 表示されるmake -kなどの文字を消してgcc -o kadai01 kadai01.c と書いてEnter
- コンパイルエラーの場所にはctrl + x ` (バッククオート記号. バッククオートはShift + @で入力) で移動出来る

Emacsの簡単な操作法

- コピーペースト
 - コピーはctrl + spaceで開始位置をセット
 - カーソルを移動してM-wでコピー
 - ctrl + yでペースト
 - バッファの消去 ctrl + x, k
 - バッファの2画面を1画面にする ctrl + x, 0
 - バッファを切り替える ctrl + x, b
-

はまりやすい罠

- ソースコードは半角で記述する(全角文字は使わない)
 - × `#include <stdio. h>`
 - ○ `#include <stdio.h>`
- 全角を使って良いのは文字列中だけ
 - ○ `printf(“本日は晴天なり”)`
- 大文字と小文字は区別される(別の意味になる)
 - × `int Main()`
 - × `Int main()`
 - ○ `int main()`

はまりやすい罠

- 行末のセミコロン「;」を必ずつける
 - `printf("Hello World\n");` ← 忘れがち
- {} や () の対応関係
 - 閉じ忘れることがある
 - 長いif文などは特に注意する

printf関数について

- printfは標準出力(stdout)に出力するための関数
 - stdio.hの中に定義されている
 - .hファイルはヘッダーファイルと呼ぶ
- #includeでstdio.hをインクルードしておかないとprintfは使えない
- printf("Hello World\n");
 - Hello Worldと書いて改行(\n)
- printf("%d + %d = %d", a, b, c);
 - a,b,c(整数)を左から順番に%dに入れて表示

変数の型

- すべての変数には型がある
- よく用いるのは, int, double, char
 - intは整数
 - doubleは実数
 - charは文字列に用いる
- 数学では $\mathbf{Z} \subseteq \mathbf{R}$ であるが, \mathbf{R} のほうがメモリ, 計算が余分に必要で無駄が多い
- 整数であることが保証される
 - n番目, などではnは整数である必要がある

課題1

課題1: 次のプログラムをコンパイルできるようにする

```
#include <stdio.h>
int main() {
    int a = 10;
    int b = 100
    c = a * b;
    printf("%d * %d = %d\n", a b c);
    return 0;
}
```

注意点

- “=”は代入
 - 等号ではない！ $x = x + 1$ は x に 1 たしたものを x に代入している
- 等号は“==”

例

```
if (x == 10) {  
    printf("ten\n");  
}
```

x が 10 の時だけ, "ten" と表示する

for文

- 繰り返し

```
for (初期化式; 条件式; カウンタ変数更新) {  
    文;  
}
```

1. 初期化式を実行する
 2. 条件式を評価する. 条件が偽ならば終了する
 3. 文を実行する
 4. カウンタ変数の更新を実行する
 5. 条件式の評価に戻る
-

for文

1から100まで足す

```
#include <stdio.h>
int main() {
    int sum=0;
    int i;
    for(i=1;i<=100;i++) {
        sum=sum+i;
    }
    printf("%d\n",sum);
    return 0;
}
```

i++は $i = i + 1$ と同じ. つまり i を一つ進める

sumにsum+iを代入する

if文

- 条件分岐

```
if (条件式) {  
    文1  
}  
else {  
    文2  
}
```

- 条件式を評価する. 条件が真ならば, 文1を実行, 条件が偽ならば, 文2を実行
 - elseはない場合もある
-

if文の例

0~99で偶数だけを表示する

```
#include <stdio.h>
int main() {
    int i = 0;
    for (i = 0; i < 100; i++) {
        if (i % 2 == 0) {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

$n \% m$ は n を m で割った余りを表す.

if文におけるelse if

```
if (条件1) {  
    文1  
}  
else if (条件2) {  
    文2  
}  
else if (条件3) {  
    文3  
}  
else {  
    文4  
}
```

上から条件式を評価。条件nが真となった場合、文nを実行し、外に出る(それ以降の条件や、elseの文は実行しない)。全ての条件が偽の場合はelseの文が実行される。

課題2

for文とif文を用いて, $n = 158340421$ の3乗根を計算しprintfで表示せよ. ループの上限は適当に決めよ.

```
#include <stdio.h>
int main() {
    int n = 158340421;
    /*ここを埋めてみる*/
    return 0;
}
```

$$3\text{乗根} = \sqrt[3]{n}$$

ヒント

- $i=1, 2, 3, \dots$ と順に3乗して, n と一致するかを確かめる.
- 正しく計算できれば, 細かな実装の差異は全く問題無い

課題3

ピタゴラス数を列挙せよ

(ピタゴラス数は以下を満たす自然数 x, y, z)

$$x^2 + y^2 = z^2$$

$x < 1000, y < 1000, z < 1000$ まで表示させてみよ.
ひな型はないので自由にプログラムせよ.

課題3

出力例

$3^2 + 4^2 = 5^2$
 $5^2 + 12^2 = 13^2$
 $6^2 + 8^2 = 10^2$
 $7^2 + 24^2 = 25^2$
 $8^2 + 15^2 = 17^2$
 $9^2 + 12^2 = 15^2$
 $9^2 + 40^2 = 41^2$
 $10^2 + 24^2 = 26^2$

...

x^n は、プログラミング言語やlatexでxのn乗を表すのによく使う。ただし、C言語では使えない！（xorを計算してしまう）

なお、重複がないように出力せよ

（つまり、 $4^2+3^2=5^2$ などは表示させない）

課題提出方法

- 課題の答えをkandai01.c~kandai03.cファイルに書く.
- ファイルをzipファイルにまとめる
 - ファイル名:学籍番号.zip
 - 例:学籍番号が641234の場合, 641234.zipとする.
- 学籍番号.zipファイルのみを提出する
 - 提出は <https://goo.gl/QJ3LMP>
 - フォームの「課題」から「小課題」を選択