

プログラミング入門(その2)

山本昌志*

2006年5月10日

概要

コメント文の書き方とタブの使い方, escape sequence, 変数を使ったプログラム方法について学習する. コメント文を使う理由を述べる. そして, エディターでのプログラム作成にタブを使い, プログラムを美しく書く方法を示す. また, escape sequence の動作を説明する. 最後に, 変数の意味と使い方を学ぶ.

1 先週の復習と本日の授業内容

1.1 復習

- プログラムの作成順序を学習した. 作成順序は, (1) 作業用のディレクトリーの作成, (2) エディターによるソースプログラムの記述, (3) コンパイル (C 言語を機械語に翻訳), (4) 実行となる. その順序を図1のフローチャートに示す.
- プログラムの書き方. プログラムは, 図2に示すようにおまじないと動作内容を記述する部分から構成する. プログラムを作成するときは, おまじないの部分は気にしないでワンパターンで書く. プログラマーは動作内容を考える.
- printf() 関数の使い方. この関数は動作内容のひとつで, ディスプレイに文字を表示させる. 使い方は, 図3のとおりである. \n は, エスケープシーケンス (escape sequence) のひとつで, 改行を表す.

1.2 本日の内容

授業内容 本日の学習範囲は, 教科書の2章の p.37-p46 に相当する.

ゴール 以下に示した内容が, 本日の授業の目標である.

- コメント文の書き方がわかる.
- エディター¹で, Tab の使い方がわかり, 美しいソースプログラムが書ける.
- escape sequence の Tab (\t) の動作が理解できる.
- 変数が使えて, 四則演算ができる.

*独立行政法人秋田工業高等専門学校電気工学科

¹editor: 編集者, テキスト・プログラム編集ソフト. 授業では emacs というエディターを使っている.

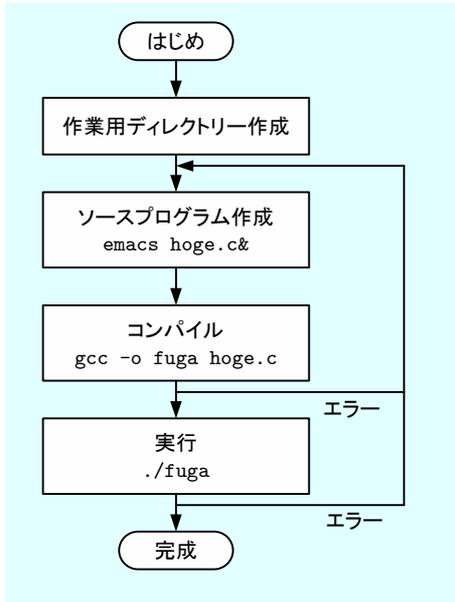


図 1: プログラムの作成手順 . ソースファイルを hoge.c , 実行ファイルを fuga としている .

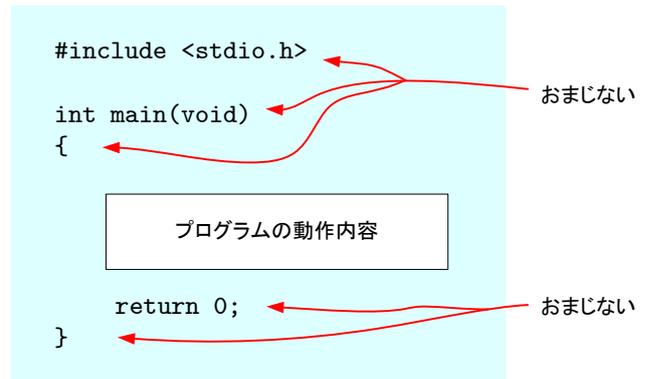


図 2: プログラムの書き方 . おまじないの部分は , 気にしないでそのまま書く .

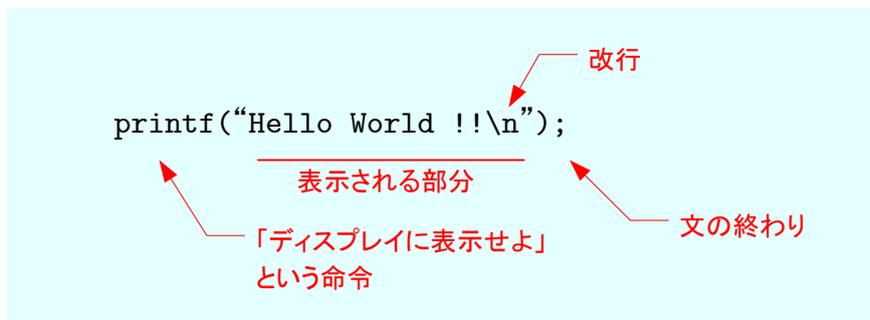


図 3: ディスプレイに表示させる printf 関数の意味

2 分かりやすいプログラムの書き方

分かりやすいプログラムが良いプログラムである。技巧をこらしたわかりにくいプログラムを書いてはならない。難解なプログラムは保守ができないため、そのうち使われなくなる。他人に自分のテクを誇示するために難解なプログラムを書きたければ、マシン語で書けばよい。

ここでは、わかりやすいプログラムを書くために、コメント文の書き方とタブ (Tab) の使い方を説明する。

2.1 インデント (p.35)

通常、行の最初を段つき (indentation) にして、プログラムの塊 (ブロック) をわかりやすくする。この段つきにする操作をインデントと言う。

行の先頭で [Tab] キーを押すことにより、インデントが施される。先週のプログラムの例をインデントしてわかり易くするためには、図4のようにする。[Tab] キーはキーボードの左上の方にある。

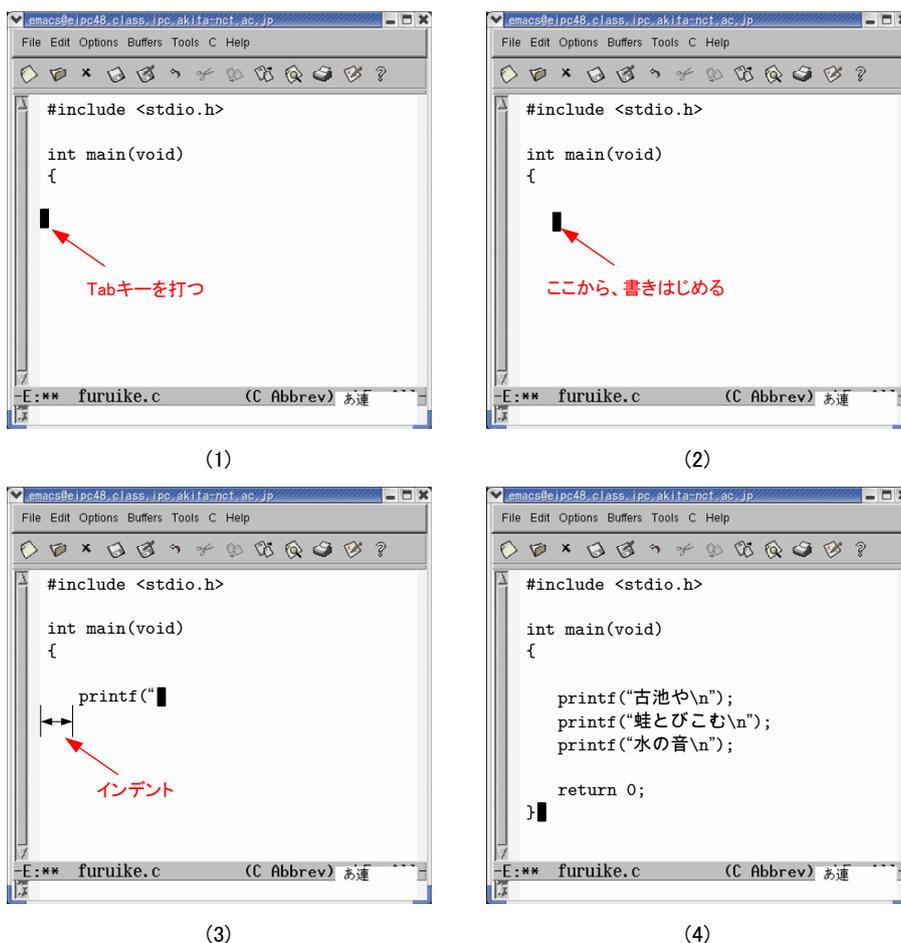


図 4: エディターでのインデントの方法

2.2 コメント文 (p.35–36)

コメント文は、プログラムの内容をわかりやすくするために記述するものである。これは、人間のためのもので、コンパイラーは無視する。プログラムを維持・管理するときの参考に用いる。良いプログラムは、コメント文が大量に書かれている。

C 言語の場合は、`/*~*/`で囲まれた部分が、コメント文となる。行をまたいでも、それは有効である。ANSI C(C90)の規格に反するが、C99規格では`//`をコメント文の開始として使える。この2つのスラッシュから、行末までがコメントとなる。秋田高専のUnixのコンパイラーはこのC99に対応している。

リスト1にコメント文が入ったソースプログラムを示す。一方、リスト2に、コメント文が無いプログラムを示す。どちらもまったく同じ実行結果になる。それどころか、コンパイルしてできた実行ファイル(機械語)も同一である。コンパイラー²はコメント文を無視する。

この程度のプログラムであれば、コメント文の御利益はわからないだろう。しかし、数百行~数千万行になる本当のプログラムを管理するとなると、コメント文が必要となる。

リスト 1: コメント文があるプログラム例

```
1 /* =====
2     このプログラムは、以下の通り作成しました。
3     目的: C言語の学習用 コメント文の使い方
4     作者: 秋田高専 山本昌志(yamamoto@akita-nct.jp)
5     ===== */
6
7 #include <stdio.h>
8
9 int main(void)
10 {
11
12     /* 松尾芭蕉の句 */
13     printf("古池や\n");      /* ふるいけや */
14     printf("蛙とびこむ\n"); // かわずとびこむ
15     printf("水の音\n");     // 水の音
16
17     return 0;
18 }
```

リスト 2: コメント文が無いプログラム例

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5
6     printf("古池や\n");
7     printf("蛙とびこむ\n");
8     printf("水の音\n");
9
10    return 0;
11 }
```

²C 言語のソースプログラムを機械語に直すことをコンパイルという。コンパイルを行うプログラムをコンパイラーと言う

3 エスケープシーケンス

注意 エスケープシーケンスは、バックスラッシュ(\)と文字からできた1文字である。キーボードの中には、バックスラッシュがないものがある。また、バックスラッシュをタイプしてもそれが打ち出されないときがある。このような場合は、すべて円記号(¥)で置き換えられる。コンピューター内部では、バックスラッシュ(\)と円記号(¥)はまったく同じ扱いである。

エスケープシーケンス これまでの学習で、改行は \n と記述すると学習した。printf() 関数のダブルクォーテーションのなかで、それを書けば画面上で改行できる。このように、改行という特殊な文字—見えないが文字と考える—を表現する方法がエスケープシーケンスである。

教科書の p.375 にエスケープシーケンスの一覧がある。いろいろあるが、普通使うのは、改行 (\n) と水平タブ (\t) である。しばらくは、この2つの動作を理解すればよい。教科書のリスト 2.8(p.39) とその上のディスプレイへの出力を見よ。

もう少し複雑な例を、図5に示す。printf() 関数を上手に使うことにより、表ができる。エスケープシーケンスの作用は、図6に示している。エスケープシーケンスの役割は、以下のとおりである。

- \t があると、次の Tab 位置まで文字位置を進める。
- \n があると、その場で改行する。

Tab の位置は、全ての行で同一である。そのため、表の各行が揃うのである。多くの場合、Tab 位置は、8 カラム—英数字 8 文字—間隔で設定されている。

```
#include <stdio.h>

int main(void)
{
    printf("[電気・磁気の単位]\n");
    printf("-----\n");
    printf("量\t量記号\t定義式\t名称\n");
    printf("=====\n");
    printf("電流\tI\t基本\tアンペア (ampere)\n");
    printf("電圧\tV\tP=VI\tボルト (volt)\n");
    printf("抵抗\tR\tR=V/I\tオーム (ohm)\n");
    printf("電荷量\tQ\tQ=It\tクーロン (coulomb)\n");
    printf("-----\n");

    return 0;
}
```

量	量記号	定義式	名称
電流	I	基本	アンペア (ampere)
電圧	V	P=VI	ボルト (volt)
抵抗	R	R=V/I	オーム (ohm)
電荷量	Q	Q=It	クーロン (coulomb)

図 5: printf() 関数を使った表の作成。左のプログラムを実行すると右の表ができる。

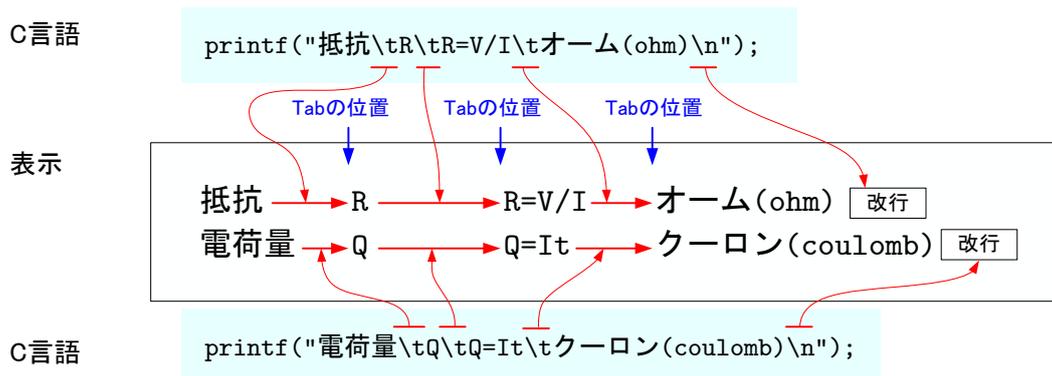


図 6: エスケープシーケンスの役割.

4 数の計算をしてみよう (p.40-46)

4.1 変数 (p.40-41)

コンピュータは計算をする機械である。絵を表示させたり、音を鳴らしたりもできるが、内部では猛烈な勢いで計算を行っている。この計算を行うためには、命令とデータが必要である。コンピュータで $3+5$ を計算する場合、 3 と 5 がデータで、 $+$ が命令である (図 7 の左)。この命令とデータを書いたものがプログラムである。

通常のプログラムでは、データが数値のまま裸で表れることは滅多に無い。数値のままデータをプログラム中に記述すると、プログラムの変更が大変難しくなる。このことは、今は理解できなくてもよい。しばらく経験を積みればわかる。当面の間、諸君の作成するプログラムには裸でデータを書いても問題ない。

データは裸の数値で表すのではなく、変数という入れ物に入れる。その入れ物—変数—には名前が付けられている。命令は、その名前を指定して、データを操作する (図 7 の右)。この辺は、プログラムを作れば、すぐに理解できる。

図 7 を見ると、C 言語で使う変数は数学と同じではないか—と思うかもしれない。しばらくは、数学と同じと考えてよい。もう少し、進んだ学習をすると、数学との違いが分かってくる。

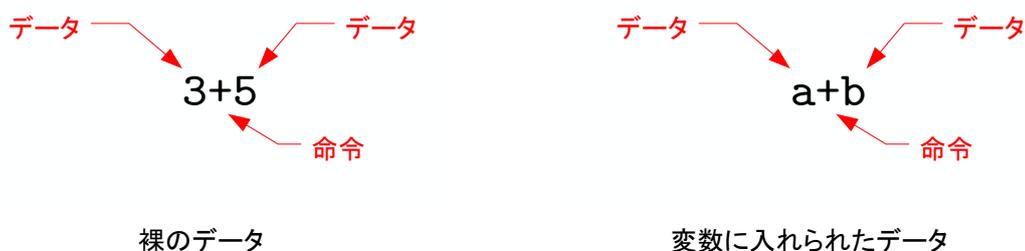


図 7: 命令とデータの表現方法の違い.

4.2 変数を使うためには

変数を使うためには、定義が必要である。数学では勝手に使えたが、C言語のプログラムでは必ず定義が必要である。例えば、整数 (integer) を入れる入れ物—整数型の変数—を使うためには、

```
int a,b,c;
```

と定義を書く。これにより、整数型の変数 a, b, c が使えるようになる。コンピューターにこの変数を使いますよと教えているのである。

この定義を書く場所も決まっている。プログラムの先頭—関数の先頭—にまとめて、書かなくてはならない。変数を使ってプログラムは、図8のように書く。

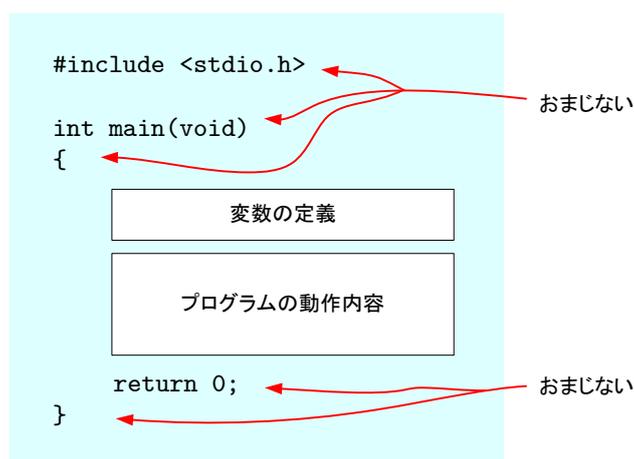


図 8: 変数を使う場合のプログラムの書き方

4.3 四則演算と代入演算子 (p.42–43)

計算の演算子 教科書 p.43 の表に、使用頻度の高い演算子がまとめられている。余りを計算する演算子 (%) 以外は、説明するまでもなく分かるだろう。余りを計算する演算子は、 $10\%4$ とすると、計算結果は 2 となる。割り算の余りを計算している。

割り算の演算 (/) も気を付けることがある。整数/整数の結果は整数になり、小数部は切り捨てられる。整数の割り算を行う場合、このことを忘れては成らない。

代入演算子 C言語のイコール (=) は代入演算子と呼ばれ、数学のイコールとはまったく異なる働きをする。

- 数学のイコールは、左辺と右辺が等しい—ということを表している。
- C言語のイコールは、右辺の式を計算して、左辺の変数に代入する—という操作の命令を表している。したがって、左辺は、必ず変数となる。

C 言語のイコール—代入演算子—の動作を図 9 に示す。

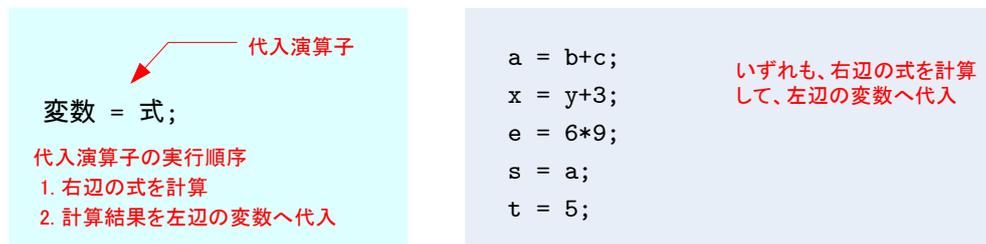


図 9: C 言語のイコール—代入演算子—の動作

4.4 整数の表示 (p.44–45)

変数に入れられた値は、`printf()` 関数を使うことにより、表示ができる (p.44)。表示の方法は、図 10 のようにする。

- 整数を書きたい部分に `%d` と書く。
- ダブルクォーテーションの後に、`%d` に対応した変数を書く。

`%d` の `d` は、10 進整数 (decimal number) を表している。

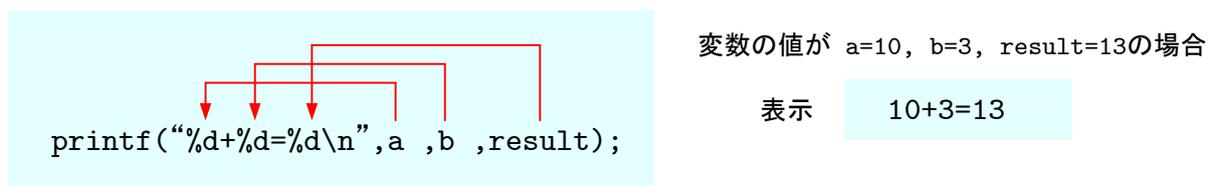


図 10: `printf()` 関数を使った変数の表示方法

5 プログラム作成の練習

[練習 1] 教科書のリスト 2.8(p.39) を参考にして、以下のような表を表示させるプログラムを作成せよ。ただし、適当にコメント文を入れること。さらに、インデントを行いプログラムを見易くすること。

電圧	V	ボルト
電流	I	アンペア
抵抗	Ohm	オーム

[練習 2] 前問のプログラムを改良して、もう少し分かりやすい以下の表を作成せよ。

電気量	単位	読み方
電圧	V	ボルト
電流	I	アンペア
抵抗	Ohm	オーム

[練習 3] 教科書のリスト 2.11(p.45) を参考に、以下を計算するプログラムを作成せよ。

変数の値	$a = 3$	$b = 4$	$c = 5$
計算すべき式	$a + b + c$	$a \times b \times c$	$a \times a + b \times b + c \times c$

[練習 4] 抵抗の両端にかかる電圧を表示するプログラムを作成せよ。ただし、電圧はオームの法則 ($V = IR$) から計算すること。

抵抗	10	オーム
電流	3	アンペア
電圧	30	ボルト

6 課題

6.1 内容

以下の課題を実施し、レポートとして提出すること。

[問 1] 教科書の p.26-50 を 3 回、読め。そして、以下のことについて、3 行以内に簡単にまとめて説明せよ。

- printf の動作
- \n と \t
- %d
- 変数
- 変数定義
- 代入
- 参照

[問 2] 以下の補助単位の表を表示するプログラムを作成せよ。0.000001 を表すマイクロはギリシャ文字なので、省略した。

補助	読み方	値
p	ピコ	0.000000000001
n	ナノ	0.000000001
m	ミリ	0.001
k	キロ	1000
M	メガ	1000000
G	ギガ	1000000000

[問 3] 教科書のリスト 2.10(p.44) の変数の値の変化を説明せよ。教科書の p.43 の上半分と同様に、左側に変数定義と式を書き、右側に □ を使って説明すること。

6.2 レポート 提出要領

提出方法は、次の通りとする。評価の 20% がレポートが占める。単位の欲しい者は、レポート提出を怠るな。

期限	5月17日(水) AM 8:45 特別な理由が無い限り、1秒でも遅れたら受け取らない。 自信の無い者は、前日に提出すること。
用紙	A4のレポート用紙。左上をホッチキスで綴じて、提出のこと。
提出場所	山本研究室の入口のポスト 授業中、私に手渡ししてはならない。期限に遅れているので、受け取らない。
表紙	表紙を1枚つけて、以下の項目を分かりやすく記述すること。 授業科目名「情報処理基礎」 課題名「課題 プログラミング入門(その2)」 提出日 1E 学籍番号 氏名
内容	2ページ以降に問いに対する答えを分かりやすく記述すること。

6.3 授業欠席者

欠課の措置として、課題のレポートに加えて、以下レポートを提出すること。課題のレポートにまとめないこと。いっしょにされると、整理に困る。

[問 1] 5節—プログラム作成の練習—に示された内容のうち、2つを選択して、プログラムを作成せよ。