配列(応用)

山本昌志*

2006年12月22日

概 要

先週の応用で配列を使ったプログラミングを学習する．ファイルから配列にデータを格納する方法と，関数の引数での配列の取扱いについて学ぶ．

1 前回の復習と本日の内容

1.1 前回の復習

配列の宣言 配列と呼ばれるデータ構造は，大量のデータを格納することができる．配列はこれまで学習してきた変数—単純型のデータ構造—とほとんど同じように扱うことができ，使用方法は簡単である．配列を使うときの宣言は，

```
int hoge[100], fuga[200][200];
double foo[300], bar[4][4][4];
```

のようにする．すると，次のようなひとつずつデータの格納できる要素が使えるようになる．

- 整数が格納できる要素 `hoge[0]`〜`hoge[99]` が使えるようになる．この場合，配列の要素数は 100 個である．
- 整数が格納できる要素 `fuga[0][0]`〜`fuga[199][199]`，合計 40000 個が使えるようになる．
- 倍精度実数が格納できる要素 `foo[0]`〜`hoge[299]`，合計 300 個が使えるようになる．
- 倍精度実数が格納できる要素 `bar[0][0][0]`〜`bar[4][4][4]`，合計 64 個が使えるようになる．

配列の初期化 配列のサイズが小さい場合，宣言と同時に初期ができる．

```
int hoge[3]={111,222,333};
int fuga[2][2]={{111,222},{333,4444}};
```

配列よりも初期値が少ない場合には，残りのゼロに初期化される．多い場合にはエラーとなる．

*独立行政法人　秋田工業高等専門学校　電気情報工学科

配列の要素へのアクセス　配列では，配列名と添え字 (インデックス) を指定する—たとえば i[3] や j[25][49]—ことにより，記憶領域から値 (データ) を入出力できる．

```
i[3]=5;          /* 配列 i[3] に 5 を代入  */
c=j[25][49];     /* 配列 j[25][49] の値を変数 c へ代入 */
```

ほとんど今まで使ってきた変数と同じである．インデックスには自然数が格納された整数型の変数を使うことも可能である．

```
for(i=0; i<=360; i++){
    my_sin[i]=sin(M_PI*i/360.0);
}
```

こうすると，my_sin[45] には 0.707107$\cdots$ が格納される．

1.2　本日の学習内容

本日の学習範囲は，教科書 [1] の p.216–245 あたりである．ただし，教科書のこの範囲は少し難しい内容も含まれる．2 年生以降に学習する範囲も含まれるし，まだ学習していない数学の内容もある．したがって，教科書は分からなくてもよい．

教科書からだいぶ離れるが，本日は以下のことを学習する．

- ファイルからデータを読み込んで，配列に格納する方法を学ぶ．
- 関数に配列を渡す方法を学ぶ．

2　配列の応用

2.1　ファイルのデータを配列に格納

配列を応用したプログラムの練習を行う．2003 年 11 月の秋田市の毎日の 1 時間毎の気温のデータファイルがある．それから，日々と 11 月の最高気温を表示するプログラムを作成する．

気温は，表 1 のようなになっている．この気温の部分のみがファイル (/tmp/1e/temperature.txt) に書かれており，そのフォーマット (書式) は次のようになっている．

- 各行には，その日の 1 時間毎のデータがある．0 時〜23 時までの計 24 個である．
- 行数は 30 行で，11 月 1 日から 11 月 30 日を表している．
- ファイルには，温度のみが書かれている．時刻や日にちは書かれていない．

表 1: 11 月の気温

	0 時	1 時	2 時	3 時	4 時	5 時	···	23 時
1 日	8.4	8.0	7.3	6.5	6.0	6.1	···	10.8
2 日	11.4	11.6	11.3	10.8	10.4	9.5	···	15.8
3 日	15.9	15.8	15.2	14.9	14.5	14.3	···	15.8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
30 日	13.9	13.7	13.9	13.8	13.6	14.7	···	11.2

次のようにすれば，emacs で温度のデータが書かれているファイルの中身を見ることができる．

```
emacs /tmp/1e/temperature.txt
```

このファイルの気温を読み込んで，日々とその月の最高温度を表示するプログラムの例をリスト 1 に示す．ここでは i 日 j 時のデータを二次元配列 temp[i][j] に格納している．二次元配列については先週述べたので，ここでは詳しく説明しない．

ファイルからデータを読み出すプログラムは，まだ学習していない．2 年生のはじめに学習することになっているが，簡単に述べておく．ファイルからデータを読み出すためには，(1) ファイル情報を格納する変数の用意，(2) ファイルのオープン，(3) データの読み込み，(4) ファイルのクローズ—という一連の動作を行う．実際には，リスト 1 のとおりで，ファイル操作に関する部分を以下に示す．

- **4 行目　FILE *fp;**　ファイルの情報を格納する変数宣言．*fp がファイルの情報を格納する．FILE が型名，fp が変数[1]である．fp は fugafuga のようにどんな名前でも良い．しかし，先頭のアスタリスク (*) は必須である．
- **14 行目　fp=fopen(”/tmp/1e/temperature.txt”,”w”);**　読み込み用[2]にファイルを開いている．temperature.txt がファイル名で，その前にはスラッシュを区切りにしてディレクトリー名を書いている．絶対パスでファイルを指定している．w が書き込み用にファイルを開くことを示している．fopen() 関数の戻り値は，ファイル情報で，FILE 型の変数に代入する．
- **18 行目　fscanf(fp, ”%lf”,&temp[i][j]);**　ファイルからデータを読み込む．ファイルから読み込む fscanf() 関数は，キーボードからデータを読み込む関数 scanf() とそっくりである．読み込むファイルを指定するファイル情報の変数を最初に書く—ことが異なる．
- **22 行目　fclose(fp);**　ファイルを閉じる．使い終わったファイルはと閉じなくてはならない．

[1]本当はポインター．ポインターについては教科書の 8 章
[2]ファイルは，書き込みと読み込みができる．

リスト 1: 日々と月の最高気温を表示するプログラム.

```
1  #include <stdio.h>
2
3  int main(void){
4    FILE *fp;
5    double temp[31][24];
6    double max_day[31];
7    double max_nov;
8    int dates, hours;
9    int i, j;
10
11   dates = 30;                                    // 日数と時間の設定
12   hours = 24;
13
14   fp=fopen("/tmp/1e/temperature.txt","r");       // ファイルをオープン
15
16   for(i=1; i<=dates; i++){                       // ファイルからデータを読む
17     for(j=0; j<=hours-1; j++){
18       fscanf(fp, "%lf", &temp[i][j]);
19     }
20   }
21
22   fclose(fp);                                    // ファイルをクローズ
23
24   /* --------------- 最大と最小、平均の計算 --------------------*/
25
26   max_nov = temp[1][0];                 // 11月の仮の最高気温 1日の0時
27
28   // ---------- i日のループ ------------------------------
29   for(i=1; i<=dates; i++){              // iは、日にちを表す
30     max_day[i] = temp[i][0];            // i日の仮の最高気温 i日の0時
31
32     // ---------- j時のループ ------------------------------
33     for(j=0; j<=hours-1; j++){          // iは時刻を表す
34
35       if(temp[i][j] > max_day[i]){      // i日の最大気温の探索
36         max_day[i] = temp[i][j];
37       }
38
39     }   // ----------j時のループの終わり --------------
40
41     if(max_day[i] > max_nov){           // 11月の最大気温の探索
42       max_nov = max_day[i];
43     }
44
45   }   // ----------i日のループの終わり --------------
46
47
48   printf("\n\nTemperature  November/2003 at Akita\n");
49   printf("----------------------------------------\n");
50   printf(" day\tmax\n");
51   printf("========================================\n");
52
53   for(i=1; i<=dates; i++){
54     printf("%3d\t%5.1lf\n",i, max_day[i]);
55   }
56
57   printf("----------------------------------------\n\n");
58
59
60   printf("max(Nov.)      = %5.1lf\n", max_nov);
```

```
61
62    return 0;
63
64  }
```

実行結果

```
Temperature  November/2003 at Akita
-------------------------------------
 day    max
=====================================
  1      20.3
  2      20.4
  3      22.8
  4      14.9
  5      18.4
  :         :
  :         :
 29      15.7
 30      15.1
-------------------------------------

max(Nov.)     =  22.8
```

2.2 配列を関数に渡す

配列を関数に渡す方法を学習する．これはちょっと難しいので概略のみを説明する．本当は，後で学習するポインターと併せて理解すべき内容である．ここでは，配列のデータを関数にまたがって受け渡せす方法が分かれば良い．以降，具体例をもって説明する．

10 人の英語の成績を処理するプログラムを考える．このプログラムは，次のような動作をする．

- 10 人分の英語のテストの点数をキーボードから読み込み，それらを配列 seiseki[0]〜seiseki[9] に格納する．
- 配列に格納されたデータを使って，平均点を計算する．そして，各人の平均点からの差を配列 seiseki[0]〜seiseki[9] に格納する．これは，専用の関数 diff_ave を用いる．この関数の戻り値は平均点を示す．
- 最後に，平均点と各人のその差を表示する．

このプログラムのでは，メイン関数とユーザー定義関数 diff_ave() で，同じ配列を使用している．最初，配列 seiseki[] には英語のテストの点数が格納されており，最後には各人の平均値からの差が格納されている．実引数と仮引数のデータの渡す方法が普通の変数と配列では異なる．

- 普通の単純型の変数であれば，実引数と仮引数は全く異なるメモリーに格納される関数を呼び出したときに，実引数の値が仮引数の変数にコピーされる．そして，関数は仮引数のメモリーを使って処理を行う．
- 引数に配列を使うと，実引数と仮引数では全く同じメモリーが使われる．実引数と仮引数の名前が異なっても同じめメモリーに格納されているので，実体は同じである．したがって，呼び出された関数側で配列に格納されいるデータを変化させると，呼出側の配列の値も変わる．

配列を引数に使うと，実引数と仮引数が同じメモリーを使う理由は，処理時間を短くするためである—と言われている．通常，配列に格納されたデータは非常に大きく，関数を呼び出すたびに実引数の配列を仮引数の配列にコピーすると膨大な時間がかかり，計算コストの増大につながる．

また，仮引数の書き方も変わっている．仮引数の配列は `data[]` のようにして，配列のサイズは不要である．二次元や三次元の仮引数の場合，`fuga[][100]` や `hoge[][200][300]` のように，左端の配列のサイズを書かない．なぜ，このような変なことをするか?．これについては，ポインターを学習しないと理解できないので，ここでは説明しない．

リスト 2: 平均点とその差を表示するプログラム．

```
1  #include <stdio.h>
2
3  int diff_ave(int n, int data[]);
4
5  /*============================================================*/
6  /*   main function                                            */
7  /*============================================================*/
8  int main(void)
9  {
10   int seiseki[10], average, i, n;
11
12   n=10;                              // 学生数
13
14   for(i=0; i<n; i++){
15     printf("%d番の成績\t", i);
16     scanf("%d",&seiseki[i]);
17   }
18
19   average=diff_ave(n, seiseki);     // 関数呼び出し
20
21   printf("平均点 = %d\n", average);
22   printf("平均点との差\n");
23
24   for(i=0; i<10; i++){
25     printf("%d番\t%d点\n", i, seiseki[i]);
26   }
27
28   return 0;
29 }
30
31 /*============================================================*/
32 /*   平気値とその差の計算                                     */
33 /*============================================================*/
34 int diff_ave(int n, int data[])
35 {
36   int i, sum=0, ave;
37
38   for(i=0; i<n; i++){          // 点数の合計の計算
39     sum += data[i];
40   }
41   ave = sum/n;                 // 平均値の計算
42
43   for(i=0; i<n; i++){          // 平均値との差の計算
44     data[i] = data[i] - ave;
45   }
46
47   return ave;
48 }
```

3　プログラム作成の練習

[**練習 1**] リスト 1 のプログラムを直して，最高気温・最低気温と平均気温を以下のようにディスプレイに表示するプログラムをを作成する．

```
Temperature  November/2003 at Akita
-----------------------------------
 day     max     min    average
===================================
  1      20.3     6.0    12.1
  2      20.4     9.5    15.2
  3      22.8    13.9    17.8
  4      14.9     6.4    11.5
  5      18.4     4.3    10.6
  6      17.9    10.5    13.3
  :        :       :       :
  :        :       :       :
 25      10.6     7.6     9.2
 26       7.0     0.6     4.3
 27       7.3    -0.3     2.8
 28      12.2     0.1     6.0
 29      15.7     7.9    12.2
 30      15.1    11.2    13.4
-----------------------------------

max(Nov.)     =  22.8
min(Nov.)     =  -0.5
average(Nov.) =   9.2
```

以下の順序にしたがい，リスト 1 のプログラムを改良せよ．

1. 最低気温を表示する．
2. 平均気温を表示する．

[**練習 2**] ファイル (`/tmp/1e/int_data.txt`) には，10000 個の整数が書かれている．データ毎の平均値との差と平均値を表示するプログラムを作成せよ．平均値およびその差の計算は，ユーザー定義関数で処理すること．

4　課題

冬休み明けの最初の講義の日 (1 月 12 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．表紙のタイトルは「冬休みの課題」とすること．

[**問 1**] (復予) 教科書 p.204–268 を 3 回，繰り返し読め．そして，重要と思われる事項に赤の下線，あるいはマーカーで印を付けよ．また，よくわからないところに疑問符 (?) を付けよ．提出するレポートには「3 回読んで，印をつけた」と書くこと．

[**問 2**] (復) 以下の動作をおこなうプログラムを作成せよ．

- ファイル (`/tmp/data.txt`) は 1000 行からなり，各行には整数が書かれている．
- ファイルを読み込んで，最大値と最小値，そして平均値を表示するプログラムを作成せよ．

[問 3] (復) これは，ちょっと難しい．

- ファイル (`/tmp/data2.txt`) は 100 行からなり，各行には整数が書かれている．
- これらのデータを配列に読み込む．
- 読み込んだデータを小さい順に並び替えて，表示させるプログラムを作成せよ．ただし，データの並び替えはユーザー定義関数を使うこと．

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.