C言語プログラミング教材 - 演習問題集（基礎編）

第1章〜第6章の演習問題

C言語学習教材プロジェクト

2024年

Table of Contents

# はじめに

本書は、C言語プログラミング教材基礎編の演習問題集です。各章で学んだ内容を実践的に確認し、理解を深めるための問題を収録しています。

## 演習問題の構成

各章の演習問題は以下の構成になっています：

* **基礎問題**: 学習内容の基本的な理解を確認
* **応用問題**: 複数の概念を組み合わせた問題
* **発展問題**: より実践的な課題

## 学習の進め方

1. まず自分で問題を解いてみる
2. コンパイル・実行して動作を確認
3. 解答例と比較して理解を深める
4. 別解や改善点を考える

# 第1章: 導入・環境構築 - 演習問題

## 演習の目的

この演習を通して、C言語の開発環境が正しく構築されているかを確認し、基本的なコンパイル手順を習得します。

## 演習課題

### 基礎問題

#### 問題1: 環境構築確認

サンプルプログラム examples/environment\_check.c を使用して、以下の作業を実行してください。

1. **C90準拠でコンパイル・実行**

* make c90

1. **C99準拠でコンパイル・実行**

* make c99

1. **C11準拠でコンパイル・実行**

* make c11

1. **C17準拠でコンパイル・実行**

* make c17

1. **実行結果の確認**
   * 各規格でコンパイルしたプログラムの出力結果を比較してください
   * 規格による違いがあるかを確認してください

#### 問題2: コンパイラ情報の確認

以下のコマンドを実行して、開発環境の情報を調べてください。

1. **コンパイラバージョンの確認**

* gcc --version

1. **対応規格の確認**

* gcc -std=c90 --help  
  gcc -std=c99 --help  
  gcc -std=c11 --help  
  gcc -std=c17 --help

1. **プリプロセッサマクロの確認**

* echo | gcc -dM -E -

#### 問題3: 基本的なプログラム作成

solutions/ex1\_3\_hello\_name.c というファイルを作成し、以下の機能を実装してください。

**要求仕様:** - ユーザーに名前の入力を求める - 入力された名前を使って挨拶メッセージを表示する - C90準拠で記述する

**実行例:**

あなたの名前を入力してください: 田中  
こんにちは、田中さん！

### 応用問題

#### 問題4: 規格比較レポート

C90とC99の主な違いについて調べ、以下の点についてまとめてください。

1. **新しく追加されたデータ型**
2. **新しく追加された機能**
3. **プログラムの書き方で変わった点**
4. **組込み開発でC90が使われ続ける理由**

回答は exercises/standards\_comparison.md に記述してください。

#### 問題5: コンパイル手順の詳細調査

以下のコンパイラオプションを使用して、コンパイル過程を段階的に確認してください。

1. **プリプロセッサ出力の確認**

* gcc -E examples/environment\_check.c > preprocessed.i

1. **アセンブリコード生成**

* gcc -S examples/environment\_check.c

1. **オブジェクトファイル生成**

* gcc -c examples/environment\_check.c

1. **実行ファイル生成**

* gcc environment\_check.o -o environment\_check

各段階で生成されるファイルの内容を確認し、exercises/compilation\_report.md にまとめてください。

#### 問題6: 環境固有情報の調査

以下の情報を調べて、solutions/ex1\_6\_system\_info.c プログラムを作成してください。

**表示する情報:** - 使用しているOS - コンパイラの種類とバージョン - CPUアーキテクチャ（32bit/64bit） - エンディアン（ビッグエンディアン/リトルエンディアン） - 各データ型のサイズ

## 提出形式

各問題の回答は以下のファイルに記述してください。

* **問題3**: solutions/ex1\_3\_hello\_name.c
* **問題4**: exercises/standards\_comparison.md
* **問題5**: exercises/compilation\_report.md
* **問題6**: solutions/ex1\_6\_system\_info.c

## 評価基準

### 基礎問題

* 環境確認プログラムが正常にコンパイル・実行できる
* 各規格でのコンパイルができる
* ex1\_3\_hello\_name.cが要求仕様を満たしている
* C90準拠で記述されている

### 応用問題

* 規格比較が正確に記述されている
* コンパイル手順が理解できている
* 環境固有情報の取得ができている
* レポートが詳細で分かりやすい

## 参考資料

* [GCC公式ドキュメント](https://gcc.gnu.org/documentation.html)
* [C言語規格書](https://www.iso.org/standard/74528.html)
* [GNU Make マニュアル](https://www.gnu.org/software/make/manual/)

## 次の段階

演習課題が完了したら、[基本文法・Hello World](../../basics-syntax/README.md) に進んでください。

# 第2章: 基本文法・Hello World - 演習問題

## 基礎課題

### 課題2-1: 基本的な出力

プログラムを作成して以下を出力してください。

こんにちは、C言語の世界へ！  
私の名前は [あなたの名前] です。  
今日からプログラミングを始めます。

**要求事項:** - printf関数を使用すること - 適切なエスケープシーケンスを使用すること - [あなたの名前]の部分は実際の名前に置き換えること

**ファイル名:** ex2\_1\_hello\_intro.c

### 課題2-2: 変数と基本データ型

以下の変数を定義し、値を代入して出力するプログラムを作成してください。

**変数一覧:** - 整数型: 年齢（例：25） - 浮動小数点型: 身長（例：170.5） - 文字型: 血液型（例：‘A’） - 整数型: 好きな数字（例：7）

**出力例:**

=== 自己紹介データ ===  
年齢: 25歳  
身長: 170.5cm  
血液型: A型  
好きな数字: 7  
================

**要求事項:** - 適切なデータ型を選択すること - printfの書式指定子を正しく使用すること - コメントで各変数の説明を記述すること

**ファイル名:** ex2\_2\_personal\_data.c

### 課題2-3: 入力と出力

ユーザーから名前と年齢を入力してもらい、それを使って計算結果を表示するプログラムを作成してください。

**機能:** 1. 名前を入力してもらう 2. 年齢を入力してもらう 3. 10年後の年齢を計算する 4. 結果を表示する

**出力例:**

お名前を入力してください: 田中太郎  
年齢を入力してください: 25  
  
こんにちは、田中太郎さん！  
現在の年齢: 25歳  
10年後の年齢: 35歳  
プログラミングの学習、頑張ってください！

**要求事項:** - scanf関数を使用すること - 適切なバッファサイズを設定すること - 計算結果を変数に保存すること

**ファイル名:** ex2\_3\_age\_calculator.c

## 応用課題

### 課題2-4: 書式指定子の練習

様々な書式指定子を使って、数値を異なる形式で表示するプログラムを作成してください。

**表示する数値:** 123, 3.14159, 255

**出力例:**

=== 書式指定子のデモ ===  
整数 123 の表示:  
 10進数: 123  
 16進数: 7b  
 8進数: 173  
 フィールド幅5: | 123|  
 ゼロ埋め: |00123|  
  
実数 3.14159 の表示:  
 デフォルト: 3.14159  
 小数点以下2桁: 3.14  
 指数表記: 3.14159e+00  
 フィールド幅10.2: | 3.14|  
  
文字コード 255:  
 文字として: ÿ  
 16進数: ff  
 10進数: 255  
====================

**要求事項:** - 各種書式指定子を使用すること（%d, %x, %o, %f, %e, %c など） - フィールド幅とゼロ埋めを実演すること - 適切なコメントを記述すること

**ファイル名:** ex2\_4\_format\_demo.c

### 課題2-5: 簡単な計算機

四則演算を実行する簡単な計算機プログラムを作成してください。

**機能:** 1. 2つの数値を入力してもらう 2. 四則演算（+, -, \*, /）の結果を表示する 3. 割り算では整数除算と実数除算の両方を表示する

**出力例:**

簡単な計算機プログラム  
===================  
第1の数値を入力してください: 7  
第2の数値を入力してください: 2  
  
計算結果:  
 7 + 2 = 9  
 7 - 2 = 5  
 7 \* 2 = 14  
 7 / 2 = 3 (整数除算)  
 7 / 2 = 3.50 (実数除算)

**要求事項:** - 整数除算と実数除算を区別すること - ゼロ除算のチェックは不要（基礎課題のため） - 適切な変数名を使用すること

**ファイル名:** ex2\_5\_simple\_calculator.c

## 挑戦課題

### 課題2-6: 文字とASCIIコード

文字とASCIIコードの関係を学ぶプログラムを作成してください。

**機能:** 1. ユーザーから文字を入力してもらう 2. その文字のASCIIコードを表示する 3. ASCIIコード表の一部を表示する

**出力例:**

文字を1つ入力してください: A  
入力された文字: A  
ASCIIコード: 65  
  
ASCII表（32-126）の一部:  
 32: 33: ! 34: " 35: # 36: $ 37: % 38: & 39: '  
 40: ( 41: ) 42: \* 43: + 44: , 45: - 46: . 47: /  
 48: 0 49: 1 50: 2 51: 3 52: 4 53: 5 54: 6 55: 7  
 56: 8 57: 9 58: : 59: ; 60: < 61: = 62: > 63: ?  
 64: @ 65: A 66: B 67: C 68: D 69: E 70: F 71: G

**要求事項:** - getchar()またはscanf(" %c", &ch)を使用すること - ループを使ってASCII表を表示すること - 適切な書式で表を整列させること

**ファイル名:** ex2\_6\_ascii\_explorer.c

## 提出について

### ファイル構成

solutions/  
├── ex2\_1\_hello\_intro.c  
├── ex2\_2\_personal\_data.c  
├── ex2\_3\_age\_calculator.c  
├── ex2\_4\_format\_demo.c  
├── ex2\_5\_simple\_calculator.c  
└── ex2\_6\_ascii\_explorer.c

### コンパイルと実行

# コンパイル例  
gcc -o hello\_intro ex2\_1\_hello\_intro.c  
gcc -o personal\_data ex2\_2\_personal\_data.c  
gcc -o age\_calculator ex2\_3\_age\_calculator.c  
  
# 実行例  
./hello\_intro  
./personal\_data  
./age\_calculator

### 評価ポイント

1. **基本文法**: 正しいC言語の文法で記述されているか
2. **コーディング規約**: 適切なインデント、変数名、コメントが使用されているか
3. **機能性**: 要求された機能が正しく実装されているか
4. **入出力**: printf/scanfが適切に使用されているか
5. **書式指定**: 適切な書式指定子が使用されているか

### 学習のポイント

* C言語の基本的な入出力方法
* 変数の宣言と初期化
* 基本データ型の理解
* 書式指定子の使い方
* コメントの書き方
* プログラムの基本構造

頑張って取り組んでください！

# 第3章: データ型と変数 - 演習問題

## 基礎問題

### 演習3-1. 変数宣言と初期化

さまざまなデータ型の変数を宣言し、初期化して値を表示するプログラムを作成してください。

**要件:** - 各基本データ型（char, short, int, long, float, double）の変数を宣言 - 適切な値で初期化 - printf関数で型に応じた書式指定子を使用して出力 - 符号付き・符号なしの両方を含める

**期待される出力例:**

char型: 文字 = 'A', 値 = 65  
unsigned char型: 値 = 255  
short型: 値 = -1000  
unsigned short型: 値 = 65535  
int型: 値 = -123456  
unsigned int型: 値 = 4294967295  
float型: 値 = 3.14159  
double型: 値 = 3.141592653589793

### 演習3-2. 四則演算計算機

2つの数値を変数に格納し、四則演算（加算、減算、乗算、除算）の結果を表示するプログラムを作成してください。

**要件:** - 整数型と浮動小数点型の両方で実装 - 除算では整数除算と実数除算の違いを表示 - 各演算結果を見やすく表示

### 演習3-3. データ型サイズの確認

sizeof演算子を使って、各データ型のサイズを表示するプログラムを作成してください。

**要件:** - 基本データ型すべてのサイズを表示 - 配列のサイズも確認 - ポインタのサイズも確認 - サイズをバイト単位で表示

## 応用問題

### 演習3-4. 型変換の理解

整数除算と実数除算の違い、および暗黙的・明示的型変換を確認するプログラムを作成してください。

**要件:** - 同じ数値で整数除算と実数除算を実行し、結果を比較 - 暗黙的型変換が発生する例を実装 - 明示的型変換（キャスト）を使用した例を実装 - 精度の損失が発生する例を示す

### 演習3-5. スコープの実験

グローバル変数、ローカル変数、静的変数を使い分けるプログラムを作成してください。

**要件:** - 同じ名前の変数を異なるスコープで宣言 - スコープの隠蔽（シャドウイング）を実演 - 静的変数を使ったカウンター関数を実装 - 各変数の値の変化を追跡して表示

### 演習3-6. 定数の活用

constと#defineを使って定数を定義し、円の面積と円周を計算するプログラムを作成してください。

**要件:** - 円周率をconstと#defineの両方で定義 - 半径を入力として受け取る - 面積と円周を計算して表示 - 定数を変更しようとした場合のエラーを確認（コメントで説明）

## チャレンジ問題

### 温度変換プログラム

摂氏・華氏・ケルビンの温度単位を相互変換するプログラムを作成してください。

**要件:** - 変換式で使用する定数を適切に定義 - 浮動小数点演算の精度に注意 - ユーザーフレンドリーな入出力 - 変換公式。 - 華氏 = 摂氏 × 9/5 + 32 - ケルビン = 摂氏 + 273.15

## 提出方法

1. 各問題に対して別々のCファイルを作成
2. ファイル名は ex3\_1\_variables.c, ex3\_2\_calculator.c のように命名
3. 各プログラムの冒頭にコメントで問題番号と簡単な説明を記載
4. C90規格でコンパイルできることを確認

## コンパイル例

gcc -std=c90 -Wall -Wextra -pedantic ex\_variables.c -o ex\_variables  
./ex\_variables

## ヒント

* printf関数の書式指定子を正しく使用する
  + %d: int
  + %u: unsigned int
  + %ld: long
  + %f: float/double
  + %c: char
* 初期化されていない変数の使用に注意
* 型の範囲を超えないよう注意
* sizeof演算子の戻り値はsize\_t型（%luで出力）

# 第4章: 演算子 - 演習問題

## 基礎問題

### 演習4-1: 四則演算計算機

2つの整数を入力として受け取り、すべての算術演算（加算、減算、乗算、除算、剰余）の結果を表示するプログラムを作成してください。

**要件:** - ユーザーから2つの整数を入力 - 各演算の結果を見やすく表示 - ゼロ除算のチェックを含める

**期待される出力例:**

2つの整数を入力してください: 10 3  
10 + 3 = 13  
10 - 3 = 7  
10 \* 3 = 30  
10 / 3 = 3  
10 % 3 = 1

### 演習4-2: 比較と論理演算

2つの整数を入力として受け取り、関係演算子と論理演算子を使って最大値と最小値を求めるプログラムを作成してください。

**要件:** - 2つの整数を入力 - if文を使わず、条件演算子（三項演算子）のみで実装 - 最大値と最小値を表示

### 演習4-3: インクリメント・デクリメント

前置と後置のインクリメント・デクリメント演算子の違いを確認するプログラムを作成してください。

**要件:** - 変数に対して前置・後置の両方を使用 - 各操作後の値を表示 - 式の中での動作も確認

## 応用問題

### 演習4-4: ビット操作

整数を入力として受け取り、その数値の各ビットを表示し、特定のビット操作を実行するプログラムを作成してください。

**要件:** - 2進数表示 - 指定したビット位置のON/OFF確認 - 特定ビットの設定・クリア・トグル機能

### 演習4-5: 条件演算子の活用

3つの数値を入力として受け取り、条件演算子（三項演算子）のみを使って昇順に並び替えるプログラムを作成してください。

**要件:** - if文、switch文を使用しない - 条件演算子のネストを活用 - 並び替えた結果を表示

### 演習4-6: 演算子優先順位

複雑な式を含むプログラムを作成し、演算子の優先順位による計算結果の違いを確認してください。

**要件:** - 同じ数値で異なる式を評価 - 括弧の有無による結果の違いを表示 - 少なくとも3種類以上の演算子を使用

## 提出方法

1. 各問題に対して別々のCファイルを作成
2. ファイル名は ex4\_1\_calculator.c, ex4\_2\_comparison.c のように命名
3. 各プログラムの冒頭にコメントで問題番号と簡単な説明を記載
4. C90規格でコンパイルできることを確認

## コンパイル例

gcc -std=c90 -Wall -Wextra -pedantic ex4\_1\_calculator.c -o ex4\_1\_calculator  
./ex4\_1\_calculator

## ヒント

* 演算子の優先順位に注意
* ゼロ除算やオーバーフローに対する考慮
* 可読性を意識したコードを心がける
* 適切なコメントを追加する

# 第5章: 制御構造（条件分岐） - 演習問題

## 演習の目的

* if文、switch文の理解を深める
* 条件演算子の適切な使い方を習得する
* 複雑な条件分岐の設計力を養う

## 演習問題

### 演習4-1: 年齢による料金計算

映画館の料金システムを実装してください。

**要件:** - 3歳未満: 無料 - 3歳以上12歳以下: 子供料金（800円） - 13歳以上18歳以下: 学生料金（1200円） - 19歳以上65歳以下: 大人料金（1800円） - 66歳以上: シニア料金（1200円）

**実装のヒント:** - if-else if文を使用 - 境界値に注意

### 演習4-2: 電卓プログラム

四則演算ができる簡単な電卓を作成してください。

**要件:** - ユーザーから2つの数値と演算子（+, -, \*, /）を入力 - switch文を使って演算を実行 - ゼロ除算のエラー処理を含める

**実装のヒント:** - 演算子は char 型で受け取る - 除算の場合は分母が0でないかチェック

### 演習4-3: BMI判定プログラム

身長と体重からBMIを計算し、判定するプログラムを作成してください。

**要件:** - BMI = 体重(kg) / (身長(m) × 身長(m)) - BMI判定基準: - 18.5未満: 低体重 - 18.5以上25未満: 標準体重 - 25以上30未満: 肥満度1 - 30以上: 肥満度2

**実装のヒント:** - 身長はセンチメートルで入力してメートルに変換 - 浮動小数点数を使用

### 演習4-4: うるう年判定

西暦年を入力してうるう年かどうか判定するプログラムを作成してください。

**うるう年の条件:** 1. 4で割り切れる年はうるう年 2. ただし、100で割り切れる年は平年 3. ただし、400で割り切れる年はうるう年

**実装のヒント:** - 複数の条件を論理演算子で組み合わせる - 条件の順序に注意

### 演習4-5: 成績評価システム

複数の科目の点数から総合評価を出すプログラムを作成してください。

**要件:** - 3科目（国語、数学、英語）の点数を入力 - 平均点を計算 - 平均点による評価（A〜F） - 全科目60点以上の場合のみ「合格」、それ以外は「不合格」 - 条件演算子を使って簡潔に表示

**実装のヒント:** - 平均点の計算は整数演算に注意 - 複数の条件を組み合わせる

## チャレンジ問題

### チャレンジ4-6: じゃんけんゲーム

コンピュータとじゃんけんをするプログラムを作成してください。

**要件:** - ユーザーの手を数値で入力（1:グー、2:チョキ、3:パー） - コンピュータの手はランダムに決定 - 勝敗を判定して表示 - 不正な入力のチェック

**追加要件:** - 3回勝負で最終的な勝者を決定 - 各1回の結果を記録して最後に表示

## 提出方法

1. 各演習問題ごとに独立したCファイルを作成
2. ファイル名は ex4\_1.c, ex4\_2.c のような形式で
3. コメントで問題番号と簡単な説明を記載
4. コンパイル・実行確認を必ず実行う

## 期限

演習問題は次の章に進む前に完了させましょう。

# 第6章: 制御構造（ループ） - 演習問題

## 演習の目的

* for文、while文、do-while文の使い分けを理解する
* break文とcontinue文を適切に使えるようになる
* ネストしたループを活用した問題解決能力を養う

## 演習問題

### 演習5-1: フィボナッチ数列

最初のN個のフィボナッチ数を表示するプログラムを作成してください。

**要件:** - ユーザーから表示する個数Nを入力 - フィボナッチ数列: 0, 1, 1, 2, 3, 5, 8, 13, … - for文を使用して実装

**実装のヒント:** - 最初の2つの数は0と1 - 3番目以降は前の2つの数の和

### 演習5-2: 数当てゲーム（改良版）

1から100までの数当てゲームを作成してください。

**要件:** - プログラムがランダムに1～100の数を決定 - ユーザーが予想を入力 - 「もっと大きい」「もっと小さい」のヒントを出す - 10回以内に当てられなければゲームオーバー - do-while文を使用

**実装のヒント:** - rand() % 100 + 1 でランダムな数を生成 - 試行回数をカウント

### 演習5-3: 素数リスト

指定された範囲内のすべての素数を表示するプログラムを作成してください。

**要件:** - ユーザーから範囲（開始値と終了値）を入力 - その範囲内の素数をすべて表示 - 素数の個数も表示 - ネストしたループとbreak文を活用

**実装のヒント:** - 2からsqrt(n)まで割り切れるか確認 - 効率化のため、偶数は2以外スキップ

### 演習5-4: 図形の描画

アスタリスク(\*)を使って様々な図形を描画するプログラムを作成してください。

**要件:** 以下の4つの図形を描画する機能を実装: 1. 正方形 2. 直角三角形 3. 逆直角三角形 4. ダイヤモンド

**例（サイズ5の場合）:**

正方形:

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

直角三角形:

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

逆直角三角形:

\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*

ダイヤモンド:

\*  
 \*\*\*  
\*\*\*\*\*  
 \*\*\*  
 \*

**実装のヒント:** - ネストしたループを使用 - 空白文字とアスタリスクの配置に注意

### 演習5-5: 掛け算表

9×9の掛け算表を表示するプログラムを作成してください。

**要件:** - ネストしたfor文を使用 - 見やすい表形式で出力 - 列と行のヘッダーを含める

**実装のヒント:** - printf()の書式指定子で桁揃えを行う - %3d などを使用して桁数を統一

### 演習5-6: 階乗計算

ユーザーから入力された数の階乗を計算するプログラムを作成してください。

**要件:** - ユーザーから正の整数を入力 - その数の階乗を計算して表示 - while文を使用して実装 - オーバーフローに注意

**実装のヒント:** - n! = n × (n-1) × (n-2) × … × 1 - long long型の使用を検討

## 提出方法

1. 各演習問題ごとに独立したCファイルを作成
2. ファイル名は ex5\_1.c, ex5\_2.c のような形式で
3. コメントで問題番号と簡単な説明を記載
4. コンパイル・実行確認を必ず実行

## 期限

演習問題は次の章に進む前に完了させましょう。

# 演習問題を解き終えたら

## チェックリスト

基礎編の演習問題をすべて解き終えたら、以下の点を確認してください：

* すべての問題でコンパイルエラーなく実行できた
* 期待される出力が得られた
* コードの意味を説明できる
* 別の方法での実装を考えられる

## 次のステップ

基礎編の演習を完了したら、応用編の学習に進んでください。応用編では、より複雑なデータ構造とアルゴリズムを扱います。

## 困ったときは

* メインテキストの該当章を再度確認
* サンプルコードと比較
* エラーメッセージを注意深く読む
* 段階的にデバッグ出力を追加

プログラミングは試行錯誤の連続です。エラーを恐れず、積極的に挑戦してください。