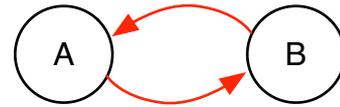


■ 補助資料

□ 二つの変数での値の入れ替え

ソート処理では良く「二つの変数の値を入れ替える」操作が必要になります。処理としては右図のようなものでしょうか。

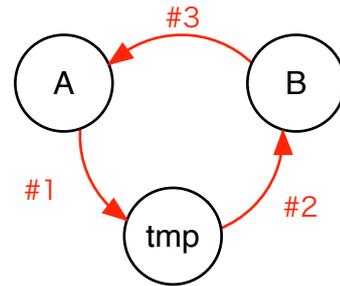


しかし C のような手続き型プログラミング言語の記述として、これを右のように書いてしまうと、実行結果としては a, b 二つとも 200 (つまり「元の b の値」) になってしまいます。

トレースしてみれば納得できるでしょう。代入処理は同時に行うことが出来ず、順に実行すると a の値が (上書きによって) 失われてしまうからです。

```
a = 100;
b = 200;
a = b;
b = a;
```

そこで一般的には右のようなアイデアに基づいて、処理を行います。赤色でつけた番号 (#1, #2, #3) の順に処理を行います。つまり a の値をいったん別の場所に退避させてから、上書きするのです。



コードで書くと右のようになります。

入れ替える処理としては常套手段 (定石) ですので、覚えておくと良いでしょう。

```
tmp = a;
a = b;
b = tmp;
```

□ Bingo プログラムの動作確認

動作を確認するためのシェル・スクリプト bingo_check.sh を教材ページに置きました。ダウンロードして下さい。

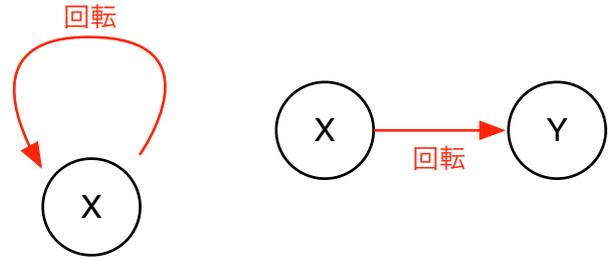
これを右のようにして実行すると、自動的に Bingo になるデータの組み合わせを 8 つ、Not Bingo になる組み合わせを 3 つ与えてくれるので、その結果を確認することができます。あなたが作ったプログラムの判定が、その通りになることを確認してください。実行例を右に示しておきます。

(あなたのプログラムをコンパイルした実行ファイルの名前が a.out でなく別の名前 (たとえば bingo) だとしたら、右の実行例の a.out の部分をそれに合わせて下さいね。)

```
$ bash bingo_check.sh a.out
////////////////////
===== 1 2 3 0 0
 1  2  3
 4  5  6
 7  8  9
-- -- --
. . .
4  5  6
7  8  9
-- -- --
Bingo
===== 0 4 5 6 0
 1  2  3
 4  5  6
..... (続く) .....
===== 2 1 6 4 9
 1  2  3
 4  5  6
 7  8  9
-- -- --
. . 3
. 5 .
7 8 .
-- -- --
Not Bingo
$
```

□ 配列要素の回転

配列要素を回転させる方法はいろいろあります。たった一つの二次元配列を使って、要素を入れ替えることでも実現できないわけではありません（ややこしい日本語になってごめんなさいね）。しかしそうではなく、配列変数を二つ用意して、X から Y に移していくほうが、プログラムとしてはシンプルに書けると思います。（課題 3. に相当）

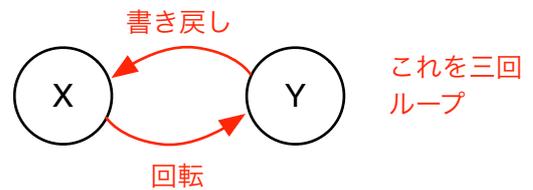


今回の課題 4. は、次々と回転させていくのですが、これを 3 回転させるなら配列変数を 3 つ追加して、つまり右のようにして実現することもできるでしょう。



しかし今回はこうしたことをせず、右図のように処理してください。つまり、

- ・配列 X から配列 Y に回転させながら移して
 - ・配列 Y の内容を X に（そのまま）移す
- 作業を 3 回繰り返すのです。



こうしないと、3 回回転させるために 4 つの配列を用意することになり、それに引きずられる形で、3 倍のコードを書き並べることになります。それを避けたいのです。