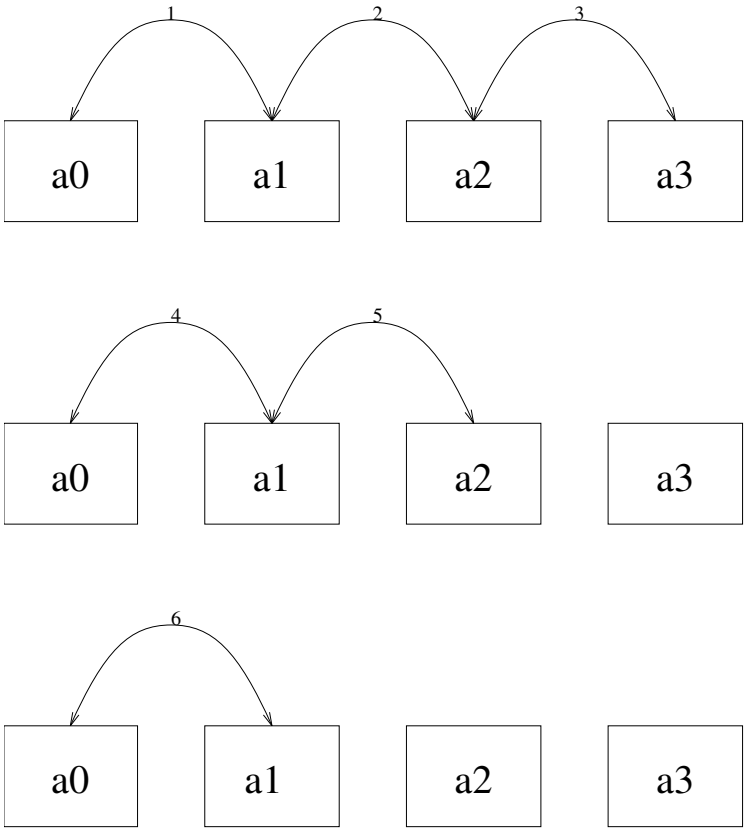
6. 配列の応用 -並べかえ (sorting)-

4 個の整数値を読み込み、それを小さい順に並べかえることを考えてみよう。データの並べかえの方法はいろいろあるが、そのうちの 1 つの方法について以下の図で説明する。
双方向矢印で指されているデータ 2 つを比べて大きい方のデータを右側に移動する。矢印上の数の順に比べて交換していく。6 回の比較・(必要であれば) 交換でデータ 4 つを並べ換えることが可能なことがわかる。



このような並べかえの方法を、大きい数が段々と後ろへ移動していくところから泡立ち法 (bubble sort) と呼んでいる。

今、4 つのデータが、

```
int   a[4];
```

で宣言された配列 a の各要素 a[0], a[1], a[2], a[3] に入っているものとする。(注意：a[4] は使えず a[0] から a[3] までの 4 個!)
上の方法を使って 4 個のデータの並べかえをするプログラムは次のように書ける。

```
 1  #include <stdio.h>
 2  main()
 3  {
 4          int     a[4];           /* a[0],a[1],a[2],a[3] */
 5          int     tmp;
 6
 7          scanf("%d",&a[0]);
 8          scanf("%d",&a[1]);
 9          scanf("%d",&a[2]);
10          scanf("%d",&a[3]);
11
12          if (a[0] > a[1]) {
13                  tmp = a[0]; a[0] = a[1]; a[1] = tmp;   /* swap a[0] and a[1] */
14          }
15          if (a[1] > a[2]) {
16                  tmp = a[1]; a[1] = a[2]; a[2] = tmp;   /* swap a[1] and a[2] */
17          }
18          if (a[2] > a[3]) {
19                  tmp = a[2]; a[2] = a[3]; a[3] = tmp;   /* swap a[2] and a[3] */
20          }
21
22          if (a[0] > a[1]) {
23                  tmp = a[0]; a[0] = a[1]; a[1] = tmp;   /* swap a[0] and a[1] */
24          }
25          if (a[1] > a[2]) {
26                  tmp = a[1]; a[1] = a[2]; a[2] = tmp;   /* swap a[1] and a[2] */
27          }
28
29          if (a[0] > a[1]) {
30                  tmp = a[0]; a[0] = a[1]; a[1] = tmp;   /* swap a[0] and a[1] */
31          }
32
33          printf("%d %d %d %d\n",a[0],a[1],a[2],a[3]);
34  }
```

(注意) 変数 i と変数 j の値を入れ換えたい時は、一時的 (テンポラリィ) に値を保持する変数を用意して、

```
tmp = i;
i = j;
j = tmp;
```

などとする。

上の例では 4 個のデータであるが、これが 100 個のデータになると if の数が 4950 回になり、プログラムの行数は 15000 行程度になってしまう。そこで、プログラム中の規則性に注目して for 文を使って書き変えてみよう。

`a[0]` から `a[3]` のうちで最大の要素を最後に移動する作業は

```
    for (j = 0 ; j <= 2 ; j++) {
        if (a[j] > a[j+1]) {
            tmp = a[j]; a[j] = a[j+1]; a[j+1] = tmp;
        }
    }
```

でできる。これで a[3] は決定したので、残りの a[0] から a[2] の最大要素を a[2] に移動する。

```
    for (j = 0 ; j <= 1 ; j++) {
        if (a[j] > a[j+1]) {
            tmp = a[j]; a[j] = a[j+1]; a[j+1] = tmp;
        }
    }
```

これで a[2] も決定したので、残りの a[0] から a[1] の最大要素を a[1] に移動する。

```
    for (j = 0 ; j <= 0 ; j++) {
        if (a[j] > a[j+1]) {
            tmp = a[j]; a[j] = a[j+1]; a[j+1] = tmp;
        }
    }
```

以上の操作は for 文の終了条件 <= 数字　のところだけが違うので、

```
for (i = 2 ; i >= 0 ; i--) {
    for (j = 0 ; j <= i ; j++) {
        if (a[j] > a[j+1]) {
            tmp = a[j]; a[j] = a[j+1]; a[j+1] = tmp;
        }
    }
}
```

とまとめられる。

```
 1  /* 4 つの要素の並べかえ */
 2  #include <stdio.h>
 3  main()
 4  {
 5          int     a[4], tmp, i, j;
 6
 7          for (i = 0 ; i <= 3 ; i++) {                /* 数値の入力 */
 8                  scanf("%d",&a[i]);
 9          }
10
11          for (i = 2 ; i >= 0 ; i--) {                /* 並べかえ */
12                  for (j = 0 ; j <= i ; j++) {
13                          if (a[j] > a[j+1]) {
14                                  tmp = a[j]; a[j] = a[j+1]; a[j+1] = tmp;
15                          }
16                  }
17          }
18
19          for (i = 0 ; i <= 3 ; i++) {                /* 結果の出力 */
20                  printf("%d\n", a[i]);
21          }
22  }
```

[練習 1] データ数を 10 個にする場合はプログラムのどこを変えればよいか?

[練習 2] データ数が変わるたびにプログラムを大きく修正しなくて済むように、データ数 n 個 (n の値は for ループに入る前にセットするものとする) の場合のプログラムはどのようになるか?

[練習 3] データ -1 が入力されるまでデータを読み込み、それを小さい順に並べるにはどうすればよいか？

break 文

今まで見てきた方法では、最初から a[0], a[1], a[2].... が小さい順に並んでいる時や、並べかえ作業の途中で並べかえが終了してしまった時も、比較が続行されるという無駄がある。そこで、「比較・交換作業の途中で一回も交換が行われなくなったら、並べ変えが終了しているのでそれ以上比較・交換作業を行わない」ということをしたい。

このためには、「ある条件が満たされたら for ループから強制的に抜ける」ことをしなければならない。これには、break 文を使う。たとえば、

```
for(i=1; i<=50; i++) {
    if(n*i > 1000) {
      break;
    }
}
```

という例では、n*i の値が 1000 を越えた時点で、for ループから抜けることができる。

もし、以下のように多重ループ中で break 文が使われている場合には、

```
for(i=1; i<=50; i++) {
     for(j=1; j<=20; j++) {
           if(n*i*j > 1000) {
                   break;
           }
     }
}
```

一番内側のループから抜けるだけで、ループ全体から一挙に抜けることはできない。

[練習 4] break 文を使って、前ページのプログラムを「途中で並べかえが終了した時点で比較・交換作業を終了する」ように変えなさい。