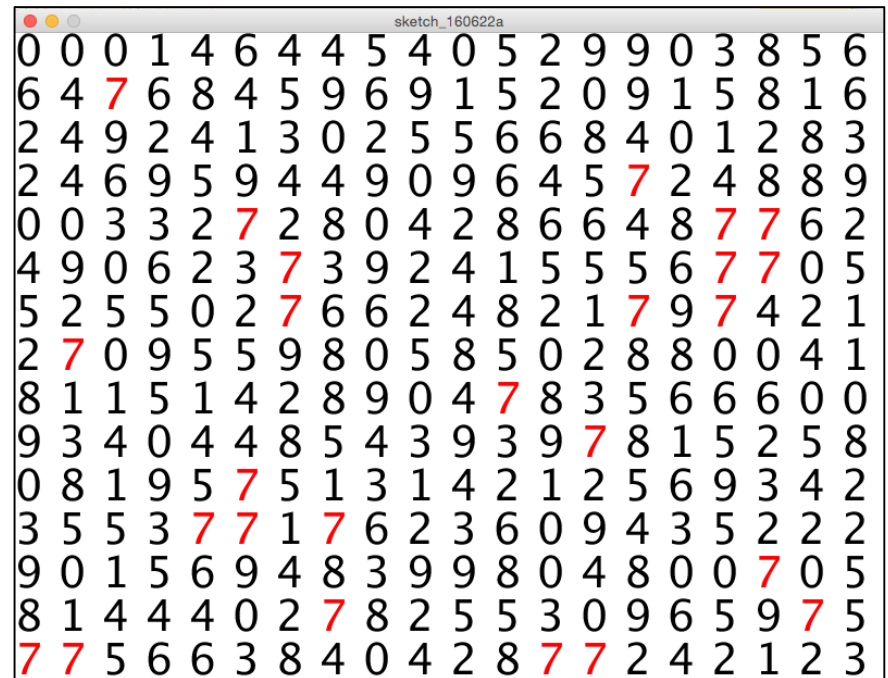


# プログラミング演習I (第7回) 課題

## • 基本① スケッチ名: random\_numbers

- 800x600のウィンドウ内を0~9の数字で埋め尽くし、その数字をどんどん切り替えよ(drawで毎フレーム毎に書き換える。1文字毎に乱数で数値を決める等する)。その数字が7の時に赤色で、それ以外の数字の時は黒色で描画するようにせよ
- 横に20個、縦に15個ならべ、またできる限り文字も大きくせよ(textSize(40)くらい?)



0	0	0	1	4	6	4	4	5	4	0	5	2	9	9	0	3	8	5	6
6	4	7	6	8	4	5	9	6	9	1	5	2	0	9	1	5	8	1	6
2	4	9	2	4	1	3	0	2	5	5	6	6	8	4	0	1	2	8	3
2	4	6	9	5	9	4	4	9	0	9	6	4	5	7	2	4	8	8	9
0	0	3	3	2	7	2	8	0	4	2	8	6	6	4	8	7	7	6	2
4	9	0	6	2	3	7	3	9	2	4	1	5	5	5	6	7	7	0	5
5	2	5	5	0	2	7	6	6	2	4	8	2	1	7	9	7	4	2	1
2	7	0	9	5	5	9	8	0	5	8	5	0	2	8	8	0	0	4	1
8	1	1	5	1	4	2	8	9	0	4	7	8	3	5	6	6	6	0	0
9	3	4	0	4	4	8	5	4	3	9	3	9	7	8	1	5	2	5	8
0	8	1	9	5	7	5	1	3	1	4	2	1	2	5	6	9	3	4	2
3	5	5	3	7	7	1	7	6	2	3	6	0	9	4	3	5	2	2	2
9	0	1	5	6	9	4	8	3	9	9	8	0	4	8	0	0	7	0	5
8	1	4	4	4	0	2	7	8	2	5	5	3	0	9	6	5	9	7	5
7	7	5	6	6	3	8	4	0	4	2	8	7	7	2	4	2	1	2	3

# プログラミング演習I (第7回) 課題

## • 基本② スケッチ名: perfect\_number

- 先週の約数を表示するプログラムを改良し、2から1億までのすべての完全数を求めましょう。なお、何番目の完全数かも表示せよ
  - 余力のある人は500京までのすべての完全数を求めてみましょう。intではなくlongを使う。
  - longで見つかる最大の完全数は2305843008139952128。
  - 鈴木先生が話していた方法を利用すると30秒程度で求まるはず
- 完全数とは、ある数の正の約数の和がその数自体になるもの。
  - $6 = 1 + 2 + 3$
  - $28 = 1 + 2 + 4 + 7 + 14$  など

1億までの完全数は...

1番目: 6

2番目: 28

3番目: 496

:

[step1]

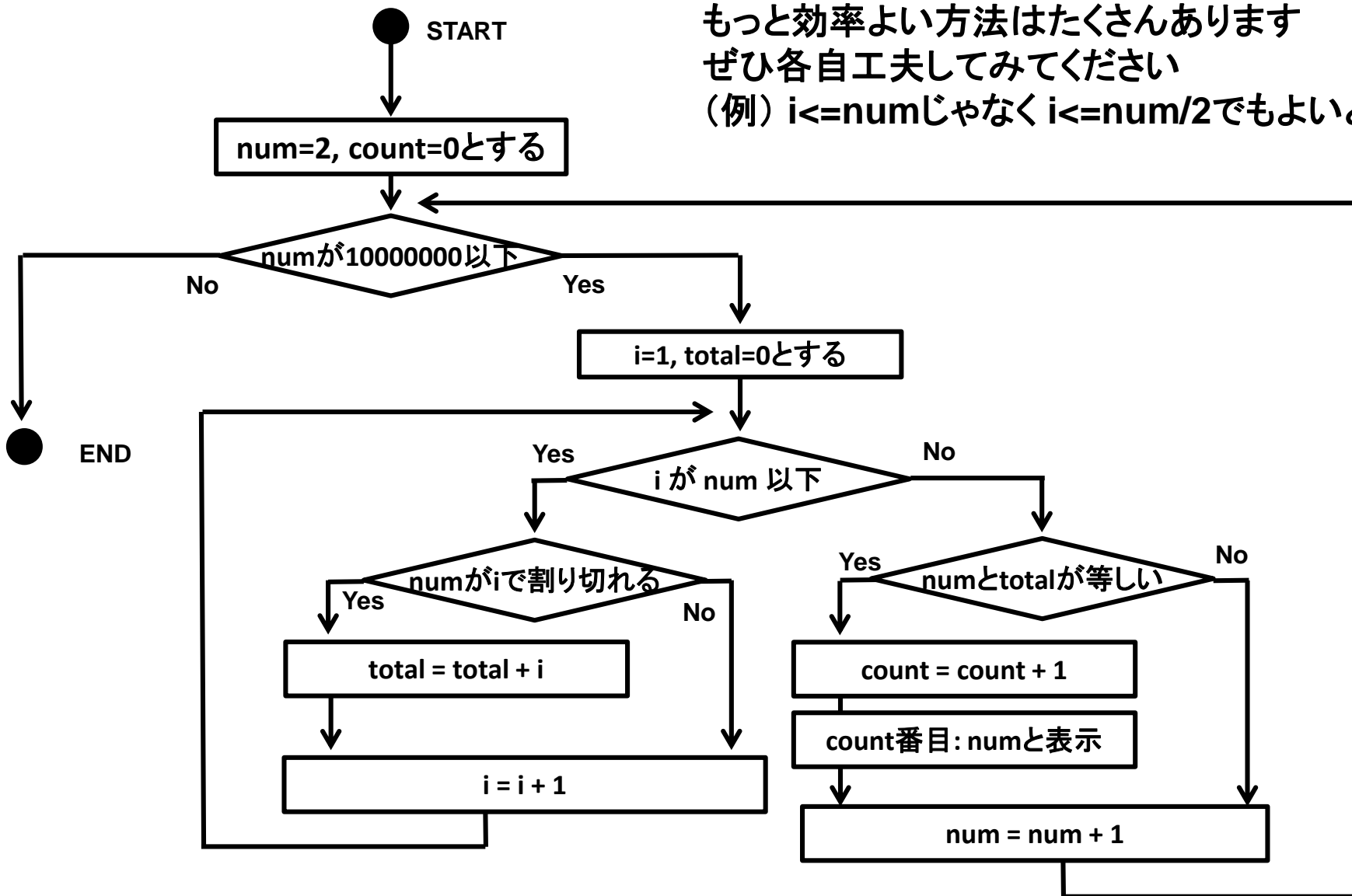
ある数字について、約数をすべて求め、その約数の和がその数字と一致するかどうか判定しよう。

[step2]

ある数字を2~1億まで変化させよう

# 完全数を求めるフローチャート

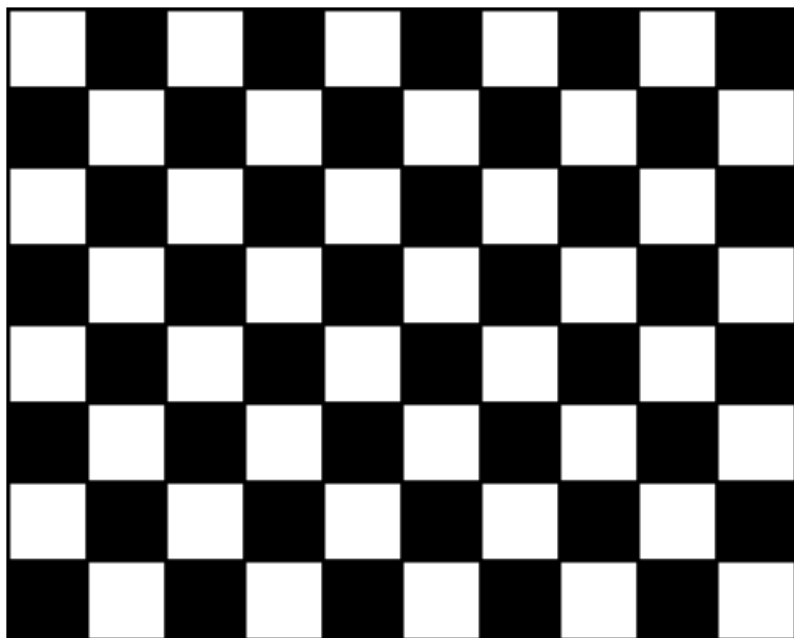
もっと効率よい方法はたくさんあります  
ぜひ各自工夫してみてください  
(例)  $i \leq \text{num}$  じゃなく  $i \leq \text{num}/2$  でもよいとか



# プログラミング演習I (第7回) 課題

## • 基本③ スケッチ名 : chess

- 白と黒の四角形をチェス盤状に並べるプログラムを作ってください。
- ウィンドウのサイズを600x500とし、1つのマスは50x50とします。
- 【左上が白】のマスになるようにしてください。



[step1]

まずは2重の繰り返し文を使って、四角形を敷き詰めてみよう。

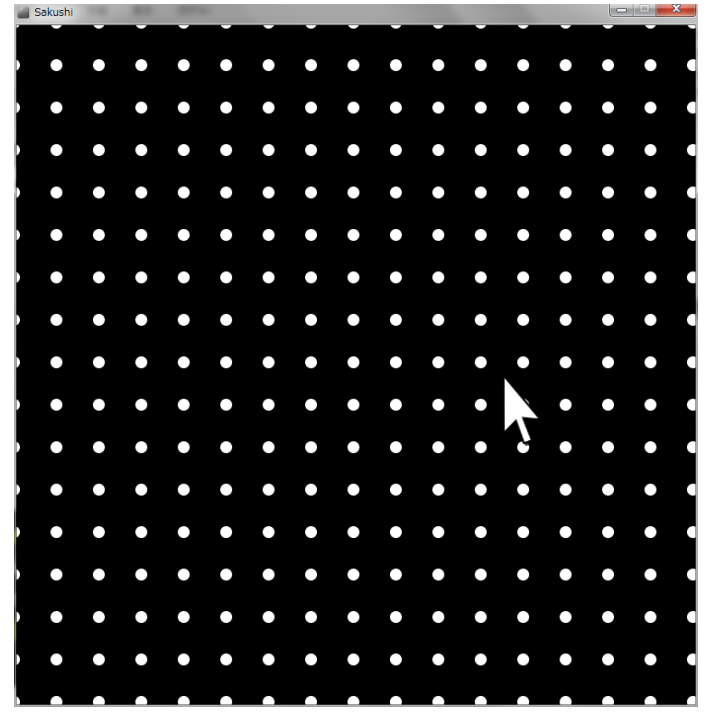
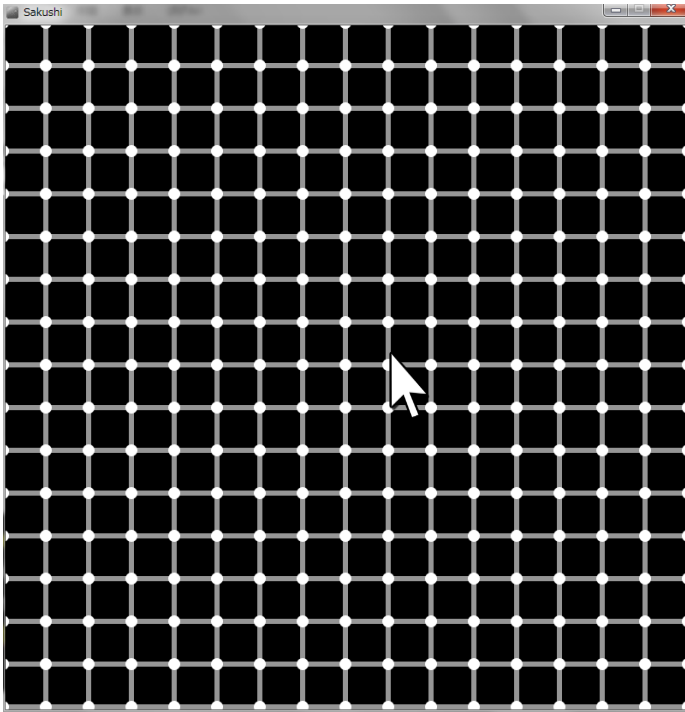
[step2]

if文の分岐によって白または黒で塗り潰されるようにしてみよう。各マス目が白または黒になる条件とはなんだろうか？

# プログラミング演習I (第7回) 課題

## • 発展① スケッチ名 : kirameki

- きらめき格子錯視を作成せよ(黒背景, 灰色の格子, 枠無し白丸)
- ただし、格子の灰色の線を、マウスクリックによって表示したり, 非表示にしたりと切り替えることが出来るようにせよ。
- <http://www.kecl.ntt.co.jp/IllusionForum/v/scintillatingGrid/ja/index.html>

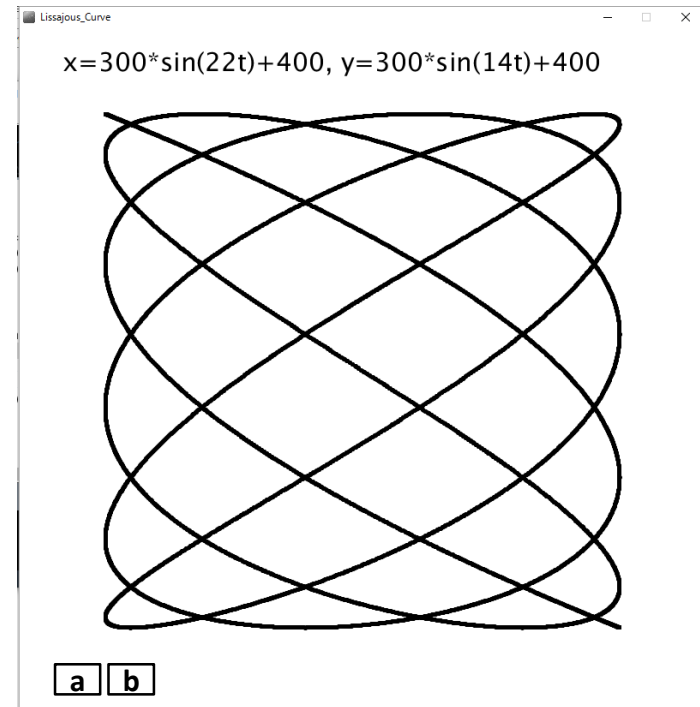


# プログラミング演習I (第7回) 課題

## • 発展課題② リサーチユカーブ: curve

–  $x$  と  $y$  の座標が  $t$  によって変化する下記の数式の計算結果の座標をプロットせよ. ただし, draw で一度に描画せよ. またウィンドウサイズは  $800 \times 800$  とせよ.  $t$  の値は  $0$  から  $2\pi$  まで変化するものとする

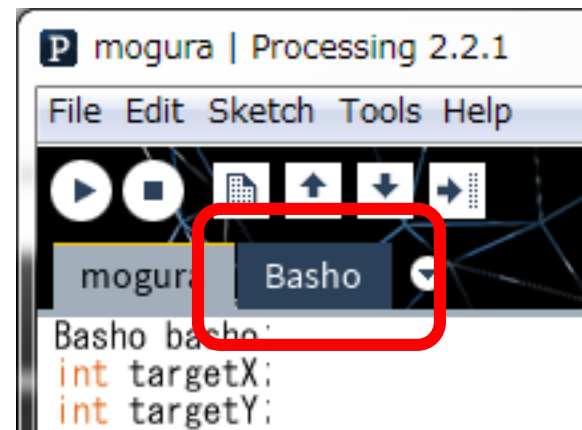
- $x = 300 \sin(at) + 400$
- $y = 300 \sin(bt) + 400$
- $a$  と  $b$  の値は整数とし, 左下の  $a$  ボタンクリックのたびに  $a$  の値,  $b$  ボタンクリックのたびに  $b$  の値をランダムに決定せよ  
また数式も表示せよ



# Bashoの利用方法

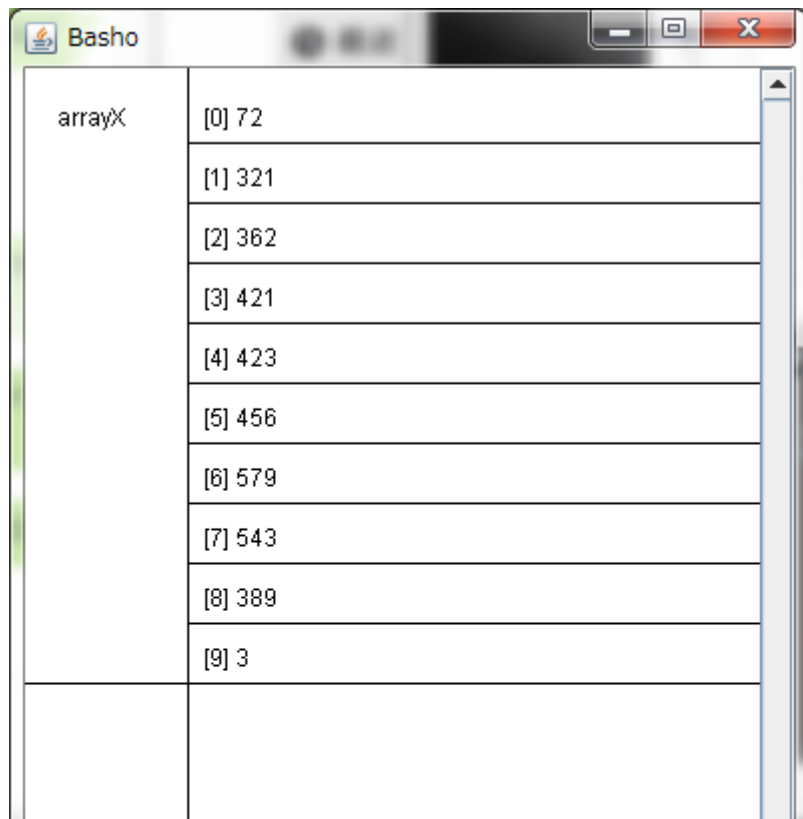
- まずは、下記URLからBashoをダウンロード
  - <https://github.com/kvvzr/Basho>
  - Basho.pdeを抜き出そう！
- 何かプログラムを書いて、そのプログラムの隣に(タブの部分に)Basho.pdeをドラッグアンドドロップしよう！
- おまじない(黄色の行のもの)を2行書こう
- プログラムを実行したら別ウィンドウで変数の状態が表示される！

```
Basho basho;  
void setup(){  
  size( xxx, xxx );  
  basho = new Basho( this );  
}
```



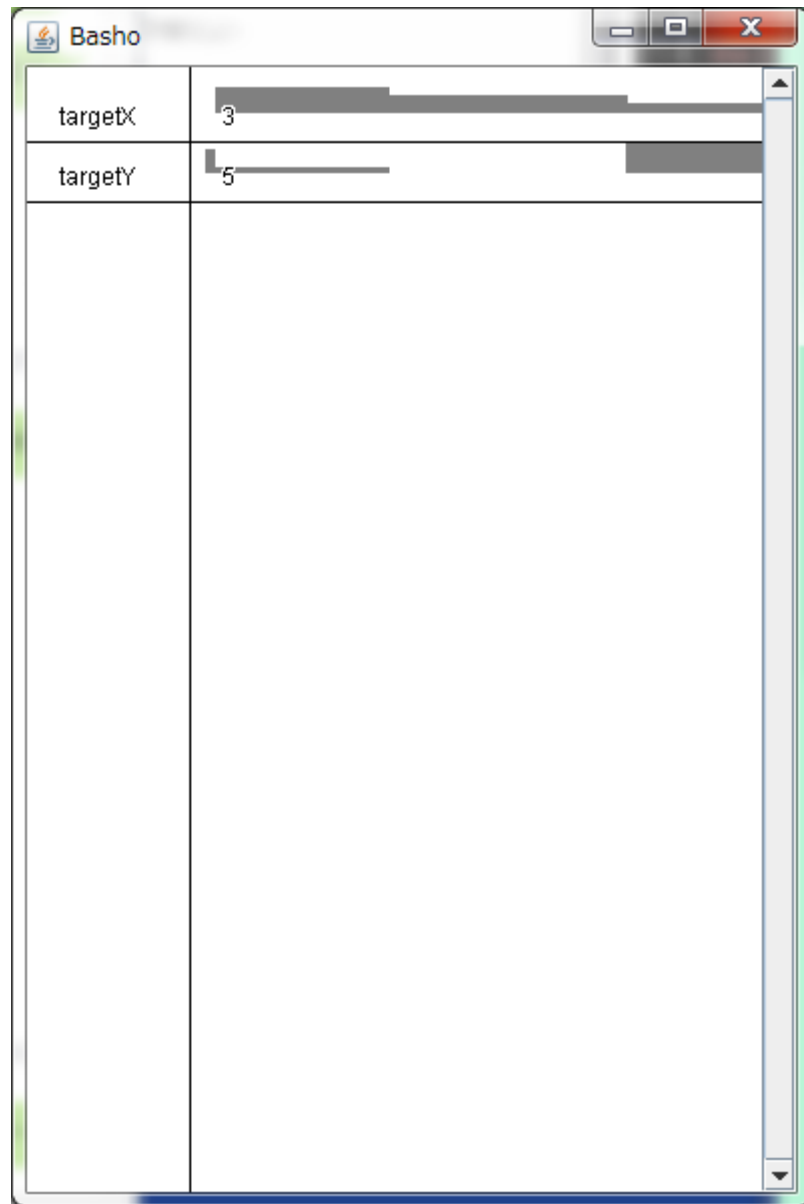
# Bashoでどう表示されるか？

- 配列ではBashoがあると理解が進むのでぜひ利用方法を覚えておこう！



A screenshot of a Basho window titled "Basho". The window displays a table with a single column labeled "arrayX". The table contains 10 rows, each representing an element of the array with its index in brackets followed by the value. The values are: [0] 72, [1] 321, [2] 362, [3] 421, [4] 423, [5] 456, [6] 579, [7] 543, [8] 389, and [9] 3.

arrayX
[0] 72
[1] 321
[2] 362
[3] 421
[4] 423
[5] 456
[6] 579
[7] 543
[8] 389
[9] 3



A screenshot of a Basho window titled "Basho". The window displays a table with two columns. The first column contains the labels "targetX" and "targetY". The second column contains the values "3" and "5" respectively. The rest of the table is empty.

targetX	3
targetY	5