

# いろいろな変数

山本昌志\*

2006年6月21日

## 概要

本日から教科書の3章の変数の学習をはじめます。まずはじめに、これまでの講義の復習を行う。その後、変数が必要な理由を説明して、その使い方を示す。文字型の変数を除いて、これまで学習してきた内容と大きく変わらない。

## 1 本日の授業内容

### 1.1 前期中間試験までの復習

- プログラムの作成順序は、(1) 作業用のディレクトリの作成、(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。
  - プログラム名やディレクトリ名には、日本語 (漢字やひらがな、カタカナ) を使わないこと。英数字とアンダースコア ( \_ ) で名前を付けること。空白は絶対にダメ。
  - ソースプログラム名には、file\_name.c のようにドット シーが絶対に必要である。これで、C 言語とわかる。
- プログラムの書き方。プログラムは、図 1 に示すようにおまじないと変数の定義、動作内容を記述する部分から構成する。

---

\*独立行政法人秋田工業高等専門学校電気工学科

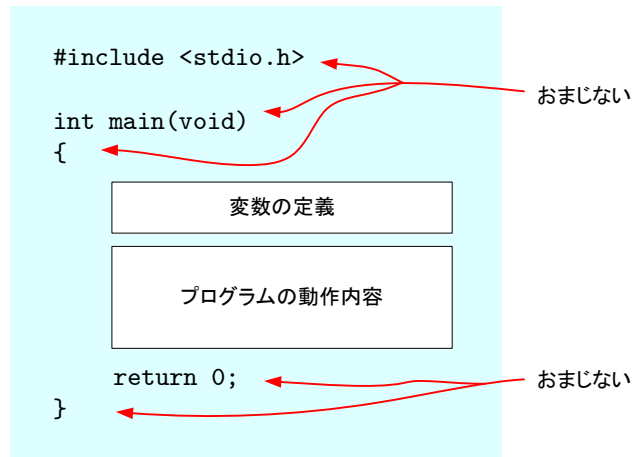


図 1: プログラムの書き方。おまじないの部分は、気にしないでそのまま書く。

- 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やデータを表示させることができる (図 2)。

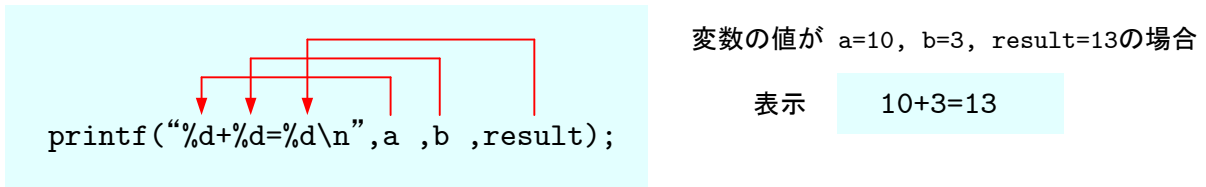


図 2: ディスプレイに表示させる printf 関数の意味

- \n は改行を, \t はタブを表すエスケープシーケンス (escape sequence) である。
- コメント文は、プログラムの動作にまったく関係が無い。プログラマーのためのメモである。/\*~\*/で囲まれた部分はコメント文である。また、//を書く于行末までコメント文になる。
- データは変数の中に入れる。変数を使うためには、変数の定義が必要である。変数定義には型名と変数名を書く。整数型の型名は int, 実数型の型名は double を使う。
- キーボードからデータ—数値—を読み込んで、その値を変数に格納することができる。整数型の変数 hoge に、キーボードから整数を代入するためには、次のように書く (図 3 も参照)。

```
scanf ("%d", &hoge);
```

倍精度実数型の変数 fuga に、キーボードから実数を代入するためには、以下のように書く。

```
scanf ("%lf", &fuga);
```

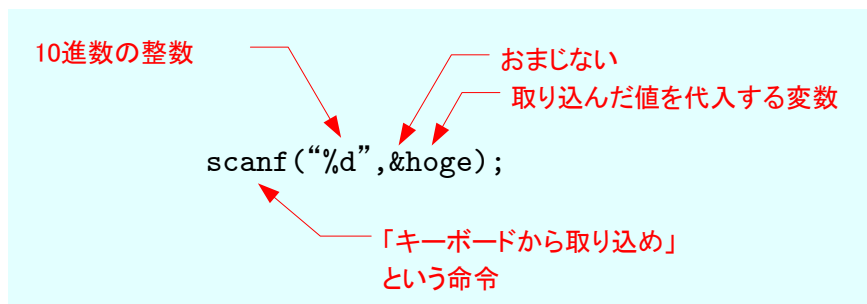


図 3: キーボードからデータを変数に取り込む scanf 関数の意味

- コンピューターの世界では、実数と整数は明確に区別され、取り扱い方法が異なる。表 1 のようにデータの型に従い、プログラムを作成する。

表 1: 整数と実数を使う場合の変数宣言など

	整数	実数
変数の定義	int	double
キーボード入力 scanf	%d	%lf
ディスプレイ出力 printf	%d	%f

- 数学の平方根  $\sqrt{x}$  を計算したい場合、C 言語では `sqrt(x)` と書く<sup>1</sup>。
- 数学の三角関数  $\sin x$ ,  $\cos x$ ,  $\tan x$  を計算したい場合、C 言語では `sin(x)`, `cos(x)`, `tan(x)` と書く。ただし、角度の単位はラジアンである。
- $\pi$  ラジアンは 180 度である。
- 数学関数を使うためには、おまじないが必要である。コンパイル時に「数学関数を使いますよ」とコンパイラーに知らせている。プログラムの先頭付近に、以下のおまじないを書くこと。

```
#include <math.h>
```

- 数学関数を使っている場合、`#include <math.h>` というおまじないに加えて、コンパイルするときも呪文を追加しなくてはならない。例えば、ソースファイル `hogehoge.c` をコンパイルして、実行ファイル `foo` を作成するためには、以下のようにする。

```
gcc -lm -o foo hogehoge.c
```

<sup>1</sup>sqrt は、square root(平方根) から来ている。

## 1.2 本日の学習内容

もう一度、変数について学習する。新たに、文字型の変数の取り扱い方について学ぶ。また、変数の型が異なる場合の問題点についても、練習問題で実際にプログラムを作成して確認する。

# 2 変数と型

## 2.1 変数とはなにか

コンピューターはデータを猛烈な速度で処理する。この処理を行うためには、計算をすることとデータを記憶することが必要となる。データはコンピューターのメモリーに記憶される。メモリーの場所に名前をつけることにより、目的のデータにアクセスする。その名前が変数名である。ようするに、変数とはメモリーの一部である。

## 2.2 変数の型

コンピューターが取り扱う代表的なデータに、文字と整数、実数がある。これの他にもデータがあるが、まだ、諸君が学習するには難しすぎる。後期の授業で説明するので楽しみにしておいてください。データには、種類があることを理解しなくてはならない。

データの種類によって、記憶する方法や必要なメモリーの大きさが異なる。そのため、データには型を決めなくてはならない。その型には文字型や整数型、実数型がある。それぞれは、表 2 のようにさらに細かく分類できる。取り扱うデータによって、最適な型を選択する必要がある。

いろいろな型があるが、諸君が使うのは以下の 3 つである。本講義では、これ以外の型を使うことはないだろう。まず、この 3 つの使い方に慣れよ。

- 1 文字を取り扱う場合は、文字型の `char` を使う。
- 整数を取り扱う場合は、整数型の `int` を使う。
- 実数を取り扱う場合は、倍精度実数型の `double` を使う。

複数の文字からなる文字列やもう少し便利な型は、後期の講義で学習する。

表 2 を見て分かるように、文字型の範囲は整数となっている。これは、文字を整数で表しているためである。教科書 p.374 の付録 1 ANSI(ASCII) コード表に文字と整数の対応が書かれている。

表 2: 変数の型と情報料

型名	データ型	ビット数	範囲
文字型	char	8	-128 ~ 127
	singed char	8	-128 ~ 127
	unsinged char	8	0 ~ 255
整数型	short int	16	-32768 ~ 32767
	signed short int	16	-32768 ~ 32767
	unsigned short int	16	0 ~ 65535
	int	32	-2147483648 ~ 2147483647
	signed int	32	-2147483648 ~ 2147483647
	unsigned int	32	0 ~ 4294967295
	logn int	32	-2147483648 ~ 2147483647
	singed long int	32	-2147483648 ~ 2147483647
	unsigned long int	32	0 ~ 4294967295
	実数型	float	32
double		64	正負とも約 $10^{-308}$ ~ 約 $10^{+308}$ 精度約 15 桁
long double		96	正負とも約 $10^{-4932}$ ~ 約 $10^{+4932}$ 精度約 18 桁

## 2.3 変数の使い方

### 2.3.1 変数定義

変数を使うためには、変数の定義が必要である。変数を定義するためには、型を指定して名前をつける。例えば、次のようにする。

```
char a, b, hoge;
int i, j, fuga;
double x, y, foo;
```

### 2.3.2 代入方法

変数に値(データ)を格納するには、いろいろな方法がある。代表的なのは、代入演算子(=)を使う方法とキーボードから入力する方法である。他にも、ファイルから読み込む方法などがあるが、これは後期の講義の範囲。

代入演算子は、以下のようにして使う。

```
a='f';
hoge=a;
```

```

i=3456;
fuga=i+789;

x=3.1415;
foo=x/9.876;

```

キーボードから読み込む場合は，以下のようにする．

```

scanf("%c",&hoge);
scanf("%d",&fuga);
scanf("%lf",&foo);

```

### 2.3.3 表示方法

データを表示するためには，以下のようにする．

```

printf("%c\n",hoge);
printf("%d\n",fuga);
printf("%f\n",foo);

```

実数型のデータの表示には，変換仕様%f を使ってきた．これだと，非常に大きな数や小さい数を表示する場合，不便である．ゼロがたくさん並ぶことになり，見苦しい．このような場合，指数形式%lf が適している．また，新しいC言語の仕様 C99 では scanf との対応を考えて，%lf が使えるようになっている．

```

printf("%e\n",foo);
printf("%lf\n",foo);

```

以上をまとめると，表3のようになる．

表 3: 整数と実数を使う場合の変数宣言など

	文字	整数	倍精度実数
変数の定義	char	int	double
キーボード入力 scanf	%c	%d	%lf
ディスプレイ出力 printf	%c	%d	%f,%lf,%e

## 3 プログラム作成の練習

[練習 1] 文字を文字型の変数に代入して，変数の値を文字型と整数型で表示せよ．いろいろな文字を打ち込んで，それが教科書 p.374 の付録 1 ANSI(ASCII) コード表と同じようになって

いることを確認せよ。 \t や \n を入れた場合どうなるか？。

ヒント 文字を変数型の変数に代入するには、

```
a='c';
```

とする。文字型の変数 a に文字 c を代入している。それを、文字型と整数型で表示させるためには、

```
printf("%c\n",a);  
printf("%d\n",a);
```

とする。

[練習 2] 文字 (1 文字) をキーボードから読み込んで、文字型の変数に代入する。そして、その変数の値を文字型と整数型で表示せよ。いろいろな文字を打ち込んで、それが教科書 p.374 の付録 1 ANSI(ASCII) コード表と同じようになっていることを確認せよ。また、タブやスペース、enter キーを入れた場合どうなるか？。

ヒント 文字型の変数をキーボードから読み込むには、

```
scanf("%c",&a);
```

とする。

[練習 3] 実数をキーボードから読み込んで、倍精度実数型の変数に代入する。そして、その変数の値を整数型と実数型で表示せよ。整数型の表示がおかしな値になることを確認せよ。

[練習 4] 実数をキーボードから読み込んで、%lf と %e 形式で表示せよ。

## 4 課題

### 4.1 内容

レポートとして提出する必要なし。

[問 1] 教科書の 2 章を 3 回通り、読むこと。分からなくても良い。分からない項目は、チェックし、次回の講義で質問すること。