プログラムは，わかりやすく書くことが重要です．以下に，C 言語の書法のひとつを示します．これは．C 言語の文法ではありませんので，守る必要があるわけではありません．しかし，統一した書き方をすることによって，見やすさが随分変わります．これは，他人に見せるときだけなく，自分が見るとき（エラーのチェックをするとき）にも威力を発揮します．

以下に，プログラム例を示します．

```
/*********************************************/
/*                                           */
/*  ２つの入力した整数を大きい順に並べなおす.    */
/*  input1 > input2 になるようにする.          */
/*  Mizuho SHIBATA  2008/10/31               */
/*                                           */
/*********************************************/

#include <stdio.h>

int main(void)
{
  int input1; // 入力１
  int input2; // 入力２
  int dummy;  // 値入れ替え用ダミー変数

  // 値の入力
  printf("Input number 1 : ");
  scanf("%d", &input1);
  printf("Input number 2 : ");
  scanf("%d", &input2);

  // input1 < input2 のとき値を入れ替える
  if( input1 < input2 ){
    dummy = input1;
    input1 = input2;
    input2 = dummy;
  }

  // 表示
  printf("Number 1 = %d\n", input1);
  printf("Number 2 = %d\n", input2);

  return 0;
}
```

以下に，注意事項をまとめます.

[1] **1 つの行に 1 つの内容を書く．**

教科書等では紙面の都合上,

```
int input1, input2; // 入力 1, 2
```

や,

```
printf("Input number 1 : "); scanf("%d", &input1);
```

と書くことが多いですが，慣れないうちは，これらの文は 2 行に分けることをお薦めします．エラーが出た際や修正を加える際に，ミスが減るはずです.

[2] **[Tab] キーを使ってインデント（字下げ）をする．**

[Space] キーよりも [Tab] キーの方が字下げのスペースが大きく，より見やすくなります.

[3] **コンマ (,) の後ろは，スペースを空ける．**

[4] **記号（ $=$, $+$, $<$ など）の前後にスペースを空ける．**

適度に空白を入れることによって可読性をあげます.

[5] **if 文，for 文などでは，前の括弧の後ろにスペースを，後ろの括弧の前にスペースを空ける．**

空白のない以下のプログラム

```
if(input1 > input2){
  input1 = input2;
}
```

に比べて，空白のある以下のプログラム

```
if( input1 > input2 ){
  input1 = input2;
}
```

の方が，比較している値・条件が分かりやすくなります.

[6] **コメントをつける．**

変数の意味，プログラムの概要，ブロックの説明，関数の説明などのコメントをつけると見やすいでしょう．適切なコメントは，後日自分でプログラムを見直す際や他人にプログラムを見せる際に，プログラムへの理解度をあげます．特に大きな（長い）プログラムでは必須です.

[7] **機能ごとに改行を入れる．**

前ページの例では，プログラムの機能を変数の定義，値の入力，値の入れ替え，表示の 4 つに分けています.

[8] **同じブロックの括弧は，縦にそろえて書く．**

構造が分かりやすくなるように書きます．for 文の中に if 文がある場合など，ブロックの中にブロックがある場合は，

```
for(;;){
  if( input1 > input2 ){
    break;
  }
}
```

のようにブロックをさらにインデントすることで表現します（タブを複数回押す）．このとき，if 文，for 文などの括弧を，

```
if( input1 < input2 ){
  dummy = input1;
  input1 = input2;
  input2 = dummy;
}
```

と書くか，

```
if( input1 < input2 )
{
  dummy = input1;
  input1 = input2;
  input2 = dummy;
}
```

と書くかは，好みです．柴田は，関数の際は，

```
int main(void)
{
   return 0;
}
```

を用い，if 文などのブロックは，

```
if( input1 < input2 ){
  dummy = input1;
  input1 = input2;
  input2 = dummy;
}
```

を用いるようにしています．

[9] **変数は小文字，定数は大文字で定義する．**

変数を定義する際は，

```
int input1;
```

のように小文字を，定数を定義する際は，

```
#defien PI 3.141592
```

のように define 文＋大文字で定義することが一般的です．

[10] **論理和（ && ）や論理積（ || ）を使う場合，条件を括弧で括る．**

論理和や論理積では，どこからどこまでが条件かを明示するために，

```
if( (number > 0) && (number <= 100) )
```

のように，条件を括弧で括ると見やすくなります．

前ページで挙げた項目以外にも，プログラムが大きくなるにつれて重要度が増すものがあります．以下に，列挙します．特に，**分割コンパイル**，**構造体**については，この授業では扱いませんが，大きなプログラムを書く際に，非常に重要になりますので，ぜひ調べてみるようにしてください．

- 変数，定数，関数の名前は意味が推測できるように付ける．
- 同じ機能を持つ部分は関数化する．
- ひとつの関数が大きくなりすぎないようにする（目安は 100 行以下）．
- define 文を上手に使う（マジックナンバーは極力減らす）．
- **分割コンパイル**を利用する．
- **構造体**を使う．

参考文献

1) 藤原，”C プログラミング専門課程”，技術評論社，1994.
2) 藤原，”改定新版 C プログラミング診断室”，技術評論社，2003.
3) 矢沢，”プログラミングのセオリー”，技術評論社，2008.