C言語プログラミング 基本例題88+88

冨永和人【編著】

生野壮一郎・菊池眞之
黒川弘章・関口暁宣【共著】

コロナ社

ま　え　が　き

本書はC言語プログラミングの基本例題集です。C言語の基本的な機能について，88問の例題と解答例を示し，解答例として与えたプログラムを解説しました。例題とプログラムの実例を通してC言語を自学したいとお考えの方のため，またC言語プログラミングの入門から中級レベルの講座での利用に向けて，C言語の基本機能を網羅した分かりやすい例題集をと考えて作成しました。各例題の解説の後には，扱ったトピックに関する注意点やテクニックを「ポイント」としてまとめて示し，さらに「発展」として，少し進んだ問題や，関連する問題を88問設けています。発展問題には解答例を示していないので，自学の読者には力試しとして，また講座で使われる先生には出題の参考として利用いただけると思います。

章の並びは，C言語の機能に対する理解を基本から積み上げられるような順序としてあります。このため，1章から順に例題を解いていくのが最も効率的でしょう。他のテキストなどと並行して利用される場合のために，各章で扱っているトピック間の順序関係を下図に示します。矢印に沿った順ならば無理なく例題に取り組めると思います。なお個々の例題の中には，前にある他章のトピックにあまり依存していないものもありますので(特に8章の外部変数と静的変数についての例題や，11章のアルゴリズムに関する例題など)，早めに挑戦していただくことも可能でしょう。

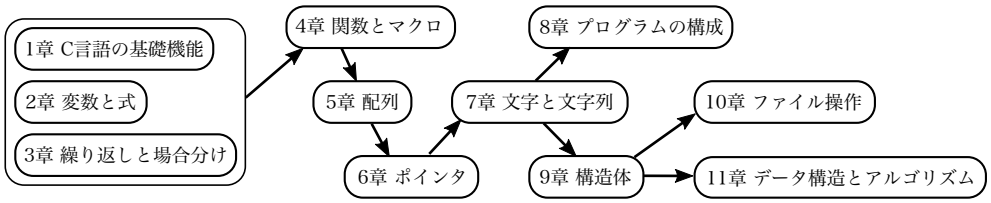


図　各章で扱っているトピック間の順序関係

本書の例題および内容は，東京工科大学における20年以上のC言語教育の経験に基づくものです。大学関係者の皆様，また授業について有益なフィードバックを下さった学生，卒業生，教育助手の皆様に深く感謝します。今も実用的な言語であるC言語を学ぶ方々の知識と技術の向上に，本書が役立つことを切に願います。

2017年1月

著　者　一　同

本書の使い方

本書の各章は基本的に，C 言語が持つ 1 つの大きな機能を扱います。本書はコンパクトな例題集とするために，読者が各機能について講座や本書以外のテキストなどである程度学んだ上で，本書の該当する章の例題に取り組まれることを想定しています。章のはじめには各機能の概略をまとめてあります。例題に入る前の知識の確認に役立てて下さい。

それぞれの章の各節は，**例題**，**考え方**，**解答例**，**解説**，**ポイント**，**発展**という構成になっており，必要に応じて**実行例**も示しました。自学される読者は，はじめに**例題**を見て解き方を考え，もしも難しければ**考え方**を参考にして，プログラムを自分で作って下さい。その後で**解答例**を見つつ**解説**を読んで，どのような実装が可能なのか学んで下さい†。解答例として与えたプログラムには説明のために注釈（コメント）を多めに与えてあります。自分で打ち込んで動作させる際にはもちろん注釈は適当に省いて結構です。その例題で重要な点を**ポイント**としてまとめてあるので，見て理解を確認して下さい。ポイントはまた，後で復習する際の参考にもなるでしょう。より実用的なプログラミングに近い問題などを**発展**に配してあります。プラスアルファの理解のために挑戦してみて下さい。

講座で利用される先生には，受講者向けに例題そのものを出題して解かせたり，例題に対する解答例をプログラミングの実例として受講者に学習させ，発展を参考に課題を出すという方法などでご活用いただけると思います。

巻末には**引用・参考文献**として，C 言語そのものや，C 言語を用いたプログラミングの学習に役立つと思われる書籍を紹介しました。参考にしてもらえれば幸いです。

記　　法

本書で用いる記法は次の通りです。プログラムコードはタイプライタ体で記します（例：`int i;`）。コード中に何かを入れるべき場所は《》で示します（例：`int a[`《要素数》`]`）。

以下のような角のある矩形の囲みはプログラムやファイルを示します。

```
#include <stdio.h>

int main(void) {
    printf("こんにちは!\n");
    return 0;
}
```

† 解答例として示したプログラムは，すべて本書のホームページからダウンロードできます。アドレスは `http://www.coronasha.co.jp/np/isbn/9784339028737/` です（2017 年 1 月現在）。

実行例は以下のように四隅が丸い囲みで示します。コマンドラインのプロンプトは $ とし，ユーザの入力は下線で示します。⏎ はエンターキーを押すことを表します。

```
$ ./a.out⏎
こんにちは!
$
```

ユーザの入力について

ユーザの入力を扱うような例題を解く場合には，特に指定がない限り，その入力に妥当な仮定を置いて，想定内の入力だけがくるとして構いません。例えば氏名をキーボードから入力するような例題に対してプログラムを作るときには，その上限を適当に数十文字として，それより長い氏名をユーザは打ち込まないと仮定して結構です。解答例として与えたプログラムもそのようにしてあります。なお，実用的なプログラムを作る際には一般に，ユーザのどのような入力も適切に扱う必要があるので注意して下さい。

文字について

C 言語で文字を扱う基本機能は 1 バイト文字（いわゆる半角英数記号）を対象としており，ひらがなや漢字などの（いわゆる全角）文字の取り扱いは基本の範囲を超えます。しかしながら，設問や解答例にある文などがローマ字では分かりにくいので，本書のプログラムでは出力する文字列と注釈に限ってかな漢字を使っています。もしもかな漢字が原因で，使っている環境でプログラムが正しくコンパイルできなかったり，出力が正しく表示されないようであれば，それらを 1 バイト文字で置き換えて，`printf("こんにちは!\n")` を `printf("Hello!\n")` とするなどして下さい。ただし，キーボードや外部ファイルからの入力はすべて 1 バイト文字を想定しています。特に断りのない限り，「文字」は 1 バイト文字，「文字列」は 1 バイト文字からなる文字列と考えて下さい。

扱う 1 バイト文字の文字コードとしては，ASCII という規格から派生したものを仮定します。現在パソコンなどで広く用いられている文字コードです。ただ，そのような文字コードの間にわずかに違いがある場合があります。本書のプログラムで使っているバックスラッシュ（\）という文字が入力できない場合には，円記号（¥）を代わりに使って下さい。同じくチルダ（~）が入力できなければ代わりにオーバライン（‾）を使って下さい。

処理系と環境

C 言語プログラムのコンパイルと実行を行うためのソフトウェアを C 言語の**処理系**といい，処理系や作成したプログラムを動かす場となるシステムを**環境**といいます。読者が

パソコンをお使いであれば，環境とはパソコンと OS のことで[†1]，処理系とはコンパイラやライブラリのことだと思っていただければおよそ間違いないでしょう。

本書の例題では，コマンドラインインタフェース（CLI）で実行するようなプログラムを作ります。CLI での入出力は最も簡単で，学習に適しているためです。Linux や macOS などの UNIX 系の環境ならターミナルで，Windows 環境ならコマンドプロンプトでプログラムを実行します。実行例としては UNIX 系環境での様子を示しました。プログラムの実行方法，実行中断の方法，キーボード入力の終了を知らせる方法などは環境によって違うので，UNIX 以外の環境を利用する場合には使うシステムの説明を参照して下さい。

C 言語のプログラムを動かすには，まずプログラム（ソースファイル）を作成し，次にそれをコンパイルして実行可能プログラムに翻訳し，実行します。テキストエディタでソースファイルを作成し，CLI でコマンド（cc や gcc や cl などで，処理系による）を入力してコンパイルして実行するのが最も容易でしょう。コマンドが cc でソースファイルが hello.c の場合，UNIX 系環境では以下のようにコンパイルします。

```
$ cc hello.c⏎
```

統合開発環境（IDE）を利用して作成とコンパイル（ビルド）を行ってから，CLI で実行するということも可能でしょう。IDE に CLI のエミュレーション機能があれば，IDE 内で実行もできます。簡単なプログラムの作成から実行までをウェブ上で行うサービスもあります。ただし IDE やウェブの場合，本書の一部の例題で用いている実行コマンドへの引数の指定や入力の切り換えなどに制約がある場合もあります。

C 言語の規格

C 言語にはいくつかの規格があります。本書で示すプログラムの例は原則として，2011 年に作られた C11 と呼ばれる規格に定められたうちの基本的な機能を使って作られています[†2]。そのため現在利用されている多くの処理系でコンパイルでき，動作するでしょう。

規格に従うと煩雑になるような用語については，より分かりやすい一般的に受け入れられていると思われる語を使いました。例えば，外部定義されたオブジェクトは「外部変数」，static を使って宣言されたオブジェクトは「静的変数」などとしました。

---

†1　統合開発環境とは別のものなので注意して下さい。統合開発環境はむしろ処理系にあたります。

†2　ただし，プログラムの動作を面白くするために，指定された秒数だけ停止する sleep という UNIX 系の処理系の機能を使う部分が 2 箇所ほどあります（4.5 節，8.2 節の発展）。これについてはその箇所で明示してあり，この機能を使わなくても問題なく学習できます。

目　　次

1. C 言語の基礎機能

2. 変　数　と　式

3. 繰り返しと場合分け

4. 関数とマクロ

5. 配 列

6. ポ イ ン タ

7. 文字と文字列

8. プログラムの構成

9. 構 造 体

10. ファイル操作

11. データ構造とアルゴリズム

1 C 言語の基礎機能

C 言語はさまざまな機能を持っています。それらを使ってプログラムを作成する際に共通に必要となる基礎的な機能には以下のものがあります。

- 画面出力，キーボード入力
- 変数を使った計算
- プログラムの動作の制御（場合分けや繰り返し）
- 標準的なライブラリ機能の利用

本章では，これらを使った例題を示します。

1.1 文字列の出力 —— メッセージを表示する

例題 1.1 以下のように画面にメッセージを表示するプログラムを作成せよ。

```
$ ./a.out⏎
Hello, world
Bye!
$
```

考　え　方

画面に文字列を出力するには **printf** を使います。

解　答　例

プログラム 1-1

```
1  #include <stdio.h>
2
3  int main(void) {
4      printf("Hello, world\n");
5      printf("Bye!\n");
6      return 0;
7  }
```

解　　説

1 行目はヘッダ **stdio.h** を取り込む指示です。このヘッダは C 言語の標準的な入出力機能を使うためのもので，このプログラムでは printf を使うために取り込みました。

Hello, world を表示して行を変えるために**改行文字** \n を出力しています（4 行目）。それぞれの行を出力するため printf を 2 つ書いていますが（4〜5 行目），以下のように " " の中に \n を 2 つ書けば，1 つの printf で 2 行表示できます。

```
printf("Hello, world\nBye!\n");
```

6 行目にある **return 文**「return 0;」はプログラムの実行を終了させる文です。

ポ イ ン ト

☞　画面出力で改行するには改行文字 \n を使う。

発　　展

printf を一度だけ使って，以下のように画面に表示するプログラムを作成せよ。出力 2 行目の先頭の空きは空白 8 個分，3 行目は 24 個分である。

```
$ ./a.out↵
To be, or not to be:
        that is the question.
                        -- Hamlet
$
```

▶ 1.2　数値の出力 —— 数値データを表示する ◀

例題 1.2　次のような変数（日付，降水確率，最高気温，最低気温を表す）を用意し

- int 型の変数：y1，y2，m1，m2，d1，d2，prec1，prec2
- double 型の変数：high1，high2，low1，low2

以下に示す値を代入してから

```
y1 = 2020; m1 = 2;  d1 = 4;  prec1 = 0;  high1 = 9.59; low1 = -10.1;
y2 = 2020; m2 = 10; d2 = 20; prec2 = 30; high2 = 15.3; low2 = 5.51;
```

それらの変数の値を以下のように画面に表示せよ。下の 2 行については，各欄の桁を示した通りに揃えること。分かりやすいように空白 1 つを ␣ で示してある。

```
$ ./a.out↵
最高␣15.300000，最低␣-10.100000
2020/02/04␣␣␣9.590000␣-10.100000␣␣0%
2020/10/20␣␣15.300000␣␣␣5.510000␣30%
$
```

考　え　方

printf で変数の値などのデータを表示するには，printf に与える**書式**（カッコ内の最初に書く " " でくくられたもの）の中で，適切に**変換**指定を行います。

解　答　例

プログラム 1-2

```
 1  #include <stdio.h>
 2
 3  int main(void) {
 4      int y1, m1, d1, prec1;
 5      int y2, m2, d2, prec2;
 6      double high1, low1, high2, low2;
 7
 8      y1 = 2020; m1 = 2; d1 = 4; prec1 = 0; high1 = 9.59; low1 = -10.1;
 9      y2 = 2020; m2 = 10; d2 = 20; prec2 = 30; high2 = 15.3; low2 = 5.51;
10
11      printf("最高 %f, 最低 %f\n", high2, low1);
12      printf("%4d/%02d/%02d %10f %10f %2d%%\n", y1, m1, d1, high1, low1, prec1);
13      printf("%4d/%02d/%02d %10f %10f %2d%%\n", y2, m2, d2, high2, low2, prec2);
14      return 0;
15  }
```

解　　説

データを文字列として出力するための変換を printf に指定します。int 型データの出力には **d 変換**を用います（12〜13 行目）。表示領域の最小幅を %《幅》d のように指定すると，データはその領域に右詰めで表示されます。データの桁数が少なくて表示領域に満たない場合にできる左側（上位桁）の空きをゼロ詰めするには %0《幅》d と指定します。double 型データは **f 変換**で出力します（11 行目）。表示領域の最小幅と小数点以下の桁数（**精度**）を指定するには %《幅》.《精度》f とします（12〜13 行目）。幅は小数点を含む全体の文字数で，精度の指定は省略すると 6 桁になります。ゼロ詰めの指定は %d と同様です。

% は変換指定の開始を意味する特殊な文字なので，% という文字そのものを表示するには %% と書きます（12〜13 行目）。

ポ　イ　ン　ト

☞　printf でデータを出力するには適切な変換を用いる。

発　　展

例題 1.2 のプログラムを改良し，データを以下のような表の形で出力するようにせよ。

```
$ ./a.out⏎
␣␣␣␣Date␣␣␣␣␣␣H␣␣␣␣␣L␣␣␣␣P
+----------+-----+-----+---+
```

```
|2020/02/04|␣␣9.6|-10.1|␣0%|
+----------+-----+-----+---+
|2020/10/20|␣15.3|␣␣5.5|30%|
+----------+-----+-----+---+
$
```

▶ 1.3　数値の入力 —— 10 進数を 16 進数に変換する ◀

例題 1.3　以下のように，入力した整数を 16 進数で表示するプログラムを作成せよ。

```
$ ./a.out⏎
decimal number? 255⏎
hexadecimal = ff
$
```

考　え　方

scanf で入力を読み取ります。16 進表示には printf の **x 変換**（%x）を使いましょう。

解　答　例

プログラム 1-3

```
 1  #include <stdio.h>
 2
 3  int main(void) {
 4      int n;
 5
 6      printf("decimal number? ");
 7      scanf("%d", &n);                  // 10進で入力
 8      printf("hexadecimal = %x\n", n);  // 16進で出力
 9      return 0;
10  }
```

解　説

int 型の変数 n を用意して（4 行目），scanf でキーボードからの入力を読み取ります（7 行目）。画面に何も表示せずに入力待ちになると何が起きているかユーザに分からないので，入力を促すメッセージ（**プロンプト**）を printf で表示します（6 行目）。

入力された文字列から変数にデータを得るために変換を指定します。10 進符号付き整数を表す文字列から int 型の数値を得るには d 変換を用います。書式に %d を指定して，コンマの後に &《int 型の変数》と書きます（7 行目）。これによって変数 n に数値が得られます。その数値を 16 進数として表示するために，printf の x 変換を使いました（8 行

目)。printf や scanf には他にも 8 進数を扱う **o 変換**など，さまざまな変換があります。

// から行末までは**注釈**で，プログラムの動作には関係ありません。この形式の注釈に対応していない処理系の場合には，/* … */ という注釈の形式を使って下さい。

ポ　イ　ン　ト

☞　scanf で変数に数値を読み取るときには，変数名に & をつける。

☞　printf や scanf にはさまざまな変換があるので適切に使い分けよう。

発　　展

入力された 2 つの 16 進数の積を求めて結果を 16 進数で表示するプログラムを作成せよ。

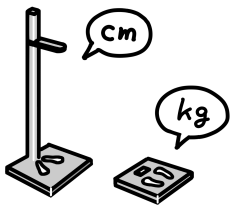
```
$ ./a.out↵
16進数を1つ入力して下さい: 100↵
16進数をもう1つ入力して下さい: dead↵
16進で 100 * dead = dead00 です
$
```

▶ 1.4　場合分けの基本（１） —— BMI で体型を判定する ◀

例題 1.4　身長と体重を入力すると，BMI と，肥満かどうかを表示するプログラムを作成せよ。BMI は次の式で求め，25 以上なら肥満，25 未満なら肥満でないとする。

$$\text{BMI} = \text{体重}\,[\text{kg}] \,/\, (\text{身長}\,[\text{m}] \times \text{身長}\,[\text{m}])$$

```
$ ./a.out↵
身長[cm]? 153↵
体重[kg]? 51↵
BMIは21.8，肥満ではありません
$ ./a.out↵
身長[cm]? 165↵
体重[kg]? 69↵
BMIは25.3，肥満です
$
```



考　え　方

BMI は実数値なので double 型で扱いましょう。合わせて身長と体重も double 型とします。肥満かどうかは，BMI の値を使って **if 文**で場合分けをして表示します。

索　　引

【A，B】

【C】

【D】

【E】

【F】

【G，I】

【L，M】

【N，O】

【P】

【R】

【S】

【T】

【V，W，X】

【数字】

【記号】

—— 編著者・著者略歴 ——

冨永　和人（とみなが　かずと）
1989年　東京工業大学工学部情報工学科卒業
1991年　東京工業大学大学院博士前期課程修了(情報工学専攻)
1994年　東京工業大学大学院博士後期課程単位取得退学(情報工学専攻)
1994年　東京工科大学講師
1996年　博士(工学)(東京工業大学)
2002年　東京工科大学助教授
2003年　イリノイ大学(米国)客員研究員
～04年
2007年　東京工科大学准教授
2012年　和情報網設立(代表)
現在に至る

菊池　眞之（きくち　まさゆき）
1994年　早稲田大学理工学部電気工学科卒業
1996年　大阪大学大学院基礎工学研究科博士前期課程修了(物理系専攻)
1996年　大阪大学大学院基礎工学研究科博士後期課程中退(物理系専攻)
1996年　大阪大学助手
1997年　大阪大学大学院助手
1999年　博士(工学)(大阪大学)
1999年　筑波大学助手
2003年　東京工科大学講師
現在に至る

関口　暁宣（せきぐち　あきのり）
1996年　東京大学工学部機械情報工学科卒業
1998年　東京大学大学院工学系研究科修士課程修了(機械情報工学専攻)
2002年　東京大学大学院工学系研究科博士課程修了(機械情報工学専攻)
博士(工学)
2002年　弘前大学助手
2007年　弘前大学大学院助教
2008年　東京工科大学講師
現在に至る

生野壮一郎（いくの　そういちそう）
1994年　山形大学工学部電子情報工学科卒業
1996年　山形大学大学院工学研究科博士前期課程修了(電子情報工学専攻)
1999年　筑波大学大学院工学研究科博士後期課程修了(電子情報工学専攻)
博士(工学)
1999年　東京工科大学講師
2006年　東京工科大学助教授
2007年　東京工科大学准教授
2016年　東京工科大学教授
現在に至る

黒川　弘章（くろかわ　ひろあき）
1993年　慶應義塾大学理工学部電気工学科卒業
1995年　慶應義塾大学大学院理工学研究科修士課程修了(電気工学専攻)
1997年　慶應義塾大学大学院理工学研究科博士課程修了(電気工学専攻)
博士(工学)
1997年　東京工科大学講師
2009年　東京工科大学准教授
現在に至る

C 言語プログラミング基本例題 88+88

C Programming : Basic Examples and Problems 88+88

2017 年 3 月 6 日　初版第 1 刷発行　★

検印省略

編著者　冨永和人
著　者　生野壮一郎
　　菊池眞之
　　黒川弘章
　　関口暁宣
発行者　株式会社コロナ社
　　代表者　牛来真也
印刷所　三美印刷株式会社
製本所　株式会社グリーン

112–0011　東京都文京区千石 4–46–10
発行所　株式会社　**コロナ社**
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　http://www.coronasha.co.jp

ISBN 978–4–339–02873–7　C3055　Printed in Japan　（松岡）