変数の表示と定数

山本昌志*

2006 年 6 月 28 日

概 要

いろいろな実数の表示方法を学ぶ．そして，変数の初期化と代入できない変数の作成方法を学ぶ．

1 本日の授業内容

1.1 前回の復習

先週は教科書の 3 章のはじめで，変数について学習した．先週の講義で理解すべき内容は，以下の通り．

- コンピューターは，データを処理—計算—する機械である．そのためには，データを記憶することが必要となる．コンピューターのメモリーに記憶する．メモリーの一部に名前をつけ，記憶領域として使うことができる．名前をつけた記憶領域の一部を変数といい，その名前が変数名である．
- 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．コンピューターがデータを適切に処理するために，型を指定しなくてはならない．
  - 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．
- 変数を使うためには，おまじない `int main(void)` の次の中括弧 { に引き続いて，その定義を行わなくてはならない．使いたい変数の型と変数名を指定する．

```
int main(void){
  char a, b, hoge;
  int i, j, fuga;
  double x, y, foo;

        ⋮
```

- キーボードから読み込み込んだデータを変数に代入するためには，型と変数を指定しなくてはならない．文字型には`%c`，整数型には`%d`，実数型には`%lf` と型の指定をする．変数名の前に`&`を忘れてはならない．

*独立行政法人秋田工業高等専門学校電気工学科

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

- 変数に格納されている値を表示するときも，型を指定しなくてはならない．文字型には%c，整数型には%d をつかう．実数型の場合，%f や%lf，%e が使える．

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

- 文字型の変数に，文字を代入する場合，シングルクォーテーションで囲む．

```
hoge='A';
```

- コンピューターの内部では，文字は整数のデータとなっている．整数と文字との対応が決まっているので，整数として表すことができる．その対応が書かれたものがアスキー (ASCII) コード 表である．例えば，文字 A はコンピューター内部では整数の 65 として表せる．
- 変数には格納できる値の範囲がある．情報量が有限のため，無限のデータの格納は不可能である．

表 1: 変数の型と情報量．ビット数が情報量を表している．

型名	データ型	ビット数	範囲
文字型	char	8	-128〜127
整数型	int	32	-2147483648〜2147483647
実数型	double	64	正負とも約 10^{-308}〜約 10^{+308} 精度約 15 桁

1.2 本日の学習内容

先週に引き続き，教科書 [1] の 3 章「変数と式」の学習を行う．本日の範囲は，p.73–84 である．しかし，この中には現在の諸君の知識では理解できない内容もあるし，当面使わないことも多く書かれている．そこで，今現在諸君が身につけるべき内容を厳選して，ここでは学習する．

本日の学習のゴールは，以下のとおりである．

- 実数の様々な表示方法が分かる．教科書の表 3.8(p.75) を上手に使って，目的に応じた分かりやすい表示ができるようになること目指す．
- 変数の初期化の意味が分かる．初期化していない変数には，とんでもない値が格納されていることを理解する．
- 型修飾子 const を付けることにより，定数のように扱うことができる—ことが分かる．

2　変数の扱いかた

2.1　実数の表示のしかた

2.1.1　指数表示

実数には非常に大きな数から小さな数まであるので，いろいろな表示方法がある．大きな値の例として光の速度[1]c [m/s] は，

$$\begin{aligned} c &= 299792458 \\ &= 2.99792458 \times 10^8 \\ &= \texttt{2.99792458E+8} \\ &= \texttt{2.99792458e+8} \end{aligned}$$

のように表すことができる．2 行目が数学等でしばしば目にする方法である．3 行目は電卓の表示方法で 10 の巾乗を大文字の `E` で表現する．4 行目が C 言語で一般的な表記である．小文字の `e` をつかう．C 言語では 3 行目のように大文字の `E` も使うことができるが，小文字の方が一般的である．

このように実数を，小数部と 10 の巾乗であらわす方法を指数形式と言う．このようにすることにより，非常に大きな数や小さな数をわかり易く簡単に表現できる．

一方，小さな値の例として，電子の質量 m_e [kg] は

$$\begin{aligned} m_e &= 0.00000000000000000000000000000009109534 \\ &= 9.109534 \times 10^{-31} \\ &= 9.109534E - 31 \\ &= 9.109534e - 31 \end{aligned}$$

となる．指数形式だと小さな数を表すのも簡単である．

2.1.2　C 言語での表示

実数で表現されたデータは非常に大きな数から小さな数まである．そのため，ディスプレイに表示する場合，いろいろな方法が選択できる (教科書 p.75–76)．小数で表す`%f` と指数形式の`%e`，小数と指数の分かりやすい方を適当に判断して表す`%g` である．これから，それぞれについて説明するが，その際，倍精度実数型の変数には以下の値が代入されているものとする．`c` は光速度，`p` は円周率，`m` は電子の質量である．

```
c=2.99792458e+8;
p=M_PI;
m=9.109534e-31;
```

C 言語での指数形式の代入は，このようにするのである．キーボードからデータを入れるときも同じである．

[1] 光の速度は整数というツッコミはなし．

- %f と表示した場合

  いままで使ってきた方法で小数で表す．整数部分と小数点以下 6 桁で表示する．

```
printf("c=%f\n",c);        ⇒        c=299792458.000000
printf("p=%f\n",p);        ⇒        p=3.141593
printf("m=%f\n",m);        ⇒        m=0.000000
```

- %0.nf と表示した場合

  整数部分と小数点以下 n 桁で表示する．小数を任意の桁数で表示したいときに使う．

```
printf("c=%0.10f\n",c);      ⇒      c=299792458.0000000000
printf("p=%0.10f\n",p);      ⇒      p=3.1415926536
printf("m=%0.10f\n",m);      ⇒      m=0.0000000000
```

- %m.nf と表示した場合

  全体で少なくとも m カラム用意して，小数点以下 n 桁で表示する．m カラムと言うのは文字が書ける枠の数である．

```
printf("c=%53.40f\n",c);  ⇒  c=   299792458.0000000000000000000000000000000000000000
printf("p=%53.40f\n",p);  ⇒  p=           3.1415926535897931159979634685441851615906
printf("m=%53.40f\n",m);  ⇒  m=           0.0000000000000000000000000000009109534000
```

- %e と表示した場合

  指数形式で表示する．小数点以下は 6 桁である．

```
printf("c=%e\n",c);        ⇒        c=2.997925e+08
printf("p=%e\n",p);        ⇒        p=3.141593e+00
printf("m=%e\n",m);        ⇒        m=9.109534e-31
```

- %0.ne と表示した場合

  小数点以下 6 桁の指数形式で表示する．

```
printf("c=%0.10e\n",c);      ⇒      c=2.9979245800e+08
printf("p=%0.10e\n",p);      ⇒      p=3.1415926536e+00
printf("m=%0.10e\n",m);      ⇒      m=9.1095340000e-31
```

- %g と表示した場合

  小数での桁数が多くなった場合，指数形式で表す．私はめったに使わない．

```
printf("c=%g\n",c);        ⇒        c=2.99792e+08
printf("p=%g\n",p);        ⇒        p=3.14159
printf("m=%g\n",m);        ⇒        m=9.10953e-31
```

2.2 初期化

ここは，教科書の p.78–81 に述べていることである．

変数は定義する—型と変数名を指定—することにより，プログラム中で使うことができるようになる．定義しただけでは，その変数の中にはいい加減な値が格納されている．ゼロになっているとは限らないのである．そのため，次のようにすると変数の定義と同時に値を代入することができる．

```
int i, j=7, hoge=456;
double x, y=3.14, fuga=M_PI;
```

ここでは，整数型の j と hoge，倍精度実数型の y と fuga の値を宣言と同時に決めている．このことを初期化と言う．

もう一度言うが，変数宣言をしただけでは，格納されている値は不定である．ゼロとは限らないのである．今のところ，諸君が作成するプログラムでは初期化を忘れたことによる失敗はない．しかし，将来は必ず初期化を忘れて，間違ったプログラムを書くだろう．私も，しばしば初期化を忘れて失敗している．

2.3 型修飾子 const の使い方

ここは，教科書の p.81–82 に述べていることである．

いろいろなプログラムを作成するうちに，定数として扱いたい変数が出てくるであろう．例えば，c を光の速度として，プログラム中で定数として扱いたい場合である．代入文—例えば，c=3.14—を書かなければ済むが，うっかり代入文を書いてしまうことがある．普通のプログラマーはおっちょこちょいである．このようなミスを防ぐために，C 言語では代入のできない変数を定義することができる．次のようにするのである．

```
const double c=2.99792458e+8;
```

変数宣言の前に，型修飾子 const をつける． const の語源は constant(定数) である．

このようにすると，プログラマーが

```
c=3.1415;
```

としようものなら，コンパイラーがエラーメッセージを出し，コンパイルに失敗する．まことに，便利な機能である．

しかし，次のようにキーボードからデータを読み込む場合は，コンパイルできてしまう．

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

const を付けたからといって，安心してはならない．const があると算術代入演算子 (=) の左辺として使えないだけである．

3 プログラム作成の練習

[**練習 1**] 円周率を以下のように表示するプログラムを作成せよ．

```
3
3.1
3.14
3.142
3.1416
3.14159
3.141593
3.1415927
3.14159265
3.141592654
3.1415926536
3.14159265359
3.141592653590
3.1415926535898
3.14159265358979
```

ヒント 以前，学習したように C 言語では円周率 π は次のようにする．

- 数学に関することなので，`#include <math.h>`を書く．
- 円周率は，マクロで M_PI で表せる．今のところ，M_PI は円周率を表す変数と考える．ただし，M_PI には値の代入はできない．
- math.h をインクルード—`#include`—しているので，コンパイル時にオプション`-lm`を付ける．

[**練習 2**] 電子の質量 m_e

$$m_e = 9.109534 \times 10^{-31}\,[\mathrm{kg}]$$

を以下のように表示せよ．最初の表示は小数点以下 37 桁，次の表示は全体で 50 カラムで小数点以下 37 桁である．3 つ目は指数表示で，小数点以下 2 桁となっている．

```
0.00000000000000000000000000000009109534
            0.00000000000000000000000000000009109534
9.11e-31
```

[**練習 3**] 前問の電子の質量を%f や%e，%g で表示せよ．

[**練習 4**] 初期化されていない変数には，とんでもない値が格納されていることを以下のプログラムで確認する．

– 整数型の変数を 4 個定義する．そのうち 2 つは，123456 と 987654 で初期化する．残りの 2 つは初期化しない．
– そのまま，変数に格納されている数値を表示する．

プログラムを実行してみると，とんでもない値が格納されていることが分かるだろう．初期化は重要である．

[**練習 5**] 型修飾子 const を付けた変数に，値が代入できないことを確認する．以下のようにしてプログラムを作成せよ．

– 型修飾子 const を付けて，値が 123456 の整数変数を用意する．
– そして，その整数型の変数に代入演算子=を使って，値 987664 を代入する．
– コンパイルするとエラーメッセージがでるはずである．

4 課題

次回の講義の日 (7 月 5 日) の AM8:45 までに，以下の課題をレポートとして提出すること．表紙等は，いつもの通り．

[**問 1**] 変数名の規則について述べよ (教科書 p.62).

[**問 2**] 角度 x をキーボードから読み込んで，$\tan x$ の値を以下のように表示するプログラムを作成せよ．表示例は，角度が 89.9[度] のときである．最初の値は角度，2 番目と 3 番目は $\tan x$ の値である．

```
89.500000        114.5886501293  1.1458865013e+02
```

[**問 3**] これは次回の予習 (教科書 p.87–88)．整数型の変数 a の値を 5 とする．この場合，以下の演算結果，a の値はどうなるか．それぞれについて，答えよ．

```
a+=3        a-=3        a*=3        a/=3        a%=3
```

[**問 4**] これも次回の予習 (教科書 p.98–99)．キャストとは何か?—調べ，内容を 5 行程度にまとめよ．

参考文献

[1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング　基礎編　第 2 版. (株) オーム社, 2006.