

情報演習 32

ステップ30



# C言語 [基礎編]

## アンサー・ブック



# まえがき 解答集をご利用の方へ

この度は本書をご利用いただきましてありがとうございました。このまえがきには、本書を用いて学習しておられる方へ、中でも本書で学生のご指導にあたっておられる先生方にお伝えしたいことをまとめました。

## 本書の方針

本書は、全く初めてプログラミングを学習し始める人のワークブックとして作成しました。今後、コンピュータやプログラミングを専門としていかないとしても、教養として、あるいはもう一步進んで補助的な仕事の道具としてプログラミングのスキルを身に付けたいという方向けです。本書の学習内容だけで、職業プログラマとしてのプログラミングスキルが十分に身につくというものではありません。今後プログラミングのスキルを高めたいならば、最初に最低限身につけておくべき知識と技能のワークブックです。小学生時代に使った計算ドリルのように、不得手なところがあれば繰り返し取り組みましょう。そのように、授業の中で学生さんにお話しておくことをおすすめします。

## フローチャート用紙、コーディング用紙を増し刷りしておきましょう

本書には綴じ込みでフローチャート用紙とコーディング用紙を収めています。これを多数印刷して、暗記するまで繰り返し書く（描く）訓練をしましょう。本書をご利用ということは、少なくとも専門的知識・技術の学習のはずです。楽しんで、体験して「プログラミングって楽しいな」という印象に残れば良い程度の学習ではないはずです。それならば、手を動かし、多少の苦労を味わうことが必要です。これらの用紙が学生のプリント綴じ込み用フラットファイルを一年間でいっぱいに分厚くするぐらいの作業をしてもらいましょう。

## フローチャートについて

フローチャートの学習にこだわったのも、それが理由です。全国工業高校校長協会主催の情報技術検定3級に出題されるから、という以上に、プログラミングを始めて学ぶ方には、手順を論理

的に図式化して考える訓練が必要だと考えています。その訓練に、フローチャートの読み書きが大変有用なのです。実際にプログラミングを職業とする場合に、フローチャートで考えることは滅多にありません。そのレベルで用いるのは、もっと抽象化した図式でマクロに考えるか、コードそのものでマイクロに考えていきます。

## 関数の書き方について

本書に掲載したソースコードのコーディング・スタイルについては、賛否があることと思います。特に関数の記述方法については、オールドスタイルです。より現代的で実用的な関数を記述しようとするれば、

- 引数の型はシグネチャの内部で定義する
- メイン関数をファイルの先頭に置き、ユーザ関数はプロトタイプ宣言する。

などのスタイルをとるべきです。このような現代的なスタイルを取らなかった理由は、学習する要素を最小限にとどめたかったからです。より実用的なソースコードを記述する習慣をつける前に、最低限の力として、動くプログラムを作る力をつけることを優先しました。ただし、全く実用的でないかという点、以下の理由で有効なスタイルでもあります。まず、引数の型の記述方法については、本書のスタイルを取ることでシグネチャを短く記述できます。後述する変数の命名ルールによって、複数の引数を持つ関数のシグネチャが長くなるからです。加えて、それぞれの引数の型宣言の行で、一行コメントを用いて説明コメントを入れることも可能になります。コーディング・スタイルは、できるだけそのプログラミング言語の標準的なものを採用すべきですが、実際にプログラミングを行う際には所属する職場、参加するプロジェクトごとに定められたスタイルに従うものです。学生さんには、本書のスタイルが唯一絶対ではないこと、また、よりベターなものであるとも言えないことなどお伝え下さい。本書のコーディング・スタイルは、あくまで著者の考える入門者向けです。

## 変数や関数の命名について

変数や関数の命名については、やや現代的なスタイルをとっています。現代的な中でも、Java 言語に近いスタイルを選択しました。オールドスタイルな C 言語の命名方法は、可能な限り省略形を用いるものです。個人か、少人数で一つのプロジェクトを進める際には短い名称である方が効率が良いからです。また、長い名称はそれだけでタイピングの手間を取りますし、ソースコードの一行が長くなってしまい、VT100 スタイルの端末エミュレータでは一覧性が落ちます。L3 スイッチや組み込み機器のメンテナンスをする方ならいざ知らず、プログラミングを学習し始めたばかりの方に、この窮屈な環境に対応するためのルールは不要だと考えています。大きくて快適な高解像度のディスプレイでプログラミングを学習する学生さんには、少々名前が長くなろうと

も、その意味するところをもれなく記述する命名スタイルを使用してもらう方が良いと考えます。指導に当たる先生方にあらましては、今後のカリキュラムに照らし合わせて命名法のご指導を検討いただきたいと思います。アルゴリズムの学習と同等か、それ以上に、この命名のスキルは大切だからです。

## 再帰的関数について

関数を用いてアルゴリズムを劇的に簡略化できる例として、再帰的関数（本文中では再帰呼び出し）を取り上げました。情報技術検定3級では、ユーザ関数の作成も再帰的関数も現在は取り上げられていないので、本書の主目的からは外れたトピックです。しかしながら、ひとクラス40人が学習していれば、中には必ず数人は、学習内容に物足りなくなる学生がいるものです。そのような学生向けに、ユーザ関数の作成や再帰的関数の学習は用意しておくべきです。私は、ユーザ関数の作成ができるようになって初めて、実用的なソフトウェアの作成ができるレベルであると考えています。長いソースコードを働きごとに分割する、すなわち処理の抽象化という活動です。これを身につければ、今後C言語の学習を確実に深めていくことができるでしょう。他のどの言語を学習し使おうとしても、どうにもならない、わからないということにはならないでしょう。

## チェックシートでモチベーション維持

本書には学習状況のチェックシートを用意しています。先生方には、これをご活用いただいて、学生が宿題、レポートとして提出したものを確認し、合格するまで繰り返し再提出してもらいましょう。そして合格しましたら、該当の欄に押印してください。先生方からの赤い印がチェックシートに溜まっていけば、学生のモチベーション維持に役立つはずです。私が機械製図の授業の中で、仕上がった図面に「合格」などの印を押すと、無精髭の学生が意図せずニヤリとする姿をよく見ます。そんな子供じゃあるまいし、と思われず、お試しください。誰しもこれは嬉しいことなのでしょう。

## トレースをご指導ください

本書では紙面の都合上取り扱えなかったトピックにトレースがあります。トレースとは、プログラムの動きを、学習者が紙の上、あるいは頭の中で追いかける作業です。変数の一覧を紙の上に書き、プログラムの実行に応じて値を更新していく作業です。単純にソース・プログラムを書き写すだけでは修得できなかった力が得られます。情報技術検定ではトレースの問題が必ず出題されています。先生方におかれましては、指導時間を確保できますなら、ソース・プログラムの模写と並行してトレースの結果をレポートとして学生に要求してください。トレースの方法については解答集のP.72に掲載しています。



# 目次

|                |    |
|----------------|----|
| 解答             | 1  |
| ステップ 1 ワークの解答  | 1  |
| ステップ 2 ワークの解答  | 2  |
| ステップ 3 ワークの解答  | 3  |
| ステップ 4 ワークの解答  | 4  |
| ステップ 5 ワークの解答  | 5  |
| ステップ 6 ワークの解答  | 7  |
| ステップ 7 ワークの解答  | 9  |
| ステップ 8 ワークの解答  | 11 |
| ステップ 9 ワークの解答  | 12 |
| ステップ 10 ワークの解答 | 14 |
| ステップ 11 ワークの解答 | 16 |
| ステップ 12 ワークの解答 | 18 |
| ステップ 13 ワークの解答 | 21 |
| ステップ 14 ワークの解答 | 24 |
| ステップ 15 ワークの解答 | 26 |
| ステップ 16 ワークの解答 | 27 |
| ステップ 17 ワークの解答 | 29 |
| ステップ 18 ワークの解答 | 33 |
| ステップ 19 ワークの解答 | 36 |
| ステップ 20 ワークの解答 | 38 |
| ステップ 21 ワークの解答 | 40 |
| ステップ 22 ワークの解答 | 42 |
| ステップ 23 ワークの解答 | 44 |
| ステップ 24 ワークの解答 | 47 |
| ステップ 25 ワークの解答 | 49 |

---

|                          |    |
|--------------------------|----|
| ステップ 26 ワークの解答 . . . . . | 51 |
| ステップ 27 ワークの解答 . . . . . | 54 |
| ステップ 28 ワークの解答 . . . . . | 57 |
| ステップ 29 ワークの解答 . . . . . | 59 |
| ステップ 30 ワークの解答 . . . . . | 61 |
| トレース . . . . .           | 72 |

# 解答

## ステップ1 ワークの解答

### level 1

ソフトウェアとは

1. あらかじめ指示した手順に従い計算を行うことができる電子回路のこと。
2. コンピュータそのもの、あるいはコンピュータが組み込まれた機械、またはその周辺装置のことをハードウェアとよぶ。
3. ハードウェアを動作させる手順。
4. プログラムそのものやプログラムの使い方、プログラムの管理の仕方、ハードウェアまで含めたシステム全体の取り扱い方法や約束事のこと。

プログラミングとは

1. (1) 機械語またはマシン語、(2) 電子回路、(3) プログラミング、(4) 低水準
2. コンピュータとユーザ、コンピュータとソフトウェアの間に入ってシステム全体を有効に動かすためのプログラムのこと。
3. UNIX,Windows,MacOS,Android,iOS,TRON の中から3つ
4. 人間がコンピュータに仕事をさせる手順を考えるために都合の良い仕組みを持つプログラミング言語のこと。

C 言語とは

1. (1) 高水準、(2) プログラム、(3) 機械語、(4) デニス・リッチー、(5)UNIX、(6) ハードウェア、(7) システム・プログラミング言語

## ステップ2 ワークの解答

### level 1

1. (1) ソース・プログラム、(2) Hello world
2. (1) フリー・フォーマット、(2) 字下げ、またはインデント
3. (1) コメント

### level 2

記号やアルファベットを忠実に書写してあればよしとします。字下げ（インデント）の数が、同じインデント位置の行同士でそろえます。唯一絶対の決まりはありません。本書では字下げ1段の数をスペース4つ分にしています。

### level 3

コメントを英語で書くのは敷居が高いと感じるかもしれません。文法的に間違っているかどうかを気にせず、拙くても勇気を出して書きましょう。分からなければ和英辞典をひいて、なるべく先に書いてある単語を選びましょう。

#### 1. 著作権者の宣言

##### リスト 1 著作権者の宣言コメント

```
1 /* This code is written by John Doe. */
```

##### リスト 2 著作権者の宣言コメントその2

```
1 /* Author : John Doe. */
```

#### 2. 著作権者、目的、略称と正式名称

##### リスト 3 著作権者、目的、略称と正式名称のコメント

```
1 /*  
2  * Author : John Doe  
3  * Purpose : Showing today's date  
4  * Acronym : sd  
5  * Acronym's Original words : Show Date  
6  */
```

## ステップ 3 ワークの解答

### level 1

1. (1) 予約語,(2) キーワード
2. (1) 識別子,(2) 英字,(3)\_またはアンダーバー,(4) 数字
3. (1) ナンバー,(2) ブレース,(3) バックスラッシュ
4. (1) ( , (2); , (3),

### level 2

1. `main` を丸で囲んでください。
2. (a),(b),(d),(e) が正解です。(c) は数字から始まっているので適切ではありません。
3. 下の予約語一覧に含まれる単語が全てあれば結構です。

|                       |                       |                       |                     |
|-----------------------|-----------------------|-----------------------|---------------------|
| <code>auto</code>     | <code>break</code>    | <code>case</code>     | <code>char</code>   |
| <code>const</code>    | <code>continue</code> | <code>default</code>  | <code>do</code>     |
| <code>double</code>   | <code>else</code>     | <code>enum</code>     | <code>extern</code> |
| <code>float</code>    | <code>for</code>      | <code>goto</code>     | <code>if</code>     |
| <code>int</code>      | <code>long</code>     | <code>register</code> | <code>return</code> |
| <code>signed</code>   | <code>sizeof</code>   | <code>static</code>   | <code>short</code>  |
| <code>struct</code>   | <code>switch</code>   | <code>typedef</code>  | <code>union</code>  |
| <code>unsigned</code> | <code>void</code>     | <code>volatile</code> | <code>while</code>  |

## ステップ 4 ワークの解答

### level 1

1. (1) 前処理司令、(2) 関数定義、(3) `main`
2. (1) ソース・プログラム、(2) コンパイラ、(3) コンパイル、(4) バグ、(5) デバッグ、(6) 実行プログラム
3. (1) ソース・プログラム、(2) 実行プログラム

### level 2

書写した後に、下記のように横線を入れてあれば結構です。都合上、横線をコメント行として入れています。

リスト 4 Helloworld

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

## ステップ 5 ワークの解答

### level 1

1. (1) 四則演算、(2) + - \* /、(3) アスタリスク、(4) スラッシュ、(5) ゼロ、(6) ゼロ除算
2. (1) 剰余、(2) %、(3) モジュロ
3. (1) 即値、(2) リテラル

### level 2

ソース・プログラムを間違いなく書きましょう。字下げはそろっているでしょうか。ブレースの数は合っているでしょうか。自分の目で見えて確認する癖をつけましょう。

### level 3

1. 筆者の環境で実行した場合の結果です。不定の場合の数値は環境に依存します。

画面出力

```
27
29
54
1
5
73832 (すなわち不定)
```

2. 特に分数計算ではカッコのつけ方を間違いなく行いましょう。

リスト 5 四則演算の練習その 2

```
1 #include <stdio.h>
2
3 void main()
4 {
5     printf("%f \n", 2 * 3.14 * 5);
6     printf("%f \n", 3.14 * 5 * 5);
7     printf("%f \n", (3.14 * 2.5 * 2.5) / (4 - 5) );
8     printf("%d \n", 1 * 2 * 3 * 4 * 5);
9     printf("%d \n", 5963 % 18);
10 }
```

画面出力

31.400000  
78.500000  
4.906250  
120  
5



## ステップ 6 ワークの解答

### level 1

1. (1) 数値、(2) 2 進数、(3) 整数、(4) 自然数
2. (1) 2 の補数
3. (1)16

### level 2

1. 2 進数、16 進数、10 進数の対応表の完成

表 1 解答 2 進数、16 進数、10 進数の対応表の完成

| 2 進数             | 16 進数     | 10 進数      |
|------------------|-----------|------------|
| 0101             | <b>5</b>  | <b>5</b>   |
| 0010 1010        | <b>2A</b> | <b>40</b>  |
| 0110 1101        | <b>6D</b> | <b>109</b> |
| <b>0110</b>      | 6         | <b>6</b>   |
| <b>0001 0000</b> | 10        | <b>16</b>  |
| <b>0010 0001</b> | 21        | <b>33</b>  |
| <b>1111</b>      | <b>F</b>  | 15         |
| <b>0001 0101</b> | <b>15</b> | 21         |
| <b>0100 0101</b> | <b>45</b> | 69         |

2. 8 ビット長の負の数の計算と対応表の完成

表 2 解答 8 ビット長の負の数の計算と対応表の完成

| 2 進数             | 16 進数     | 10 進数      |
|------------------|-----------|------------|
| <b>11111001</b>  | <b>F9</b> | -7         |
| 1010 1010        | <b>AA</b> | <b>-86</b> |
| <b>1101 0100</b> | D4        | <b>-44</b> |

3. 2 の補数による 2 進数での  $21 - 9$  の計算

$$\begin{aligned} 21 - 9 &= (00010101)B - (00001001)B \\ &\quad (00001001)B \text{ を 2 の補数に変換する。} \\ &\quad \text{反転して (11110110)、1 足して (11110111)B。以上より} \\ &= (00010101)B + (11110111)B \\ &= (100001100)B \\ &\quad \text{8 ビットを超える桁上がりは無視するので} \\ &= (00001100)B \\ &= 12 \end{aligned}$$

## ステップ7ワークの解答

### level 1

1. (1) メモリ、(2) 名前、(3) 機械語、(4) 番地、(5) データ長、(6) 英語
2. (1) `int`、(2) イント、(3) `float`、(4) フロート、(5) 上限

### level 3

1. 変数名をローワーキャメルケースで綴ったソース・プログラム

リスト 6 変数の宣言の練習 解答 ローワーキャメルケースで綴った場合

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int userHeight;
6     userHeight = 170;
7     int userWeight = 70;
8     float BMI = 0;
9     BMI = userWeight / (userHeight * 0.01 * userHeight * 0.01);
10    printf("BMI = %f \n", BMI);
11 }
```

2. 変数名を頭文字をとった略語で綴ったソース・プログラム

リスト 7 変数の宣言の練習 解答 頭文字をとった略語で綴った場合

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int uh;
6     uh = 170;
7     int uw = 70;
8     float BMI = 0;
9     BMI = uw / (uh * 0.01 * uh * 0.01);
10    printf("BMI = %f \n", BMI);
11 }
```

変数の宣言と初期化を一行で済ませることには二つのメリットがあります。

- 2行のプログラムを1行に減らすことができる。
- 変数への値の代入を確実に行うことができる。

実際により重要なのは二つ目のポイント、代入を確実に行うことです。これは「変数の初期化」と呼ばれます。C 言語では変数を宣言した直後、その変数にどんな値が入っているか規定や保証がありません。数値変数を宣言すれば、常識的に 0 が入っていて欲しいのですが、実際にはどんな値が入っているか、実行してみるまでわかりません。このため、プログラムや式の中で予想外の結果が出るのを防ぐために、宣言した直後に必要な値、あるいは望ましい値を代入しておくのです。

10 行目の `printf` にある `%f` は、実数型の変数の値を表示したい時に用います。整数型の変数の値を `%f` を用いて表示しようとしても正しく表示されません。

## ステップ 8 ワークの解答

### level 1

1. (1) 英語、(2)6、(3)31
2. (1) ローワーキャメルケース
3. (1)ind、(2)mnu

### level 3

著者ならこう書いてこう省略するよ、という例としてご覧ください。英語の堪能な方ならばまた別の表現や省略をされるでしょう。

1. widthWindow,wdthWnd
2. diameterCircle,diaCir
3. timeExecution,tmExec
4. numberElement,numElmnt
5. elapsedTimeSeconds,etSeconds
6. weightParts,wgtParts
7. amountOfDeposit,amtOfDep
8. numberOfStudents,numOfStudents
9. mileage,-

## ステップ9 ワークの解答

### level 3

1. 体重 70kg の人が地表面に加える荷重

リスト 8 体重 70kg の人が地表面に加える荷重を計算するソース・プログラム

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     float loadPerson = 0; /* [N] */
6     float massPerson = 70; /* [kg] */
7     float g = 9.8; /* [m/s^2] */
8
9     loadPerson = massPerson * g;
10    printf("Load is %f[N].\n", loadPerson);
11    return 0;
12 }
```

2. 直径 3mm のワイヤの断面積

リスト 9 直径 3mm のワイヤの断面積

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     float A = 0; /* [mm^2] */
6     float D = 3; /* [mm] */
7     float Pi = 3.1415; /* 円周率 */
8
9     A = ( Pi * D * D ) / 4;
10    printf("Cross section is %f[mm^2]\n", A);
11    return 0;
12 }
```

3. 直径 3mm のワイヤに 70kg の人がぶら下がった時の応力計算

リスト 10 直径 3mm のワイヤに 70kg の人がぶら下がった時の応力計算

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     float loadPerson = 0; /* [N] */
6     float massPerson = 70; /* [kg] */
7     float g = 9.8; /* [m/s^2] */
```

```
8      float A          = 0; /* [mm^2] */
9      float D          = 3; /* [mm] */
10     float Pi          = 3.1415; /* [-] */
11     float stress      = 0; /* [MPa] */
12
13     loadPerson = massPerson * g; /* [N] */
14     A = (Pi * D * D) / 4; /* [mm^2] */
15
16     stress = loadPerson / A;
17
18     printf("Stress is %f [MPa]\n", stress);
19     return 0;
20 }
```

物理や工学に親しくない方にとって、応力の計算は初めてだったことでしょう。ワイヤの直径が  $[mm]$  であるのに、応力の単位  $[MPa]$  が  $[N/mm^2]$  であることの整合性どう取られているのか、悩まれることでしょう。ご心配なく。物理や工学を学ぶ学生さんたちが皆悩むところです。じっくり手で計算をしてみればわかりますので、諦めず確認してください。

## ステップ 10 ワークの解答

### level 1

1. (1) 標準出力、(2)printf
2. (1) 拡張表記、(2) エスケープシーケンス
3. (1) 書式、(2)%d、(3)%x

### level 2

それぞれ短いソースプログラムです。今回の課題である printf 文や書式文字列だけでなく、全体的に復習を兼ねて書写しましょう。

### level 3

1. あなたの氏名、年齢を表示するプログラム

リスト 11 問題 氏名年齢を表示する 解答

```

1 #include <stdio.h>
2 int main(void)
3 {
4     printf("氏名: 平田 敦 年齢: 45歳");
5     return 0;
6 }
```

2. クリスマスツリーの出力を得るプログラム

リスト 12 問題 クリスマスツリーを表示する 解答

```

1 #include <stdio.h>
2 int main(void)
3 {
4     printf("      *      \n");
5     printf("    < >      \n");
6     printf("  < c >      \n");
7     printf(" <  c  >      \n");
8     printf(" <   c   >      \n");
9     printf("    |      \n");
10    printf("    |      \n");
11    printf("  -----  \n");
12    printf(" \a\a\a      \n");
13    return 0;
14 }
```



---

---

## ステップ 11 ワークの解答

### level 1

- 1 書式
- 1 まとめて、2 行を変えて、3 初期化
- 1 書式、2 16

### level 3

- 直径 28mm の球の体積

リスト 13 直径 28mm の球の体積

```
1 #include <stdio.h>
2 int main(void)
3 {
4     float r = 10;
5     float pi = 3.14;
6     float v = 4. / 3 * pi * r * r * r;
7     printf("d = %f \n", r * 2);
8     printf("v = %f \n", v);
9     return 0;
10 }
```

画面出力

```
d = 20.000000
v = 4186.666992
```

- 100 時間後が何日後の何時

リスト 14 100 時間後が何日後の何時

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int currentTime = 6;
5     int passageTime = 100;
6     int passageDay = 0;
7     int objectTime = 0;
8
9     passageDay = ( currentTime + passageTime ) / 24;
10    objectTime = ( currentTime + passageTime ) % 24;
11 }
```

```
12     printf("current time = %d, passage time = %d \n", currentTime, passageTime);
13     printf("passage day  = %d \n",passageDay);
14     printf("object time  = %d \n",objectTime);
15     return 0;
16 }
```

画面出力

```
current time = 6, passage time = 100
passage day  = 4
object time  = 10
```

### 3. 172.5cm、60kg の人の BMI

リスト 15 172.5cm、60kg の人の BMI

```
1  #include <stdio.h>
2  int main(void)
3  {
4      float height = 172.5 / 100;
5      float weight = 60.0;
6      float BMI    = weight / (height * height);
7      printf("height = %f, weight = %f \n", height, weight);
8      printf("BMI    = %f \n",BMI);
9      return 0;
10 }
```

画面出力

```
height = 1.725000, weight = 60.000000
BMI    = 20.163832
```

## ステップ 12 ワークの解答

### level 1

1. (1)scanf、(2)&、(3) アンパサンド、(4) アドレス

### level 2

& (アンパサンド) や %, ; と言った記号を書くことになれましたか? これら特殊な記号に慣れ親しむことが C 言語上達の意外な障害のようです。

### level 3

1. 年齢を入力し、出力する

リスト 16 InputAndShowAge.c

```
1 #include <stdio.h>
2 void main()
3 {
4     int age = 0;
5     scanf("%d",&age);
6     printf("Thank you! You are %d years old.\n",age);
7 }
```

画面出力

```
45
Thank you! You are 45 years old.
```

2. 任意の直径 D の球の体積を計算する

リスト 17 InputAndCalcSphereVolume.c

```
1 #include <stdio.h>
2 void main()
3 {
4     float d = 0;
5     float r = d / 2;
6     float pi = 3.14;
7     float v = 4. / 3 * pi * r * r * r;
8     scanf("%f",&d);
9     r = d / 2;
10    v = 4. / 3 * pi * r * r * r;
```

```

11     printf("d = %f \n", d);
12     printf("v = %f \n", v);
13 }

```

画面出力

```

5
d = 5.000000
v = 65.416672

```

### 3. 現在の時刻と経過時間から、何日後の何時かを表示する

リスト 18 現在の時刻と経過時間から、何日後の何時かを表示する

```

1  #include <stdio.h>
2  void main()
3  {
4      int currentTime = 6;
5      int passageTime = 100;
6      int passageDay = 0;
7      int objectTime = 0;
8
9      scanf("%d", &currentTime);
10     scanf("%d", &passageTime);
11     passageDay = ( currentTime + passageTime ) / 24;
12     objectTime = ( currentTime + passageTime ) % 24;
13
14     printf("current time = %d, passage time = %d \n", currentTime, passageTime);
15     printf("passage day = %d \n", passageDay);
16     printf("object time = %d \n", objectTime);
17 }

```

画面出力

```

11
24
current time = 11, passage time = 24
passage day = 1
object time = 11

```

### 4. 身長と体重を入力し、BMI を計算する

リスト 19 InputAndCalcBMI.c

```

1  #include <stdio.h>
2  int main(void)
3  {
4      float height = 0;
5      float weight = 0;
6      scanf("%f", &height);
7      scanf("%f", &weight);

```

```
8     printf("height = %f, weight = %f \n", height, weight);
9     float BMI    = weight / (height * height);
10    printf("BMI    = %f \n",BMI);
11    return 0;
12 }
```

画面出力

```
1.74
79
height = 1.740000, weight = 79.000000
BMI    = 26.093275
```

## ステップ 13 ワークの解答

### level 1

1. (1) printf、(2) ユーザの混乱
2. (1) 区切り文字、(2) 空白文字、スペース

### level 3

1. 年齢を入力し、出力する

リスト 20 GoodInputAndShowAge.c

```
1 #include <stdio.h>
2 void main()
3 {
4     int age = 0;
5     printf("Input your age >");
6     scanf("%d",&age);
7     printf("Thank you! You are %d years old.\n",age);
8 }
```

画面出力

```
Input your age >45
Thank you! You are 45 years old.
```

2. 任意の直径 D の球の体積を計算する

リスト 21 GoodInputAndCalcSphereVolume.c

```
1 #include <stdio.h>
2 void main()
3 {
4     float d = 0;
5     float r = d / 2;
6     float pi = 3.14;
7     float v = 4. / 3 * pi * r * r * r;
8     printf("Input diameter >");
9     scanf("%f",&d);
10    r = d / 2;
11    v = 4. / 3 * pi * r * r * r;
12    printf("d = %f \n", d);
13    printf("v = %f \n", v);
14 }
```

画面出力

```
Input diameter 5
d = 5.000000
v = 65.416672
```

## 3. 現在の時刻と経過時間から、何日後の何時かを表示する

リスト 22 現在の時刻と経過時間から、何日後の何時かを表示する

```
1 #include <stdio.h>
2 void main()
3 {
4     int currentTime = 6;
5     int passageTime = 100;
6     int passageDay = 0;
7     int objectTime = 0;
8
9     printf("Input current time >");
10    scanf("%d", &currentTime);
11    printf("Input passage time >");
12    scanf("%d", &passageTime);
13    passageDay = ( currentTime + passageTime ) / 24;
14    objectTime = ( currentTime + passageTime ) % 24;
15
16    printf("current time = %d, passage time = %d \n", currentTime, passageTime);
17    printf("passage day = %d \n", passageDay);
18    printf("object time = %d \n", objectTime);
19 }
```

画面出力

```
Input current time >11
Input passage time >24
current time = 11, passage time = 24
passage day = 1
object time = 11
```

## 4. 身長と体重を入力し、BMI を計算する

リスト 23 InputAndCalcBMI.c

```
1 #include <stdio.h>
2 int main(void)
3 {
4     float height = 0;
5     float weight = 0;
6     printf("Input your height >");
7     scanf("%f", &height);
8     printf("Input your weight >");
9     scanf("%f", &weight);
10    printf("height = %f, weight = %f \n", height, weight);
```



```
11     float BMI    = weight / (height * height);  
12     printf("BMI    = %f \n",BMI);  
13     return 0;  
14 }
```

画面出力

```
Input your height >1.74  
Input your weight >79  
height = 1.740000, weight = 79.000000  
BMI    = 26.093275
```

## ステップ 14 ワークの解答

### level 1

1. (1) フローチャート、(2) 矢印
2. (1) JIS
3. (1) 順次、(2) 選択、(3) 繰返し
4. (1) アルゴリズム

### level 2

フローチャートの図記号が正しく描かれ、単純に上から下に処理が流れないところには矢印の矢を描き込んでいるか確認しましょう。

### level 3

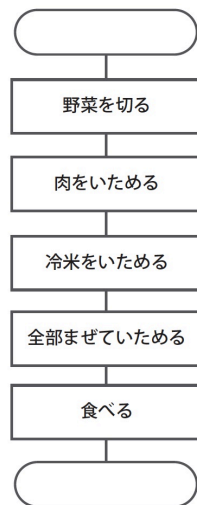


図 1 1. チャーハンの作り方のフローチャート

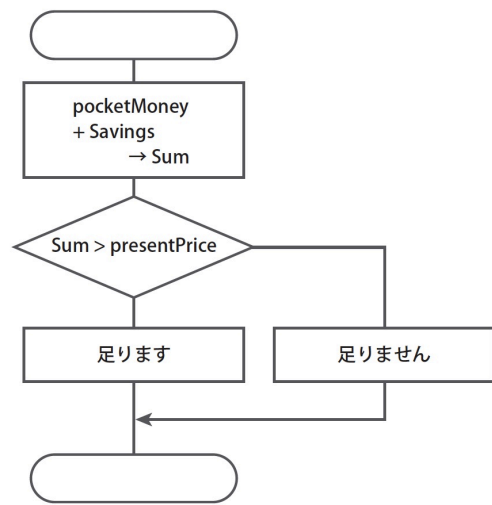


図 2 2. プレゼントを買うためのフローチャート

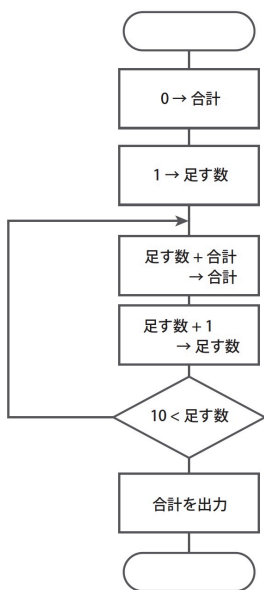


図3 3. 1 から 10 まで足すフローチャート

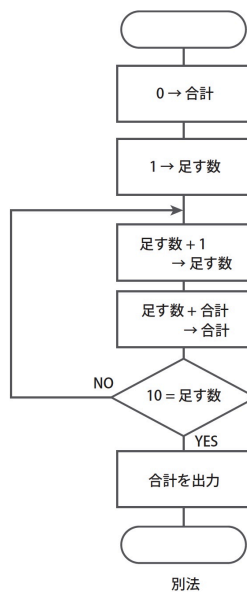


図4 4. 1 から 10 まで足すフローチャート 別法

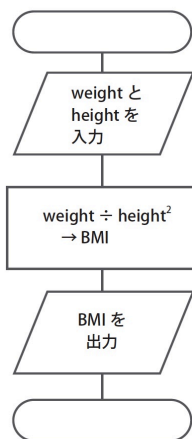


図5 5. BMI を計算するフローチャート

## ステップ 15 ワークの解答

### level 1

1. (1) 単純な繰り返し作業、(2) 創造的
2. (1)1、(2)10、(3)2、(4)1,2,5,7,9、(5)5、(6) 変数「人数」

### level 3



図 6 1. 1 から 10 の数字を出力するプログラムのフローチャートを、ループの図記号を使って描く

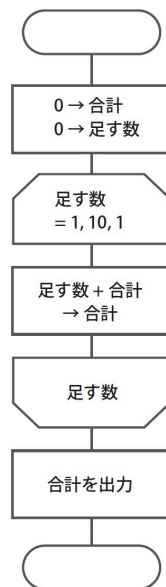


図 7 2. 1 から 10 の合計を求めるフローチャートを、ループの図記号を使って描く

## ステップ 16 ワークの解答

### level 1

1. (1) 分岐
2. (1)if、(2) 条件、(3)else

### level 3

1. 選挙権の有無の判定

リスト 24 選挙権の有無の判定

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int age = 0;
5     printf("年齢を入力してください? [int] > ");
6     scanf("%d",&age);
7     if (age < 18){
8         printf("あなたは選挙権がありません。\\n");
9     } else {
10        printf("あなたは選挙権があります。\\n");
11    }
12    return 0;
13 }
```

2. 選挙権があるか、飲酒喫煙が認められるかの判定

リスト 25 選挙権があるか、飲酒喫煙が認められるかの判定

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int age = 0;
5     printf("年齢を入力してください? [int] > ");
6     scanf("%d",&age);
7     if (age < 18){
8         printf("あなたには選挙権がなく、飲酒喫煙もできません。\\n");
9     } else {
10        if (age < 20){
11            printf("あなたには選挙権があります。飲酒喫煙はできません。\\n");
12        } else {
13            printf("あなたには選挙権があり、飲酒喫煙も許されます。\\n");
14        }
15    }
```

```
16     return 0;
17 }
```

### 3. BMI の計算と肥満度の判定

リスト 26 BMI の計算と肥満度の判定

```
1  #include <stdio.h>
2  int main(void)
3  {
4      float height = 0.0;
5      float weight = 0.0;
6      float BMI    = 0.0;
7
8      printf("身長を入力してください。[ float ,cm]>" );
9      scanf("%f",&height);
10     printf("体重を入力してください。[ float ,kg]>" );
11     scanf("%f",&weight);
12     BMI = weight / (height * 0.01 * height * 0.01);
13     printf("BMI = %f\n",BMI);
14     if (BMI < 18.5) {
15         printf("あなたは低体重です。 \n");
16     } else {
17         if (BMI < 25) {
18             printf("あなたは標準体重です。 \n");
19         } else {
20             printf("あなたは肥満です。 \n");
21         }
22     }
23     return 0;
24 }
```

## ステップ 17 ワークの解答

### level 1

1. 1 真、2 偽
2. 1 論理
3. 1 偽、2 偽
4. 1 論理積
5. 1 論理和
6. 1 排他的論理和

### level 3

1. リスト 17.2 から 17.5 の書き換え  
リスト 17.2 の書き換え

リスト 27 if 文で書き換えた、論理否定のあるプログラム

```
1 #include <stdio.h>
2 int main(void){
3     int a = 10;
4     if (a <= 18) {
5         printf("This is False.\n");
6     } else {
7         printf("This is True.\n");
8     }
9     return 0;
10 }
```

#### リスト 17.3 の書き換え

if 文を使うことで、printf 文を一つ多く書く必要がありました。このような全く同じ仕事を複数の場所に書くプログラムは避けたいところです。

リスト 28 if 文で書き換えた、論理積のあるプログラム

```
1 #include <stdio.h>
2 int main(void){
3     int a = 10;
4     int b = 25;
5     if (a <= 18) {
6         if (b <= 20) {
7             printf("This is True.\n");
8         } else {
```

```
9         printf("This is False.\n");
10     }
11     } else {
12         printf("This is False.\n");
13     }
14     return 0;
15 }
```

#### リスト 17.4 の書き換え

それぞれの条件が一つでも成り立てば”This is True.”と表示したいのですが、どちらも成り立つ場合に二度同じ出力をしたくありません。シンプルなプログラムにするために変数 **result** を用意しました。このような用途の変数をフラグ（旗）と呼びます。

#### リスト 29 if 文で書き換えた、論理和のあるプログラム

```
1 #include <stdio.h>
2 int main(void){
3     int a = 10;
4     int b = 5;
5     int result = 0;
6     if (a <= 18) {
7         result = 1;
8     }
9     if (b <= 20) {
10         result = 1;
11     }
12     if (result == 1) {
13         printf("This is True.\n");
14     } else {
15         printf("This is False.\n");
16     }
17     return 0;
18 }
```

#### リスト 17.5 の書き換え

排他的論理和の仕事をシンプルに書くために、今回もフラグを使いました。変数 **result** の値は条件式が成り立つたびに 1 増加します。どちらか一方だけ成り立つ場合に **result** は 1 ですから、それを最後の if 文で判定する形にしました。

#### リスト 30 if 文で書き換えた、排他的論理和のあるプログラム

```
1 #include <stdio.h>
2 int main(void){
3     int a = 19;
4     int b = 5;
5     int result = 0;
6     if (a <= 18) {
7         result += 1;
8     }
```



```

9      if (b <= 20) {
10         result += 1;
11     }
12     if (result == 1) {
13         printf("This is True.\n");
14     } else {
15         printf("This is False.\n");
16     }
17     return 0;
18 }

```

## 2. 選挙権年齢か、飲酒喫煙が許される年齢かを判定し表示するプログラム

if 文を組み合わせるだけでは複雑なプログラムになりますが、論理演算子を用いるとこんなにシンプルに書けます。

リスト 31 選挙権年齢か、飲酒喫煙が許される年齢かを判定

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int age = 0;
5      printf("年齢を入力してください? [int] > ");
6      scanf("%d",&age);
7      if (age < 18){
8          printf("あなたには選挙権がなく、飲酒喫煙もできません.\n");
9      }
10     if ( (18 <= age) && (age < 20) ) {
11         printf("あなたには選挙権があります。飲酒喫煙はできません.\n");
12     }
13     if (20 <= age) {
14         printf("あなたには選挙権があり、飲酒喫煙も許されます.\n");
15     }
16     return 0;
17 }

```

## 3. 論理演算子を使って書き換えた、肥満度の判定プログラム

なるべく if 文を入れ子にしない方針で書くと、このようになります。else 文を使って場合を切り分けるのも一つの方法です。

リスト 32 論理演算子を使って書き換えた、肥満度の判定プログラム

```

1  #include <stdio.h>
2  int main(void)
3  {
4      float height = 0.0;
5      float weight = 0.0;
6      float BMI    = 0.0;
7
8      printf("身長を入力してください。[float ,cm]>");

```

```

9      scanf("%f",&height);
10     printf("体重を入力してください。[float ,kg]>");
11     scanf("%f",&weight);
12     BMI = weight / (height * 0.01 * height * 0.01);
13     printf("BMI = %f\n",BMI);
14     if (BMI < 18.5) {
15         printf("あなたは低体重です。\\n");
16     }
17     if ((18.5 <= BMI) && (BMI < 25)) {
18         printf("あなたは標準体重です。\\n");
19     }
20     if (25 <= BMI) {
21         printf("あなたは肥満です。\\n");
22     }
23     return 0;
24 }

```

#### 4. 必要なら論理演算子を使って FizzBuzz

「必要なら」というところがミソで、今回こうして論理演算子を使わずに書くことができます。もちろん使っても書けますが、リスト 33 のように使わない方がよりシンプルです。

リスト 33 必要なら論理演算子を使って FizzBuzz

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int num = 0;
5      printf("What number? [int] >");
6      scanf("%d",&num);
7      if ((num % 3) == 0) {
8          printf("Fizz");
9      }
10     if ((num % 5) == 0) {
11         printf("Buzz");
12     }
13     printf("\\n");
14     return 0;
15 }

```

## ステップ 18 ワークの解答

### level 1

1. 1 整数値、2 多い
2. 1 switch、2 case、3 default
3. 1 break

### level 3

英語の月名を表示するプログラムについて、後半部分を省略しています。それぞれの演習においては最後まで作成してください。

1. 入力された数値に対応する月名を表示する

わざと題意と異なる解答例を示します。それは 4 から 6 を入力すると、break 文を書き忘れているために、break 文に行き当たるまで続くコードブロックが実行されることを実感してもらうためです。また、9 月以降の月名は表示するようにしていませんので、辞書を片手に残りのプログラムを補足作成してください。

リスト 34 入力された数値に対応する月名を表示する

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int month = 0;
5     printf("Which month ? [1..12, int] >");
6     scanf("%d",&month);
7     switch( month ) {
8         case 1 :
9             printf("January\n");
10            break;
11        case 2 :
12            printf("February\n");
13            break;
14        case 3 :
15            printf("March\n");
16            break;
17        case 4 :
18            printf("April\n");
19        case 5 :
20            printf("May\n");
21        case 6 :
22            printf("June\n");
```

```
23         case 7 :
24             printf(" July\n");
25             break;
26         case 8 :
27             printf(" August\n");
28             break;
29         default :
30             printf("I don't know!\n");
31             break;
32     }
33     return 0;
34 }
```

2. 経過月数から年数と現在の月、月名を表示する

リスト 35 経過月数から年数と現在の月、月名を表示する

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int months = 0;
5      printf("Months ? [int] >");
6      scanf("%d",&months);
7      int years = months / 12;
8      int month = months % 12;
9      printf("%d month(s) is %d year(s) %d ",months,years,month);
10     switch ( month ) {
11         case 1 :
12             printf(" January");
13             break;
14         case 2 :
15             printf(" February");
16             break;
17         case 3 :
18             printf(" March");
19             break;
20         case 4 :
21             printf(" April");
22             break;
23         case 5 :
24             printf(" May");
25             break;
26         case 6 :
27             printf(" June");
28             break;
29         case 7 :
30             printf(" July");
31             break;
32         case 8 :
33             printf(" August");
34             break;
```

```
35         default :
36             printf("I don't know!");
37             break;
38     }
39     printf("\n");
40     return 0;
41 }
```

## ステップ 19 ワークの解答

### level 1

- 1 真
- 1 継続条件式
- 1 無限ループ
- 1 break

### level 3

- 1 から N までの数字の和を求める

リスト 36 1 から N までの数字の和を求める

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int num = 0;
5     int count = 0;
6     int sum = 0;
7     printf("合計を求めたい数を入力してください [int,0<]");
8     scanf("%d",&num);
9     while(count <= num){
10         sum +=count;
11         ++count;
12     }
13     printf("%d までの合計は %d です。\\n",num,sum);
14     return 0;
15 }
```

- 選挙権年齢か飲酒喫煙が許される年齢かを判定し続ける

リスト 37 選挙権年齢か飲酒喫煙が許される年齢かを判定し続ける

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int age = 0;
5     while (age != -1) {
6         printf("年齢を入力してください? [intで終了,-1] > ");
7         scanf("%d",&age);
8         if (age == -1) {
9             printf("が入力されたので終了します。-1\\n");
10        } else {
```

```

11         if (age < 18){
12             printf("あなたには選挙権がなく、飲酒喫煙もできません。\\n");
13         }
14         if ( (18 <= age) && (age < 20) ) {
15             printf("あなたには選挙権があります。飲酒喫煙はできません。\\n");
16         }
17         if (20 <= age) {
18             printf("あなたには選挙権があり、飲酒喫煙も許されます。\\n");
19         }
20     }
21 }
22 return 0;
23 }

```

### 3. while 文を使って FizzBuzz

7 行目の "%3d" は整数を 3 桁右詰で表示するための書式です。

リスト 38 while 文を使って FizzBuzz

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int num = 0;
5      while(num < 100) {
6          ++num;
7          printf("%3d : ", num);
8          if ((num % 3) == 0) {
9              printf("Fizz");
10         }
11         if ((num % 5) == 0) {
12             printf("Buzz");
13         }
14         printf("\\n");
15     }
16     return 0;
17 }

```

## ステップ 20 ワークの解答

### level 1

1. 1 初期化、2 継続条件、3 再初期化
2. 1 繰り返し回数、2 無限ループ
3. 1 break

### level 3

1. for 文を使って 1 から N までの数字の和を求める

リスト 39 for 文を使って 1 から N までの数字の和を求める

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int num = 0;
5     int count = 0;
6     int sum = 0;
7     printf("合計したい数を入力してください。[int]");
8     scanf("%d",&num);
9     for ( count = 0; count <= num; count++) {
10         sum += count;
11     }
12     printf("1から%dまでの数の合計は%dです。d\n",num,sum);
13     return 0;
14 }
```

2. for 文を使って選挙権年齢か飲酒喫煙が許される年齢かを判定し続ける

リスト 40 for 文を使って選挙権年齢か飲酒喫煙が許される年齢かを判定し続ける

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int age = 0;
5     for ( ; age != -1; ) {
6         printf("年齢を入力してください？ [intで終了,-1] > ");
7         scanf("%d",&age);
8         if (age == -1) {
9             printf("が入力されたので終了します。-1\n");
10        } else {
11            if (age < 18){
12                printf("あなたには選挙権がなく、飲酒喫煙もできません。\\n");
13            }
14        }
15    }
16 }
```



```

14         if ( (18 <= age) && (age < 20) ) {
15             printf("あなたには選挙権があります。飲酒喫煙はできません。\\n");
16         }
17         if (20 <= age) {
18             printf("あなたには選挙権があり、飲酒喫煙も許されます。\\n");
19         }
20     }
21 }
22 return 0;
23 }

```

### 3. for 文を使って FizzBuzz

初期化条件式で変数の宣言から初期化までを一気に行うこの形が、実際の C 言語プログラミングで常套句として用いられています。

リスト 41 for 文を使って FizzBuzz

```

1 #include <stdio.h>
2 int main(void)
3 {
4     for (int num = 1; num <= 100; ++num) {
5         printf("%3d : ", num);
6         if ((num % 3) == 0) {
7             printf("Fizz");
8         }
9         if ((num % 5) == 0) {
10            printf("Buzz");
11        }
12        printf("\\n");
13    }
14    return 0;
15 }

```

### 4. 棒グラフを表示する

リスト 42 棒グラフを表示する

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int length = 0;
5     printf("Length > ");
6     scanf("%d", &length);
7     for (int count = 1; count <= length; ++count) {
8         printf("O");
9     }
10    printf("\\n");
11    return 0;
12 }

```

## ステップ 21 ワークの解答

### level 1

1. 1 A、2 B、3 入れ子
2. 1 15

### level 3

1. 3x3 の正方形に O を並べる

リスト 43 3x3 の正方形に O を並べる

```
1 #include <stdio.h>
2 int main(void)
3 {
4     for ( int col = 0; col < 3; ++col ) {
5         for (int row = 0; row < 3; ++row ) {
6             printf("O");
7         }
8         printf("\n");
9     }
10    return 0;
11 }
```

2. 10x10 のマス目状に、1 から 100 までの数値を表示する

リスト 44 10x10 のマス目状に、1 から 100 までの数値を表示する

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int count = 1;
5     for(int col = 0; col < 10; ++col) {
6         for(int row = 0; row < 10; ++row) {
7             printf("%3d", count);
8             ++count;
9         }
10        printf("\n");
11    }
12    return 0;
13 }
```

3. 入れ子の for 文を使って FizzBuzz

リスト 45 入れ子の for 文を使って FizzBuzz

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int count = 1;
5     for (int col = 0; col < 10; ++col) {
6         for (int row = 0; row < 10; ++row) {
7             if (((count % 3) == 0) && ((count % 5) != 0)) {
8                 printf(" Fizz");
9             }
10            if (((count % 5) == 0) && ((count % 3) != 0)) {
11                printf(" Buzz");
12            }
13            if (((count % 3) == 0) && ((count % 5) == 0)) {
14                printf(" FzBz");
15            }
16            if (((count % 3) != 0) && ((count % 5) != 0)) {
17                printf("%5d", count);
18            }
19            ++count;
20        }
21        printf("\n");
22    }
23    return 0;
24 }
```

## ステップ 22 ワークの解答

### level 1

1. 1 連続、2 データ型
2. 1 配列名、2 要素
3. 1 添え字（そえじ）、2 0、3 n-1
4. 1 1 次元
5. 1 要素数

### level 3

1. 10 人分の成績を入力し、そのまま出力する。

リスト 46 10 人分の成績を入力し、そのまま出力する

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int score[10];
5     for ( int count = 0; count < 10; ++count ){
6         printf("score %d > ",count+1);
7         scanf("%d",&score[count]);
8     }
9     for ( int count = 0; count < 10; ++count ){
10        printf("score[%d] is %d\n",count+1,score[count]);
11    }
12    return 0;
13 }
```

2. 最高点、最低点、平均点を出力

リスト 47 最高点、最低点、平均点を出力

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int score[10];
5     int max = 0;
6     int min = 100;
7     int sum = 0;
8     float ave = 0;
9     for ( int count = 0; count < 10; ++count ){
10        printf("score %d > ",count+1);
11        scanf("%d",&score[count]);
```

```
12         sum += score[count];
13         if ( max < score[count] ) {
14             max = score[count];
15         }
16         if ( score[count] < min ) {
17             min = score[count];
18         }
19     }
20     for ( int count = 0; count < 10; ++count ){
21         printf("score[%d] is %d\n",count+1,score[count]);
22     }
23     printf("max = %d\n",max);
24     printf("min = %d\n",min);
25     ave = sum / 10;
26     printf(" average = %f\n",ave);
27     return 0;
28 }
```

## ステップ 23 ワークの解答

### level 1

1. 1 サーチ
2. 1 ソート

### level 3

1. 10 人分の成績を入力し、高得点から順に並べ替え、出力。最高、最低、平均も出力する。

リスト 48 10 人分の成績を入力し、高得点から順に並べ替え、出力。最高、最低、平均も出力する。

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int score[10];
5      int max = 0;
6      int min = 100;
7      int sum = 0;
8      float ave = 0;
9      for ( int count = 0; count < 10; ++count ){
10         printf("score %d > ", count+1);
11         scanf("%d", &score[count]);
12         sum += score[count];
13         if ( max < score[count] ) {
14             max = score[count];
15         }
16         if ( score[count] < min ) {
17             min = score[count];
18         }
19     }
20     /* sort */
21     for( int i = 0; i < 10-1; ++i ){
22         for ( int j = 10-1; i < j; --j ){
23             if ( score[j] < score[j-1] ){
24                 int temp = score[j];
25                 score[j] = score[j-1];
26                 score[j-1] = temp;
27             }
28         }
29     }
30     /* sort finished */
31     for ( int count = 0; count < 10; ++count ){
32         printf("score[%d] is %d\n", count+1, score[count]);
33     }
```

```

34     printf("max = %d\n",max);
35     printf("min = %d\n",min);
36     ave = sum / 10;
37     printf("average = %f\n",ave);
38     return 0;
39 }

```

2. 先ほどのプログラムに、リニアサーチで50点の位置を見つけるプログラムを加える。

リスト 49 先ほどのプログラムに、リニアサーチで50点の位置を見つけるプログラムを加える。

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int score[10];
5      int max = 0;
6      int min = 100;
7      int sum = 0;
8      float ave = 0;
9      for ( int count = 0; count < 10; ++count ){
10         printf("score %d > ",count+1);
11         scanf("%d",&score[count]);
12         sum += score[count];
13         if ( max < score[count] ) {
14             max = score[count];
15         }
16         if ( score[count] < min ) {
17             min = score[count];
18         }
19     }
20     /* sort */
21     for( int i = 0; i < 10-1; ++i ){
22         for ( int j = 10-1; i < j; --j ){
23             if (score[j] < score[j-1]){
24                 int temp = score[j];
25                 score[j] = score[j-1];
26                 score[j-1] = temp;
27             }
28         }
29     }
30     /* sort finished */
31     for ( int count = 0; count < 10; ++count ){
32         printf("score[%d] is %d\n",count+1,score[count]);
33     }
34     printf("max = %d\n",max);
35     printf("min = %d\n",min);
36     ave = sum / 10;
37     printf("average = %f\n",ave);
38     /* search 50 */
39     int pos = -1;
40     int target = 50;

```

```
41     for(int i = 0; i < 10; ++i){
42         if(score[i] == target){
43             pos = i;
44             break;
45         }
46     }
47     if( (pos != -1) && (score[pos] == target) ){
48         printf("Target(%d) position is score[%d].\n",target ,pos+1);
49     } else {
50         printf("No match!\n");
51     }
52     /* search 50 finished */
53     return 0;
54 }
```



## ステップ 24 ワークの解答

### level 1

1. 1 2
2. 1 要素数、2 2、3 3
3. 1 行、2 列

### level 3

1. 3 行 4 列の行列をスカラ倍して出力する

リスト 50 3 行 4 列の行列をスカラ倍して出力する

```

1  #include <stdio.h>
2  int main(void){
3      int a[3][4] = { { 1, 2, 3, 4 },
4                      { 5, 6, 7, 8 },
5                      { 9,10,11,12 } };
6      int row = 3;
7      int col = 4;
8      int c = 0;
9      for ( int i = 0; i < row; ++i ){
10         for ( int j = 0; j < col; ++j ){
11             printf("a[%d][%d] = %3d ", i+1,j+1,a[i][j]);
12         }
13         printf("\n");
14     }
15     printf("c >");
16     scanf("%d",&c);
17     for ( int i = 0; i < row; ++i ){
18         for ( int j = 0; j < col; ++j ){
19             a[i][j] = c * a[i][j];
20         }
21     }
22     for ( int i = 0; i < row; ++i ){
23         for ( int j = 0; j < col; ++j ){
24             printf("a[%d][%d] = %3d ", i+1,j+1,a[i][j]);
25         }
26         printf("\n");
27     }
28     return 0;
29 }
```

2. 3 行 3 列の行列の足し算

リスト 51 3 行 3 列の行列の足し算

```

1  #include <stdio.h>
2  int main(void){
3      int a[3][3] = { { 1, 2, 3 },
4                      { 4, 5, 6 },
5                      { 7, 8, 9 } };
6      int b[3][3] = { {10,11,12 },
7                      {13,14,15 },
8                      {16,17,18 } };
9      int row = 3;
10     int col = 3;
11     for (int i = 0; i < row; ++i ){
12         for (int j = 0; j < col; ++j ){
13             a[i][j] = a[i][j] + b[i][j];
14         }
15     }
16     for (int i = 0; i < row; ++i ){
17         for (int j = 0; j < col; ++j ){
18             printf("c[%d][%d] = %3d ", i+1, j+1, a[i][j]);
19         }
20         printf("\n");
21     }
22     return 0;
23 }

```

## 3. 2 クラスの国語テストの成績集計

リスト 52 2 クラスの国語テストの成績集計

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int score[2][10]
5      = { { 56, 80, 66, 70, 45, 95, 56, 77, 82, 73 }, /* 1 組 */
6          { 78, 30, 88, 69, 70, 50, 72, 100, 28, 46 } }; /* 2 組 */
7      int sum1 = 0;
8      int sum2 = 0;
9      for ( int count = 0; count < 10; ++count ){
10         sum1 += score[0][count];
11         sum2 += score[1][count];
12     }
13     printf("class A sum = %d , ave = %f \n", sum1, sum1/10.);
14     printf("class B sum = %d , ave = %f \n", sum2, sum2/10.);
15     printf("all sum      = %d , ave = %f \n", sum1 + sum2, (sum1 + sum2)/20.);
16     return 0;
17 }

```

## ステップ 25 ワークの解答

### level 1

1. 1 数値リテラル、2 マクロ定数
2. 1 継続条件式
3. 1 1行コメント
4. 1 文字列、2 char
5. 1 \0、2 { 'a','b','c','\0' }

### level 3

1. 3行3列の行列データを保持し表示する

リスト 53 3行3列の行列データを保持し表示する

```

1 #include <stdio.h>
2 #define ROW 3
3 #define COL 3
4 int main(void){
5     int a[ROW][COL] = { { 1, 2, 3 },
6                           { 4, 5, 6 },
7                           { 7, 8, 9 } };
8     for (int i = 0; i < ROW; ++i){
9         for (int j = 0; j < COL; ++j){
10             printf("a[%d][%d] = %d ", i+1, j+1, a[i][j]);
11         }
12         printf("\n");
13     }
14     return 0;
15 }
```

2. マクロ定数を使って成績の集計プログラムを書き換える

リスト 54 マクロ定数を使って成績の集計プログラムを書き換える

```

1 #include <stdio.h>
2 #define CLASSES 2
3 #define MEMBERS 10
4 int main(void)
5 {
6     int score[CLASSES][MEMBERS]
7     = { { 56, 80, 66, 70, 45, 95, 56, 77, 82, 73 }, /* 1 組 */
8         { 78, 30, 88, 69, 70, 50, 72, 100, 28, 46 } }; /* 2 組 */
9     int sum1 = 0;
```

```
10     int sum2 = 0;
11     for ( int count = 0; count < MEMBERS; ++count ){
12         sum1 += score[0][count];
13         sum2 += score[1][count];
14     }
15     printf("class A sum = %d , ave = %f \n",sum1, sum1/10.);
16     printf("class B sum = %d , ave = %f \n",sum2, sum2/10.);
17     printf("all sum      = %d , ave = %f \n", sum1 + sum2, (sum1 + sum2)/20.);
18     return 0;
19 }
```

3. キーボードから名前を入力して、それをそのまま表示する。

リスト 55 キーボードから名前を入力して、それをそのまま表示する。

```
1  #include <stdio.h>
2  #define STR_SIZE 20
3  int main(void)
4  {
5      char yourName[STR_SIZE];
6      printf("What is your name? > ");
7      scanf("%20s",yourName);
8      printf("Your name is %s .\n",yourName);
9      return 0;
10 }
```

## ステップ 26 ワークの解答

### level 1

1. 標準入出力関数、2 標準 C ライブラリ
2. 1 `stdio.h`、2 `stdlib.h`
3. 1 `abs`、2 `rand`、3 `atoi`

### level 3

1. ジャンケンプログラム

リスト 56 ジャンケンプログラム

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int pcHand = 0;
6     int userHand = 0;
7     while (userHand != -1){
8         printf("Jan Ken ?[Goo=0,Choki=1,Pa=2,Quit=-1]>");
9         scanf("%d",&userHand);
10        if (userHand > 2) {
11            printf("Wrong hand.\n");
12        } else {
13            if (userHand == -1) {
14                break;
15            }
16            pcHand = rand() % 3;
17            printf("PC hand is %d : ",pcHand);
18            if ((userHand+1)%3 == pcHand){
19                printf("You win!\n");
20            } else if (userHand == pcHand){
21                printf("Let's do again!\n");
22            } else {
23                printf("You lose!\n");
24            }
25        }
26    }
27    return 0;
28 }
```

2. 数当てゲーム

リスト 57 数当てゲーム

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int guess = -1;
6     int answer = rand() % 10 + 1;
7     while( guess != answer ){
8         printf("Guess a number [1..10] >");
9         scanf("%d",&guess);
10        if (guess == answer){
11            printf("HIT!!\n");
12        } else {
13            printf("MISS!\n");
14        }
15    }
16    printf("Game over.\n");
17    return 0;
18 }
```

### 3. rand 関数でサイコロ

リスト 58 rand 関数でサイコロ

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int dice = rand()%6+1;
6     printf("Dice rolled [ %d ].\n",dice);
7     return 0;
8 }
```

解答例に示した rand 関数を使ったサイコロのプログラムには問題があります。次の実行画面例をみてください。何度実行しても同じ目しか出ていません。これは rand 関数の乱数発生仕組みによるものです。rand 関数は「あらかじめ用意してある乱数列」を常に同じ順に出力します。これを本当のサイコロのようにするには、「あらかじめ用意してある乱数列」の並べ替え命令を実行しなければなりません。今回の学習の範囲を超えるので詳細を省きますが、学習意欲のある方は、srand 関数と時間関数ライブラリ time.h について調べて、学習を深めてください。

画面出力

```
pc$ ./a.out
Dice rolled [ 2 ].
pc$ ./a.out
Dice rolled [ 2 ].
pc$ ./a.out
Dice rolled [ 2 ].
pc$ ./a.out
Dice rolled [ 2 ].
pc$ ./a.out
Dice rolled [ 2 ].
```

## ステップ 27 ワークの解答

### level 1

1. 1 sqrt、2 pow
2. 1 ラジアン、2 度
3. 1 floor、2 ceil
4. 1 exp、2 自然対数、3 常用対数
5. 1 clock、2 time

### level 3

1. 等加速度直線運動の移動距離を計算する

リスト 59 等加速度直線運動の移動距離を計算する

```

1 #include <stdio.h>
2 #include <math.h>
3 int main(void)
4 {
5     int v0 = 5; /* [km/h] */
6     int a = 10; /* [m/2^2] */
7     int t = 10; /* [s] */
8     float x = v0 * t + 1. / 2 * a * pow(t, 2);
9     printf("x = %f\n", x);
10    return 0;
11 }
```

2. 斜方投射の到達高とその水平距離

リスト 60 斜方投射の到達高とその水平距離

```

1 #include <stdio.h>
2 #include <math.h>
3 int main(void)
4 {
5     int v0 = 0; /* [m/s] */
6     float theta = 0; /* [deg] */
7     float h = 0; /* [m] */
8     float l = 0; /* [m] */
9     float g = 9.8; /* [m/s^2] */
10    printf("v0 > [int] ");
11    scanf("%d",&v0);
12    printf("theta > [int, degree] ");
13    scanf("%f",&theta);
```



```

14     theta = 3.1415 * theta / 180; /* [rad] */
15     printf("theta = %f\n", 1.0*theta);
16     h = pow(v0,2) * pow(sin(theta),2)/(2*g);
17     l = pow(v0,2) * sin(theta)*cos(theta)/g;
18     printf("h = %f[m], l = %f[m]\n", h, l);
19     return 0;
20 }

```

### 3. カウントダウンタイマー

リスト 61 カウントダウンタイマー

```

1  #include <stdio.h>
2  #include <time.h>
3  int main(void)
4  {
5      int duration = 0;
6      time_t currentTime = 0;
7      time_t previousTime = 0;
8      printf("Duration > [int]");
9      scanf("%d",&duration);
10     time(&previousTime);
11     while(duration > 0){
12         time(&currentTime);
13         duration = duration - (currentTime - previousTime);
14         previousTime = currentTime;
15         printf("**%d**\n", duration);
16     }
17     printf("**DONE**\n");
18     return 0;
19 }

```

### 4. モンテカルロ法で円周率

計算式の中で、整数同士の割り算を行うと小数点以下の計算を行ってくれないため、莉式に工夫が必要です。諦めずに地道に立式してください。

リスト 62 モンテカルロ法で円周率

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  int main(void)
5  {
6      float pi = 0;
7      float x = 0; /* 生成した点の座標x */
8      float y = 0; /* 生成した点の座標y */
9      float r = 0; /* 生成した点と中心までの距離 */
10     int hit = 0; /* 生成した点が円の中なら+1 */
11     int max_count = 10000000; /* 試行回数 */
12     printf("Wait a minute....\n");

```

```
13     for (int count = 0; count < max_count; ++count){
14         x = (rand()%1000)/1000.;
15         y = (rand()%1000)/1000.;
16         r = sqrt( pow(x,2) + pow(y,2) );
17         if (r <= 1) {
18             ++hit;
19         }
20     }
21     printf("hits %d / trys %d.      done\n",hit,max_count);
22     pi = (4. * hit) / max_count;
23     printf("Approximation of PI is %f\n",pi);
24     return 0;
25 }
```

## ステップ 28 ワークの解答

### level 1

- 1 ユーザ関数
- 1 引数、2 戻り値
- 1 ユーザ関数

### level 3

1. 現在の日時を表示する関数 `printDate()`

リスト 63 現在の日時を表示する関数 `printDate()`

```
1 #include <stdio.h>
2 #include <time.h>
3 void printDate(void)
4 {
5     time_t timer;
6     time(&timer);
7     printf("%s\n", ctime(&timer));
8 }
9
10 int main(void)
11 {
12     printDate();
13     return 0;
14 }
```

2. 棒グラフを描く関数 `barGraph`

リスト 64 棒グラフを描く関数 `barGraph`

```
1 #include <stdio.h>
2 void barGraph(length)
3 int length;
4 {
5     for(int i=0; i < length; i++){
6         printf("*");
7     }
8     printf("\n");
9 }
10 int main(void)
11 {
12     int length = 0;
13     for(int i = 1; i<=10; ++i){
```

```
14         barGraph(i);
15     }
16     return 0;
17 }
```

### 3. 円筒形の断面積を計算する関数

リスト 65 円筒形の断面積を計算する関数

```
1  #include <stdio.h>
2  #include <math.h>
3  float calcCrossSection(d)
4  float d;
5  {
6      return (M_PI * pow(d,2))/4;
7  }
8  int main(void)
9  {
10     float diameter = 0;
11     printf("Enter diameter > [float]");
12     scanf("%f",&diameter);
13     printf("A cross section is %f\n",calcCrossSection(diameter));
14     return 0;
15 }
```

## ステップ 29 ワークの解答

### level 1

1. 1 再帰呼び出し、2 関数
2. 1 シンプル、2 実行速度、3 メモリ使用量

### level 3

1. 再帰呼び出しでユークリッドの互除法

リスト 66 再帰呼び出しでユークリッドの互除法

```
1 #include <stdio.h>
2 int gcd(a,b)
3 int a;
4 int b;
5 {
6     if (a == b){
7         return a;
8     }
9     if (b < a) {
10         return gcd(a-b,b);
11     } else {
12         return gcd(a,b-a);
13     }
14 }
15 int main(void)
16 {
17     int a = 0;
18     int b = 0;
19     printf("Enter 2 integers > [int int]");
20     scanf("%d %d",&a,&b);
21     printf("gcd = %d\n",gcd(a,b));
22     return 0;
23 }
```

2. ハノイの塔の操作を再帰呼び出しで解く

リスト 67 ハノイの塔の操作を再帰呼び出しで解く

```
1 #include <stdio.h>
2 void hanoi(num,disc1,disc2,disc3)
3 int num;
4 int disc1;
5 int disc2;
```

```
6  int disc3;
7  {
8      if (0 < num){
9          hanoi(num-1,disc1 , disc3 , disc2 );
10         printf(" Disc No.%d %d move to %d\n",num,disc1 , disc2 );
11         hanoi(num-1,disc3 , disc2 , disc1 );
12     }
13 }
14 int main(void)
15 {
16     int n = 3;
17     hanoi(n,1 , 2 , 3);
18     return 0;
19 }
```

## ステップ 30 ワークの解答

### < 語彙 >

1. (1) 手順、(2) 電子回路、(3) 使い方、(4) 約束事、(5) 機械語、あるいはマシン語、(6) 電子回路、(7) OS、(8) 高
2. (1) フリー・フォーマット、(2) 字下げ、あるいはインデント、(3) コメント、(4) /\*、(5) \*/、(6) 英語
3. (1) 予約語、(2) 識別子
4. (1) 前処理指令部、(2) 関数定義部、(3) コンパイラ、(4) コンパイル、(5) バグ、(6) デバッグ
5. (1) 四則、(2) \*、(3) /、(4) ゼロ除算、(5) 即値、(6) リテラル、(7) 剰余、(8) %
6. (1) 数値、(2) 2、(3) 整数、(4) 2 の補数、(5) 16
7. (1) メモリ、(2) 名前、(3) 英語、(4) int、(5) float
8. (1) 英語、(2) 6、(3) 31、(4) ローワーキャメル
9. (1) 標準出力、(2) printf、(3) 拡張表記、(4) エスケープシーケンス、(5) 書式
10. (1) scanf、(2) &、(3) アンパサンド、(4) アドレス
11. (1) フローチャート、(2) 矢印、(3) 順次、(4) 選択、(5) 繰り返し
12. (1) 1、(2) 1 0、(3) 2
13. (1) 分岐、(2) if、(3) 条件、(4) else
14. (1) 真、(2) 偽、(3) 論理積、(4) 論理和、(5) 排他的論理和
15. (1) 整数値、(2) switch、(3) case、(4) default、(5) break
16. (1) 継続条件、(2) 無限ループ、(3) break
17. (1) 初期化、(2) 継続条件、(3) 再初期化、(4) break
18. (1) A、(2) B、(3) 入れ子
19. (1) 連続、(2) 型、(3) 配列名、(4) 要素、(5) 添え字、(6) 1 次元
20. (1) ソート、(2) サーチ
21. (1) 2
22. (1) マクロ定数、(2) 文字列、(3) char
23. (1) 標準入出力関数、(2) 標準 C
24. (1) ユーザ関数、(2) 引数、(3) 戻り値、(4) ユーザ関数
25. (1) 再帰呼び出し、(2) 関数、(3) シンプル

## < プログラミング技能 >

1. プログラムの目的や著作権表示を記述した複数行コメント

リスト 68 プログラムの目的や著作権表示を記述した複数行コメント

```

1  /*
2   * author   : Atsushi Hirata
3   * purpose  : to guide a way to write comments.
4   */

```

2. (b)、(d)、(f)
3. 四則計算の結果を表示するプログラム

リスト 69 四則計算の結果を表示するプログラム

```

1  #include <stdio.h>
2  int main(void)
3  {
4      printf("(a) ans = %d\n", (5-3)/3*(9-6)/4);
5      printf("(b) ans = %d\n", (108 / 31)%5);
6      return 0;
7  }

```

4. (a) (1010)B、(b) (5)D、(c) (6A)H、(d) (1100)B、(e) (-7)D
5. 変数名の例
  - (a) 「毎朝新聞発行部数」 `circulationOfMaiaasaNewspaper`
  - (b) 「トミタ自動車株価」 `stockPriceOfTomitaMotorCorporation`
  - (c) 「斧多工業粗鋼生産量」 `productionCapacityOfCrudeSteelOnoda Industry`
6. (1)widthWindow、(2)diameterCircle、(3)numberElement、(4)elapsedTimeMinutes、(5)numberOfAllStudents
7. 直径 50mm、高さ 40mm の円錐の体積を求めるプログラム

リスト 70 直径 50mm、高さ 40mm の円錐の体積を求めるプログラム

```

1  #include <stdio.h>
2  int main(void)
3  {
4      float d = 50;
5      float r = d / 2;
6      float h = 40;
7      float pi = 3.14;
8      float v = 1. / 3 * pi * r * r * h;
9      printf("d = %f \n", d);
10     printf("h = %f \n", h);
11     printf("v = %f \n", v);

```



```

12     return 0;
13 }

```

#### 8. 整数値 255 を 10 進数と 16 進数で表示する

リスト 71 整数値 255 を 10 進数と 16 進数で表示する

```

1 #include <stdio.h>
2 int main(void)
3 {
4     printf("(%d),(%x)\n",255,255);
5     return 0;
6 }

```

#### 9. 立方体の一辺の長さを入力して体積を求めるプログラムの解答例

リスト 72 立方体の一辺の長さを入力して体積を求めるプログラムの解答例

```

1 #include <stdio.h>
2 #include <math.h>
3 int main(void)
4 {
5     int l = 0;
6     printf("Input length of a cube > [int]");
7     scanf("%d",&l);
8     printf("Volume = %f [cm^3]. \n",pow(l*0.1,3));
9     return 0;
10 }

```

#### 10. 階乗のフローチャート

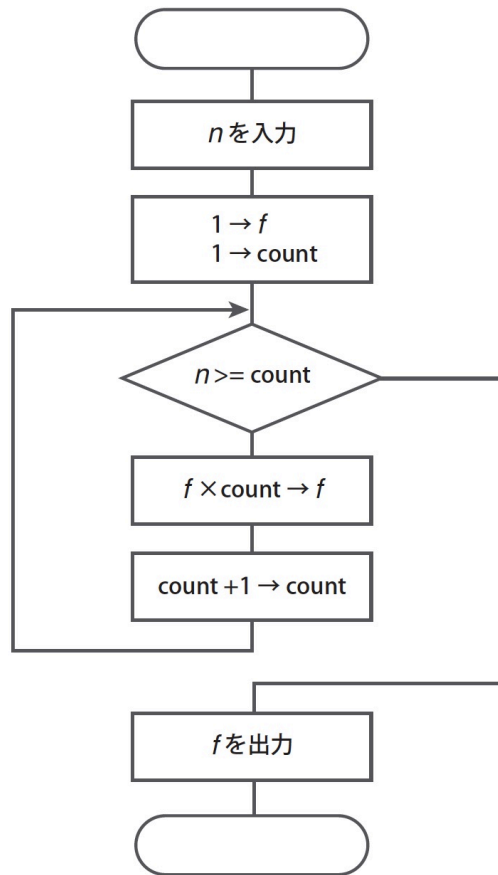
#### 11. ループ端の図記号を使った階乗のフローチャート

リスト 73 自然数、ゼロ、負の数の判定

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int val = 0;
5     printf("Enter Integer > ");
6     scanf("%d",&val);
7     if (val > 0) {
8         printf("%d is positive number.\n",val);
9     } else {
10         if (val == 0) {
11             printf("%d is zero.\n",val);
12         } else {
13             printf("%d is negative number.\n",val);
14         }
15     }
16     return 0;
17 }

```

図 8  $n$  の階乗を求めるフローチャート

## 12. どちらかだけ正の数の時”YoooHooo”

リスト 74 どちらかだけ正の数の時”YoooHooo”

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int a = 0;
5      int b = 0;
6      printf("Input two integer numbers.\n");
7      printf("1st > ");
8      scanf("%d",&a);
9      printf("2nd > ");
10     scanf("%d",&b);
11     if (((a > 0) && (b < 0)) || ((a < 0) && (b > 0))) {
12         printf("YoooHooo!\n");
    
```

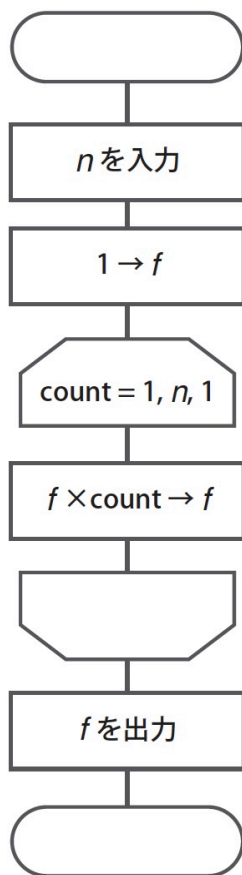


図9 ループ端の図記号を使った階乗のフローチャート

```

13     } else {
14         printf("...\n");
15     }
16     return 0;
17 }

```

## 13. switch 文を使って偶奇の判定

リスト 75 switch 文を使って偶奇の判定

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int num = 0;
5     printf("Enter a number > ");

```

```

6     scanf("%d",&num);
7     switch ( num % 2 ){
8         case 0:
9             printf("%d is even number.\n",num);
10            break;
11        case 1:
12            printf("%d is odd number.\n",num);
13            break;
14    }
15    return 0;
16 }

```

#### 14. while 文を使った階乗のプログラム

リスト 76 while 文を使った階乗のプログラム

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int num = 0;
5      printf("Enter a positive number > ");
6      scanf("%d",&num);
7      int count = 1;
8      int fact = 1;
9      while( count <= num ){
10         fact = fact * count;
11         ++count;
12     }
13     printf("%d! = %d\n",num,fact);
14     return 0;
15 }

```

#### 15. for 文を使った階乗のプログラム

リスト 77 for 文を使った階乗のプログラム

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int num = 0;
5      printf("Enter a positive number > ");
6      scanf("%d",&num);
7      int fact = 1;
8      for (int count = 1; count <= num; count++){
9         fact = fact * count;
10     }
11     printf("%d! = %d\n",num,fact);
12     return 0;
13 }

```

#### 16. 偶数のマスには O、奇数のマスには X

リスト 78 偶数のマスには O、奇数のマスには X

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int count = 1;
5      for(int i = 0; i < 3; ++i){
6          for(int j = 0; j < 3; ++j){
7              if ( count % 2 == 0 ){
8                  printf("O");
9              } else {
10                 printf("X");
11             }
12             count++;
13         }
14         printf("\n");
15     }
16     return 0;
17 }

```

## 17. 合格者の人数調査

リスト 79 合格者の人数調査

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int data[5] = {45,80,75,28,95};
5      int passingMark = 70;
6      int count = 0;
7      for(int i = 0; i < 5; ++i){
8          if(passingMark <= data[i]){
9              count++;
10         }
11     }
12     printf("Successfals are %d.\n",count);
13     return 0;
14 }

```

## 18. ソートしサーチする

リスト 80 平均点に最も近い点を表示する

```

1  #include <stdio.h>
2  #include <math.h>
3  int main(void)
4  {
5      int data[5] = {45,80,75,28,95};
6      int temp = 0;
7      //get average
8      float average = 0.0;
9      for(int i = 0; i < 5; ++i){

```



```

10     int col = 3;
11     for (int i = 0; i < row; ++i){
12         for (int j = 0; j < col; ++j){
13             a[i][j] = a[i][j] - b[i][j];
14         }
15     }
16     for (int i = 0; i < row; ++i){
17         for (int j = 0; j < col; ++j){
18             printf("c[%d][%d] = %3d ", i+1, j+1, a[i][j]);
19         }
20         printf("\n");
21     }
22     return 0;
23 }

```

20. キーボードから姓名を入力し、表示する。

リスト 82 キーボードから姓名を入力し、表示する

```

1  #include <stdio.h>
2  #define STR_SIZE 40
3  int main(void)
4  {
5      char yourFirstName[STR_SIZE];
6      printf("Enter your first name > ");
7      scanf("%40s", yourFirstName);
8      char yourLastName[STR_SIZE];
9      printf("Enter your last name > ");
10     scanf("%40s", yourLastName);
11     printf("Your first name is %s, your last name is %s .\n",
12           yourFirstName, yourLastName);
13     return 0;
14 }

```

21. 綴り当てクイズ

リスト 83 綴り当てクイズ

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #define HIDES 3
4  #define STR_LENGTH 7
5  int main(void)
6  {
7      char question[STR_LENGTH] = "flower";
8      int hidingPos[STR_LENGTH];
9      for (int i = 0; i < STR_LENGTH; i++) {
10         hidingPos[i] = 0;
11     };
12     for (int i = 0; i < HIDES; ++i){
13         hidingPos[rand() % STR_LENGTH] = 1;

```

```

14     }
15     char maskedQuestion[STRLENGTH];
16     for (int i = 0; i < STRLENGTH; i++){
17         if (hidingPos[i] == 1) {
18             maskedQuestion[i] = '*';
19         } else {
20             maskedQuestion[i] = question[i];
21         }
22     }
23     printf("What is this word? [%s] > ", maskedQuestion);
24     char answer[STRLENGTH];
25     int isGoodAnswer = 1; // 1 is good, 0 is bad.
26     scanf("%s", answer);
27     printf("Answer is %s.\n", answer);
28     //check
29     for (int i = 0; i < STRLENGTH; i++){
30         if ( question[i] == answer[i] ) {
31             // still good.
32         } else {
33             // not same.
34             isGoodAnswer = 0;
35         }
36     }
37     if (isGoodAnswer == 1) {
38         printf("Correct answer!\n");
39     } else {
40         printf("Not correct!\n");
41     }
42     return 0;
43 }

```

## 22. clock 関数を使ってよりサイコロらしいサイコロプログラムを

リスト 84 clock 関数を使ってよりサイコロらしいサイコロプログラムを

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int main(void){
5      int emptyRolls = clock() % 100;
6      for (int i = 0; i < emptyRolls; ++i){
7          int emptyRoll = rand();
8      }
9      printf("Dice rolled. Val is %d.\n", (rand()%6+1));
10     return 0;
11 }

```

## 23. 前の問題で作成したプログラムを元に、サイコロ関数

リスト 85 前の問題で作成したプログラムを元に、サイコロ関数



```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int rollDice6(void){
5      int emptyRolls = clock() % 100;
6      for (int i = 0; i < emptyRolls; ++i){
7          int emptyRoll = rand();
8      }
9      return rand()%6+1;
10 }
11
12 int main(void){
13     for (int i = 0; i < 5; ++i){
14         printf("Dice rolled. Val is %d.\n",rollDice6());
15     }
16     return 0;
17 }

```

24. n までの数の総和を求める再帰呼び出し関数 sigma

リスト 86 n までの数の総和を求める再帰呼び出し関数 sigma

```

1  #include <stdio.h>
2  #define TARGET 5
3  int sigma(n)
4  int n;
5  {
6      if ( n < 0 ) {
7          return -1;
8      } else if ( n == 0 ){
9          return 0;
10     } else if ( n == 1 ){
11         return 1;
12     } else {
13         return n + sigma( n - 1 );
14     }
15 }
16
17 int main(void)
18 {
19     printf("sigma(%d) = %d\n",TARGET,sigma(TARGET));
20     return 0;
21 }

```

## トレース

### トレースとは

トレース とは、プログラムの実行過程を人間がステップごとに確認する作業のことです。変数の値やオブジェクトの内部状態を一覧表の形にし、刻々と更新することで、プログラムが目的の動作をしているのかを確かめます。

### トレースの具体例

リスト 87 をトレースします。

リスト 87 1 から 3 までの整数を合計するプログラム

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int sum = 0;
5     for ( int i = 1; i <= 3; ++i ){
6         sum += i;
7     }
8     printf("sum is %d\n",sum);
9     return 0;
10 }
```

これをトレースします。まずはプログラムの実行過程を追いかけるために必要な変数を網羅した表を描きましょう。実行ステップの列に 1 から 4 の数値を入れたのは、まずはこのぐらいのステップを追いかけてみようという意図で、4 行で終わるプログラムだと宣言しているわけではありません。

表 3 リスト 87 のトレース表 001

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      |     |   |    |
| 2      |     |   |    |
| 3      |     |   |    |
| 4      |     |   |    |

ソース・プログラムを見ながら、上から順に頭の中で実行し、その様子をトレース表に反映します。

4 行目で変数 `sum` が宣言され、0 に初期化されますので、トレース表には次のように書き込みます。

表 4 リスト 87 のトレース表 002

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     |   |    |
| 3      |     |   |    |
| 4      |     |   |    |

5 行目で変数 `i` が宣言され、1 に初期化されますので、トレース表を更新します。

表 5 リスト 87 のトレース表 003

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      |     |   |    |
| 4      |     |   |    |

6 行目で変数 `sum` に合計操作が行われます。`sum` の値を更新します。値を更新したら、一つ前の値に打ち消し線を引きましょう。打ち消し線を引かない流儀もありますが、常に最新の値がどれか、明確になるように、古い値は打ち消しておきましょう。

表 6 リスト 87 のトレース表 004

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      | 1   |   |    |
| 4      |     |   |    |

for 文ですから実行が5行目に戻ります。ここで変数 `i` の値が更新されます。

表7 リスト87のトレース表 005

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      | 1   |   |    |
| 4      |     | 2 |    |

6行目で変数 `sum` の値が更新されます。実行ステップ数の列を少し増やしておきましょう。

表8 リスト87のトレース表 006

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      | 1   |   |    |
| 4      |     | 2 |    |
| 5      | 3   |   |    |
| 6      |     |   |    |
| 7      |     |   |    |
| 8      |     |   |    |

順次、更新されるトレース表を掲載します。

表9 リスト87のトレース表 007

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      | 1   |   |    |
| 4      |     | 2 |    |
| 5      | 3   |   |    |
| 6      |     | 3 |    |
| 7      |     |   |    |
| 8      |     |   |    |

表 10 リスト 87 のトレース表 008

| 実行ステップ | sum | i | 備考 |
|--------|-----|---|----|
| 1      | 0   |   |    |
| 2      |     | 1 |    |
| 3      | 1   |   |    |
| 4      |     | 2 |    |
| 5      | 3   |   |    |
| 6      |     | 3 |    |
| 7      | 6   |   |    |
| 8      |     |   |    |

表 11 リスト 87 のトレース表 009

| 実行ステップ | sum | i | 備考              |
|--------|-----|---|-----------------|
| 1      | 0   |   |                 |
| 2      |     | 1 |                 |
| 3      | 1   |   |                 |
| 4      |     | 2 |                 |
| 5      | 3   |   |                 |
| 6      |     | 3 |                 |
| 7      | 6   |   |                 |
| 8      |     | 4 |                 |
| 9      |     |   | 変数 i が範囲を超えた    |
| 10     |     |   | 画面に sum の値を表示する |
| 11     |     |   | 実行終了            |

以上でこのプログラムの実行が終了しました。このように、大雑把ではあれ実行ステップでトレース表の行を更新する流儀と、次のトレース表 12 のように、実行ステップで行を更新しない流儀があります。私は、学生にトレースを指導する際はトレース表 11 のように、実行ステップごとに行を更新する流儀を使います。その方が、プログラムを頭の中で実行することに不慣れな学生には理解しやすいと考えるからです。しかし、自分が必要に応じてトレースするときには、トレース表 12 のようにします。紙の上でコンパクトに見渡せるからです。

表 12 リスト 87 のトレース表 実行ステップで行を更新しない流儀

| sum | i | 備考                                      |
|-----|---|---|
| 0   | 1 |   |
| 1   | 2 |   |
| 3   | 3 |   |
| 6   | 4 | 変数 i が範囲を超えた<br>画面に sum の値を表示する<br>実行終了 |