

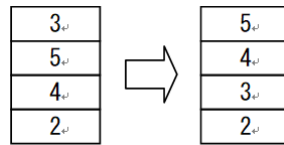
----- 今日の目標 -----

教科書第7章「配列」の7.6 (配列の応用) と7.7 (多次元配列) を学ぶ。 次の問に答えられるようにする。

- 配列上のデータをソートするプログラムが書けるようになる。
 - 多次元 (2次元以上) の配列の書き方を理解し、2次元配列を使ったプログラムを書けるようになる。
-

1. 配列をソートする (教科書7.6 p 201)

「ソート」 ～～ 大きさの順に並べ変えること、日本語だと「整列」。



コンピュータでは、2つの数の大小を比べることしか、できない。「2つの数の比較」を使って、図のような並べ替えをするにはどうするか？

ここでは、前回 (第8回) の4の「最大値」と同じ原理を使ってみる。(教科書p 203)

まず、最初にa[0]の値をa[1]、a[2]、a[3]と比較し、最大値にしよう。(4の最大値を求める原理を見直してみよう)

このとき、a[0]を書き換えるだけでなく、書き換えられる方の要素 (たとえばa[1]) には元のa[0]を入れる。← (前回と違うところ)

つまり、a[0]とa[1]を交換する。これによって、a[1]、a[2]、a[3]と比較し終わった時に、残りのa[1]、a[2]、a[3]には元のデータの最大値を除いた要素 (2番目から4番目まで) が入っていることになる。ここまで、よいだろうか？

次に、残りのa[1]、a[2]、a[3]について、同じようにa[1]の値をa[2]、a[3]と比較して、a[1]、a[2]、a[3]の最大値とする。

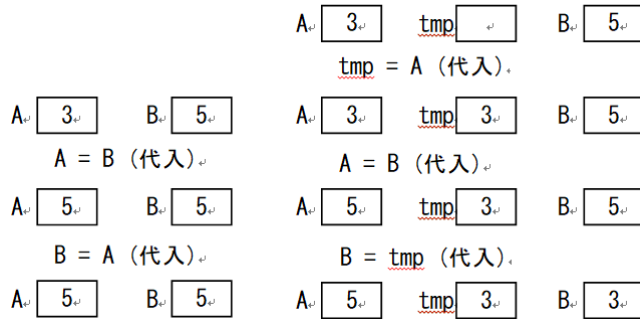
そうすると、a[1]が全体の2番目に大きい数になり、a[2]、a[3]には全体から1番目と2番目に大きい数を除いた残りが入っている。

更に、残りのa[2]、a[3]について、同じことを繰り返して、a[2]を (3番目に大きい数に) 決める。

このようにすれば、a[0]、a[1]、a[2]、a[3] の順番に「残りの中から最大のもの」を選んでゆくことになるので、「整列」することができる。

(注) なお、4の例と違って、最大値を取出すときに (4でいう) maxを書き換えると同時に、上書きする方の (大きかった方の) 数の場所に、もともとmaxに入っていた値を書き戻さなければならない。これは、2つの変数の値を交換する操作になる。(BをAに入れると同時に、AをBに入れる。)

これは、トリックが必要である。もし単純に、A=Bのように代入すると、次にBに入れる (戻す) 値が上書きされるので、無くなってしまっている。



そこで、別の臨時の変数tmpを用意して、上書きする前のAの値を退避しておき、最後にBに戻すときにはtmpから戻す。

```

tmp = A;
A = B;
B = tmp;

```

の3段階の手順を踏む必要がある。プログラムの構造は、およそ下図のような形になるだろう。

```

int a[] = { 3, 5, 4, 2 };
for (int i=0; i<3; i++) {
    for (int j=i; j<4; j++) {
        もし a[i]>a[j] なら
            a[i] と a[j] を入れ替える
    }
}

```

[演習 1] 上記のプログラムの構造がどのように実行されるかを、コンピュータになったつもりで追いかけてみよ。

それぞれのループ本体の先頭（本体部分の開始時点）で、各変数がどのような値になっているかを書いてみよ。

i の値	j の値	a[0] の値	a[1] の値	a[2] の値	a[3] の値
0	0
0	1
0	2
	
	
	
	
	
	
	
	
	

[演習 2] 上記のようなプログラムを完成し、JCPad上で動作を確認せよ

[演習 3] 上記のプログラムは、大きい方が先で、小さい方が後から出てくる順番（下り順、降順）になるようにソートしている。これを改造して、逆方向（上り順、昇順）のソートを作り、JCPad上で動作を確認せよ。

2. 多次元の配列 （教科書 7.7 p 205 ~ ）

利用イメージ： 行列みたいなものが扱いたいとか 1次元配列（ベクトル）がたくさんあるので... とか

次元数は、2次元に限定しないが、まずは2次元から試そう。

[演習4] 教科書 p 205~206を参考にして、Sample10.javaが動作することを確認せよ。

[演習5] 上記、Sample10.java を、下記の様に、一部変更して試せ。

6行目: `test = new int[2][5];` ⇒ `test = new int[2][3];`と変更せよ。

当然、それ以降の処理は、上記の変更によって書き換えなければならない。(自分でどこを書き換えるか考えよ)

ここまでで、思った通りに動くことを確認せよ。

[演習6]

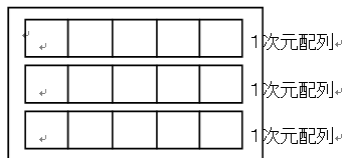
次に、上記に加えてもう1点変更せよ。下から8行目にあるfor文の後System.out.println(...)が2回出てくる。

この2回のprintlnを、1つのfor文で2回繰り返すように、書き換えよ。つまり、for文は入れ子になる。入れ子のfor文が書けることを確認せよ。

一般に、2次元の配列（つまり2次元の行列）は、2重のforループを使うと、すべての要素をスキャンする（なめる）ことができる。これを拡張すると、N次元の配列は、N重のforループですべての要素をスキャンすることができる。

「配列の配列」という見方

2次元の配列は、1次元の配列を複数並べて配列にしたもの、と見ることができる。



「要素配列の長さが揃っていない必要はない」(いびつな配列も作れる) p208~209

⇒ 考え方としては、上記の「配列の配列」で説明が付くが、やめておいたほうがいい。

(脱線話: 「参照」との関連)

多次元配列の宣言の仕方 と 初期化

1次元の場合

- ① `int[] hairetsu;`
`hairetsu = new int[5];`
- ② `int[] hairetsu = new int[5];`
- ③ `int[] hairetsu = { 1, 3, 5, 7, 9 };`

2次元の場合

- `int[][] hairetsu2;`
`hairetsu2 = new int[2][5];`
- `int[][] hairetsu2 = new int[2][5];`
- `int[][] hairetsu2 = {{ 1, 3, 5, 7, 9 },`
`{ 2, 4, 6, 8, 10 }};`

[演習7] 次の、行列とベクトルの積の計算を考える。

$$\begin{pmatrix} y1 \\ y2 \end{pmatrix} = \begin{pmatrix} A11 & A12 \\ A21 & A22 \end{pmatrix} \times \begin{pmatrix} x1 \\ x2 \end{pmatrix}$$

ベクトルx = (x 1, x 2) を入力としたとき、y = (y 1, y 2) を求めるプログラムを作りたい。行列Aはたとえば

$$A = \begin{pmatrix} \cos\left(\frac{\pi}{6}\right) & -\sin\left(\frac{\pi}{6}\right) \\ \sin\left(\frac{\pi}{6}\right) & \cos\left(\frac{\pi}{6}\right) \end{pmatrix} = \begin{pmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{pmatrix}$$

としてみよう。

プログラミングの練習なので、行列Aは2次元配列として作り（添え字が11,12, 21, 22ではなくて0から始まる、つまりA[0][0], A[0][1], A[1][0], A[1][1] となることに注意）。

同様に、ベクトルxとyも1次元配列 $x = (x[0], x[1])$ 、 $y = (y[0], y[1])$ として作れ。forループで繰り返して計算すること。forループは二重になるはずである。テスト入力として次の4つのベクトル $p = (0, 0)$ 、 $q = (0.5, 0.866)$ 、 $r = (-0.366, 1.366)$ 、 $s = (-0.866, 0.5)$ を与え、それぞれの結果のベクトル p' 、 q' 、 r' 、 s' を表す点を求め、元の p 、 q 、 r 、 s と得られた p' 、 q' 、 r' 、 s' を（紙上で）図に描いてみよ。

[演習8] 次の表は、月別の一世帯当たりのアイスクリーム支出金額と、平均気温を描いたものである。

支出欄と気温欄のデータを用いて初期化した2次元の配列 `ice[2][12]` を作れ。

月	1	2	3	4	5	6	7	8	9	10	11	12
気温	10.6	12.2	14.9	20.3	25.2	26.3	29.7	31.6	27.7	22.6	15.5	13.8
支出	464.0	397.0	493.0	617.0	890.0	883.0	1292.0	1387.0	843.0	621.0	459.0	561.0

気温出典 2016年 月別平均気温 気象庁

http://www.data.jma.go.jp/obd/stats/etrn/view/monthly_s3.php?prec_no=44&block_no=47662&view=a2

支出出典 2016年 一世帯当たりアイスクリーム支出金額 一般社団法人日本アイスクリーム協会

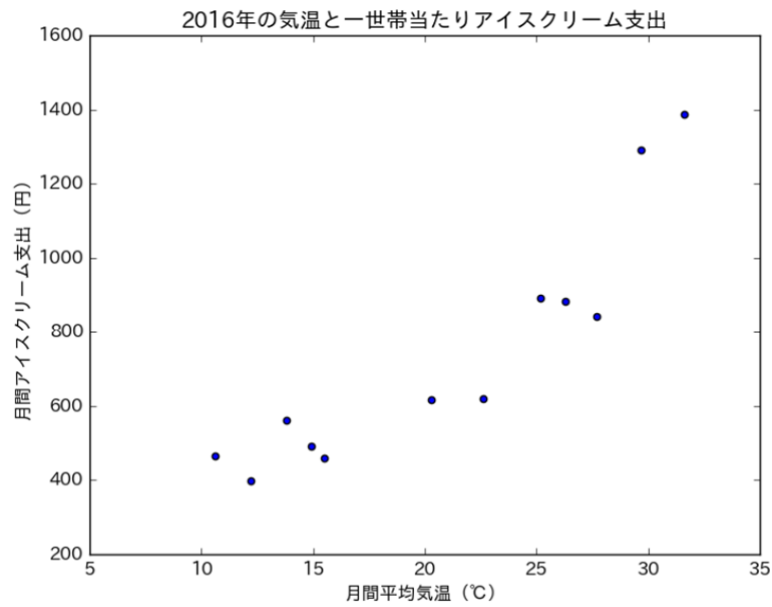
<https://www.icecream.or.jp/data/expenditures.html>

この配列を用いて、支出の平均値と、気温の平均値を計算せよ。

[演習9] 演習8の気温をx軸、支出をy軸としてグラフを描くと、右上がりに集まっていることが分かる。これらの点を近似する直線の傾きbと切片aは、次の式で求められる（最小二乗法、回帰直線）。データから傾きbと切片aの値を計算し、グラフに（手で）書込んでみよ。但し、 \bar{x} 、 \bar{y} はそれぞれ x_i 、 y_i の平均とする。

$$b = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sum x_i^2 - n \bar{x}^2}$$

$$a = \bar{y} - b \bar{x}$$



$$y = 40.70x - 107.1$$

(ヒント) 繰り返し自体は単純であるが、いろいろなもの(式)についての総和を求める必要がある。それぞれを別々に計算するように、問題を整理するとよい。たとえば、(以下、各自でよく検算した上で用いよ)

$$sx = \sum x_i \text{の総和} = n * \bar{x}, \quad sy = \sum y_i \text{の総和} = n * \bar{y}$$

$$sxy = \sum (x_i * y_i) \text{の総和}$$

$$s2x = \sum x_i^2 \text{の総和}$$

をそれぞれ別のループで計算しておき、次の式を求めればよからう。

$$b = (sxy - ((sx * sy) / n)) / (s2x - ((sx * sx) / n))$$

$$a = (sy / n) - (b * (sx / n))$$