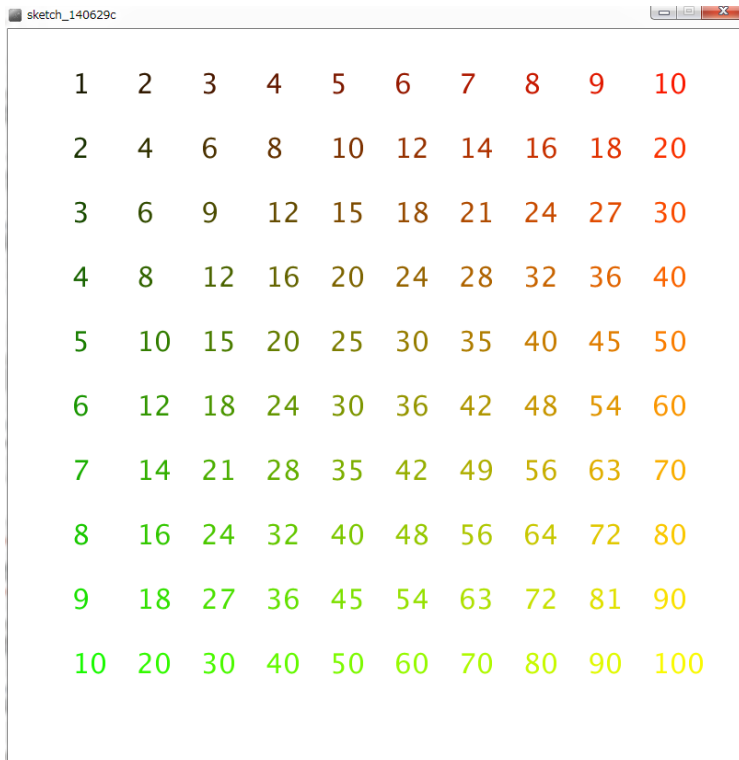


プログラミング演習I (第7回) 課題

• 基本① スケッチ名: table99

- 九九の表を十の段まで書くプログラムを作成してください。
- ただし色は左上を黒色、右上を赤色、左下を青色、右下を黄色とするようにせよ。



1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

[step1]

まずは2重のwhile文(またはfor文)を使って、2つの変数の値を変化させて九九の表を作ってみよう。文字が思い通りの位置に表示されるようにするにはどうしたらいいだろうか？

[step2]

色をどのように変化させるか考えよう

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
 - 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

6は完全数です
28は完全数です
120は完全数です
:

[step1]

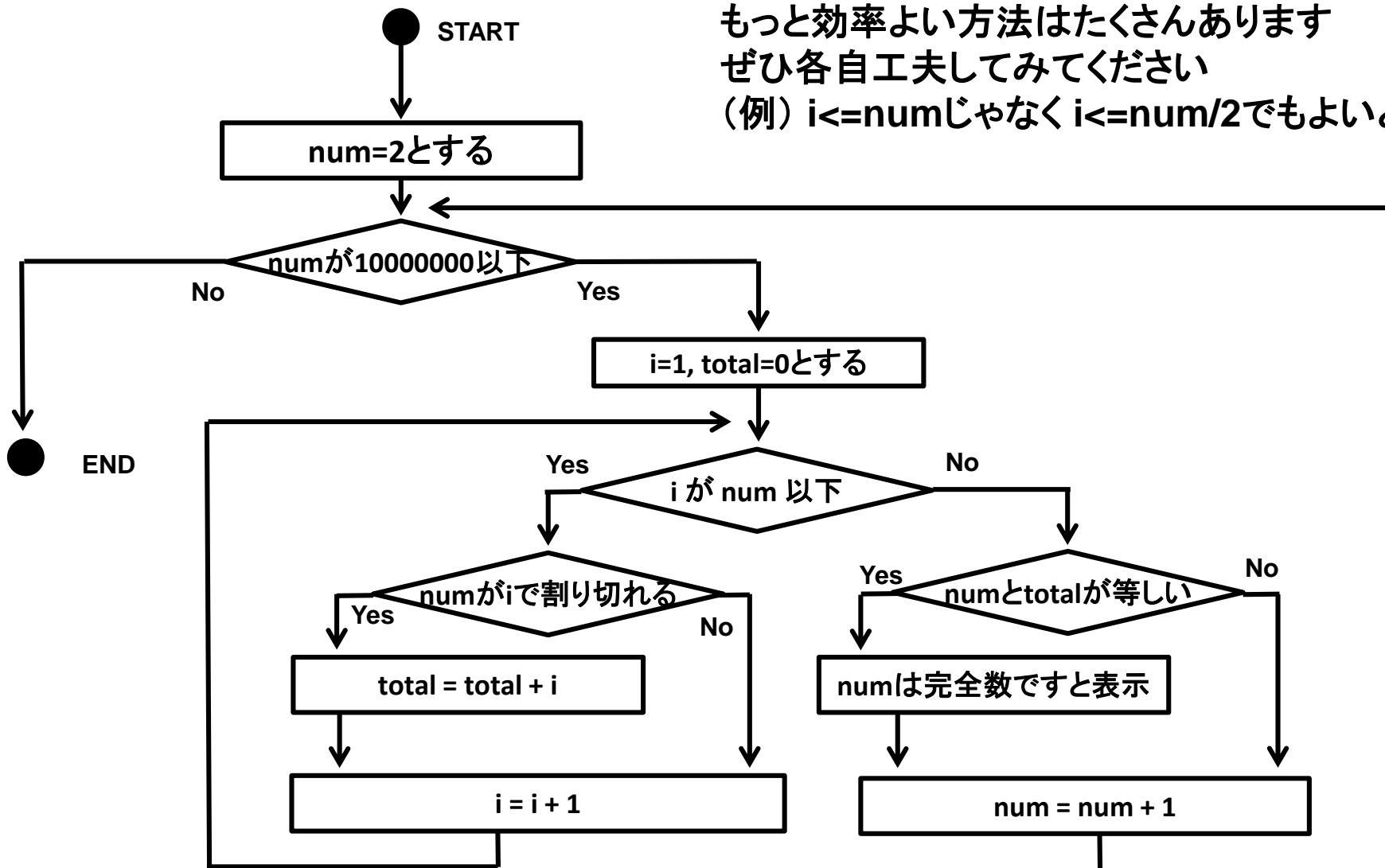
ある数字について、約数をすべて求め、その約数の和がその数字と一致するかどうか判定しよう。

[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

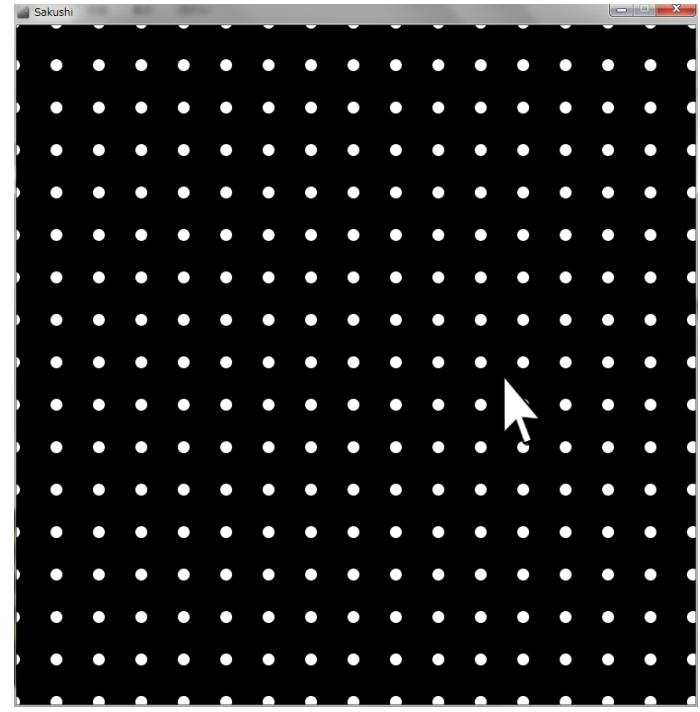
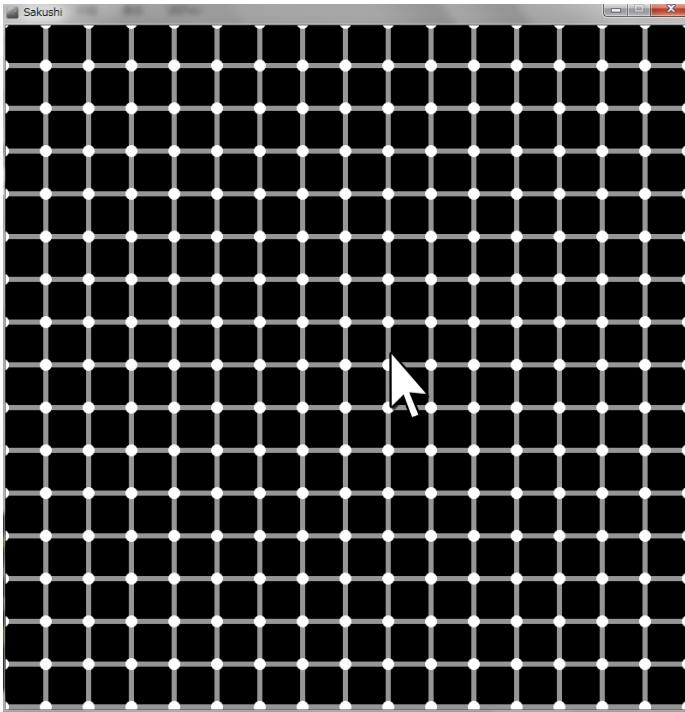
もっと効率よい方法はたくさんあります
ぜひ各自工夫してみてください
(例) $i \leq \text{num}$ じゃなく $i \leq \text{num}/2$ でもよいとか



プログラミング演習I (第7回) 課題

• 発展② スケッチ名 : kirameki

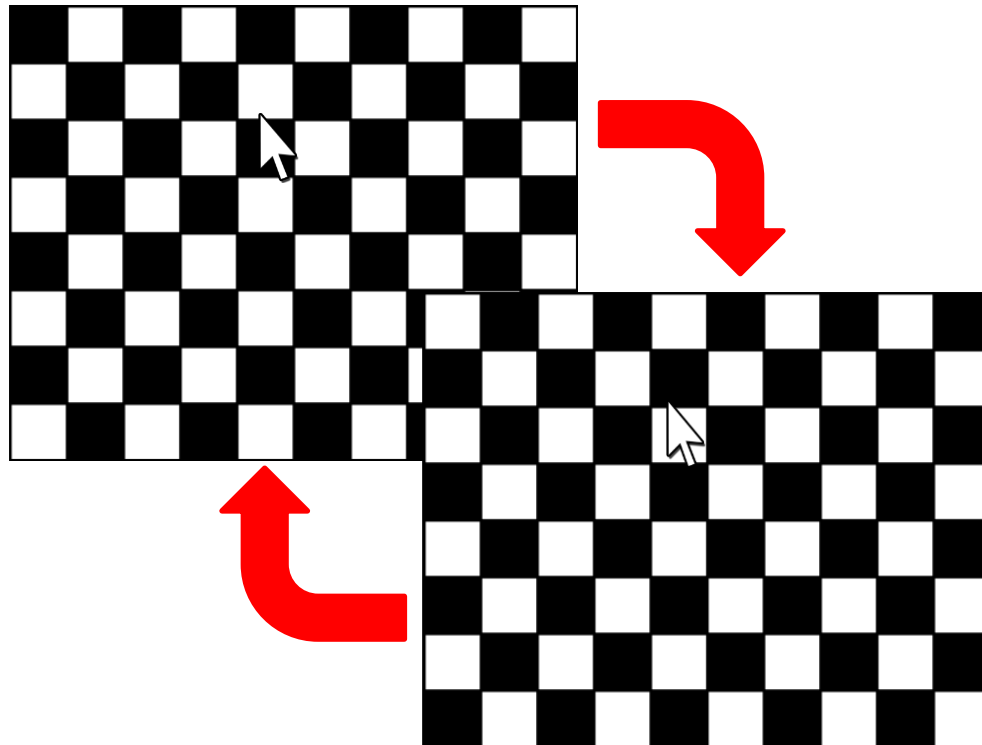
- きらめき格子錯視を作成せよ(黒背景, 灰色の格子, 枠無し白丸)
- ただし、格子の灰色の線を、マウスクリックによって表示したり, 非表示にしたりと切り替えることが出来るようにせよ。
- <http://www.kecl.ntt.co.jp/IllusionForum/v/scintillatingGrid/ja/index.html>



プログラミング演習I (第7回) 課題

• 発展① 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500x400とし、1つのマスは50x50とします。
- 【左上が黒】のマスになるようにし、クリックすると【左上が白】のマスになるようにせよ。また、クリックの度に入れ替わるようにせよ。



[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？

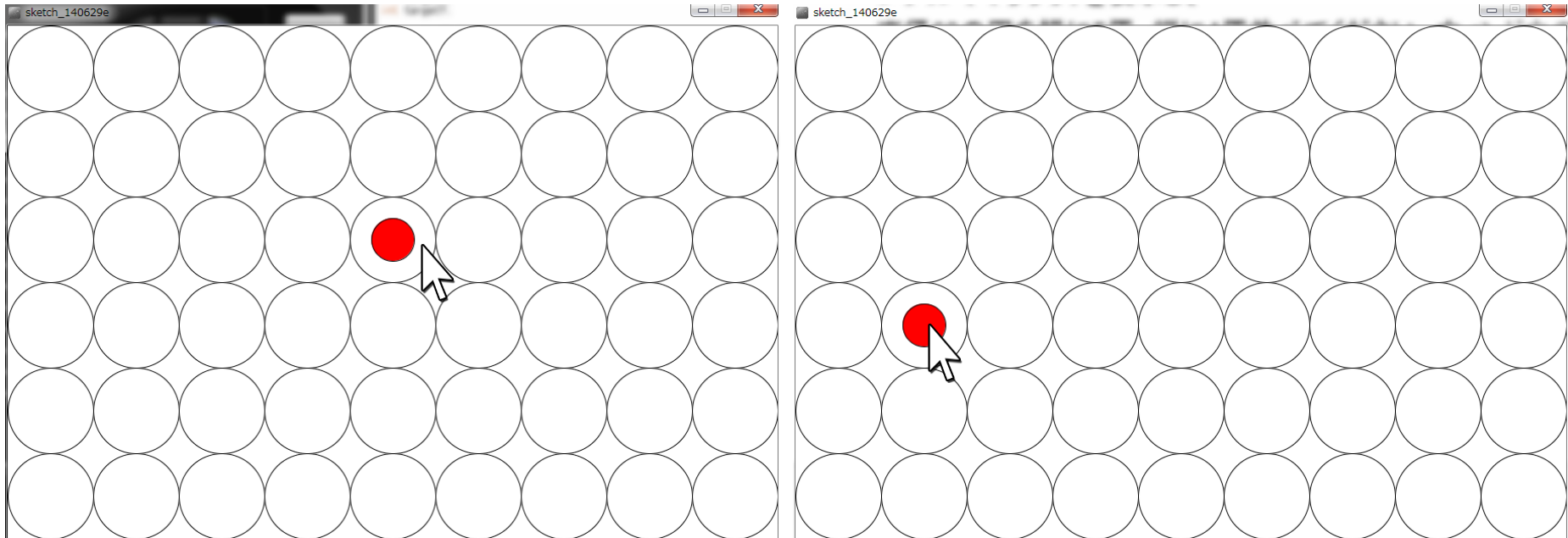
[step3]

マウスクリックによって切り替えよう

プログラミング演習I (第7回) 課題

• 発展② スケッチ名 : mogura

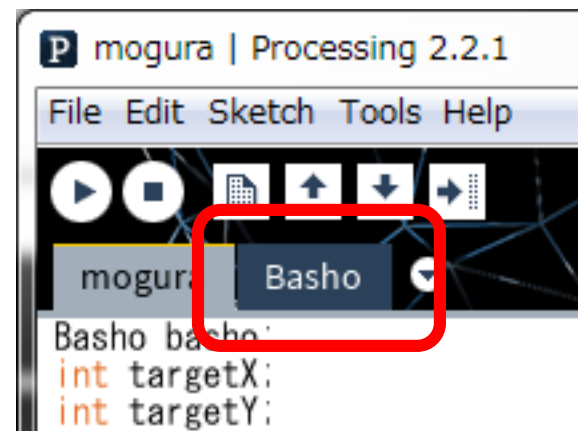
- 画面内を円で敷き詰め、ランダムな円の中に小さな赤い丸(か適当なキャラクタ)を表示し、その赤い丸が含まれる白丸内にマウスをもっていくと、赤丸が別の任意の場所に移動するようにせよ。
- 直径100の円を横に9個、縦に6個並べてください。ウィンドウのサイズは900x600とします。また、円が画面内にぴったり収まるようにしてください。また、Bashoを利用して現在の変数の状態を示せ。



Bashoの利用方法

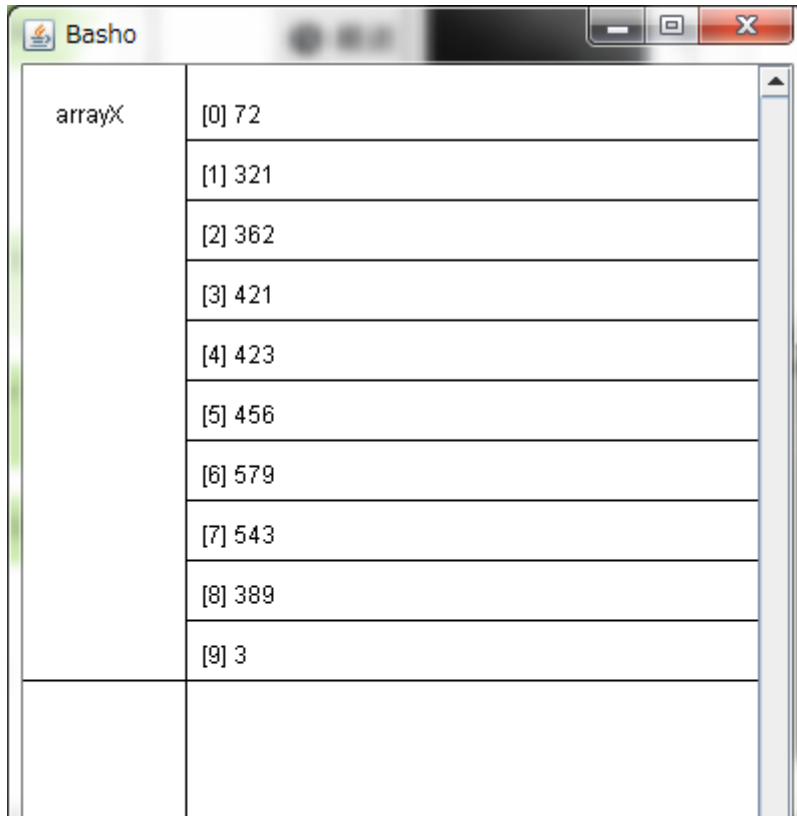
- まずは、下記URLからBashoをダウンロード
 - <https://github.com/kvvzr/Basho>
 - Basho.pdeを抜き出そう！
- 何かプログラムを書いて、そのプログラムの隣に(タブの部分に)Basho.pdeをドラッグアンドドロップしよう！
- おまじない(黄色の行のもの)を2行書こう
- プログラムを実行したら別ウィンドウで変数の状態が表示される！

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



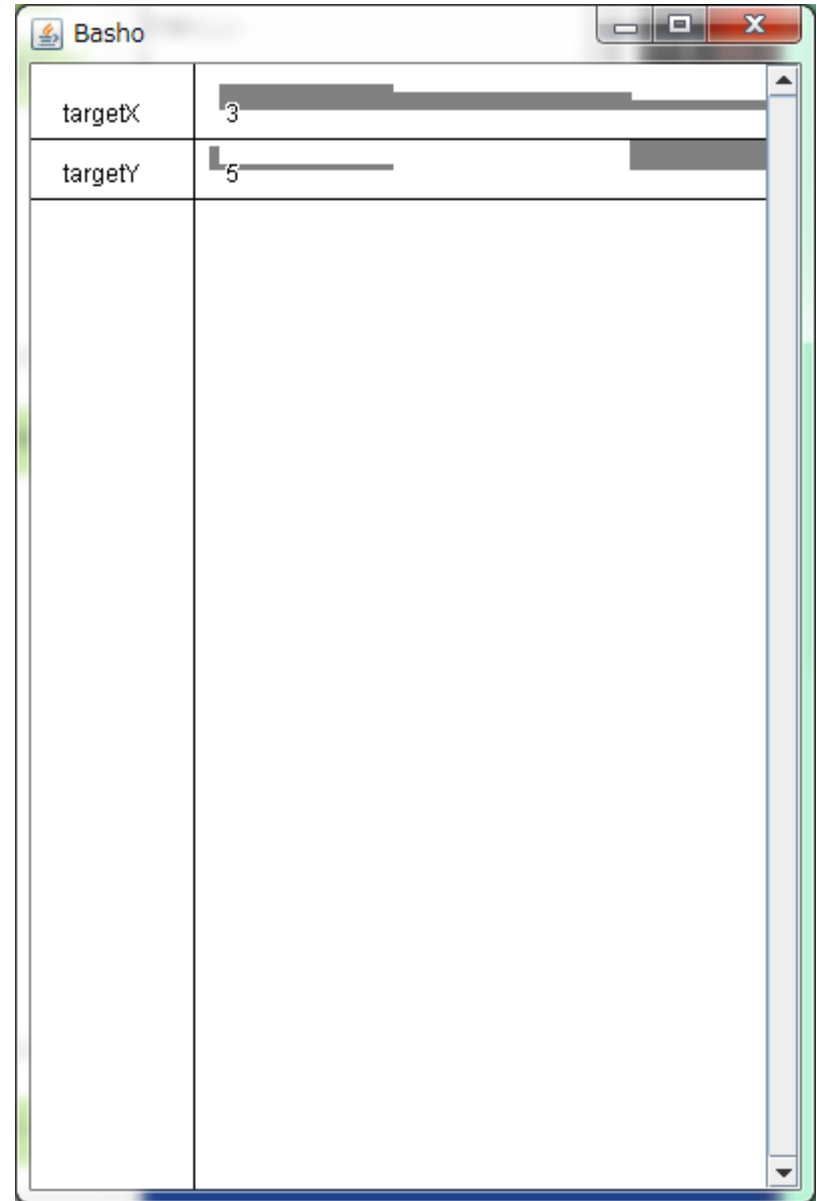
Bashoでどう表示されるか？

- 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 two columns. The left column is labeled "arrayX" and the right column contains the following values:

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



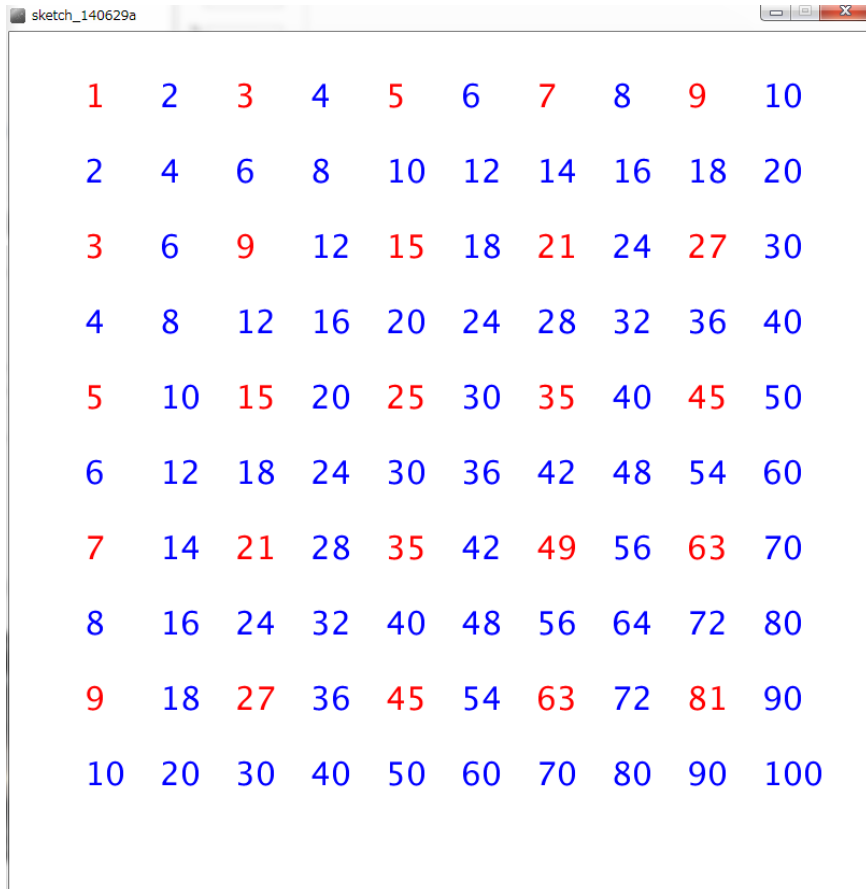
A screenshot of a Basho window titled "Basho". The window displays a table with two columns. The left column contains the labels "targetX" and "targetY", and the right column contains the values "3" and "5" respectively.

targetX	3
targetY	5

プログラミング演習I (第7回) 課題

• 基本① スケッチ名: table99

- 九九の表を十の段まで書くプログラムを作成してください。
- ただし【奇数】を赤文字に、【偶数】を青文字にしてください。



sketch_140629a

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

[step1]

まずは2重のwhile文(またはfor文)を使って、2つの変数の値を変化させて九九の表を作ってみよう。文字が思い通りの位置に表示されるようにするにはどうしたらいいだろうか？

[step2]

if文を使って各数値が偶数かどうかを調べ、条件分岐によって塗り色が変わるようにしてみよう。

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
 - 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

6は完全数です
28は完全数です
120は完全数です
:

[step1]

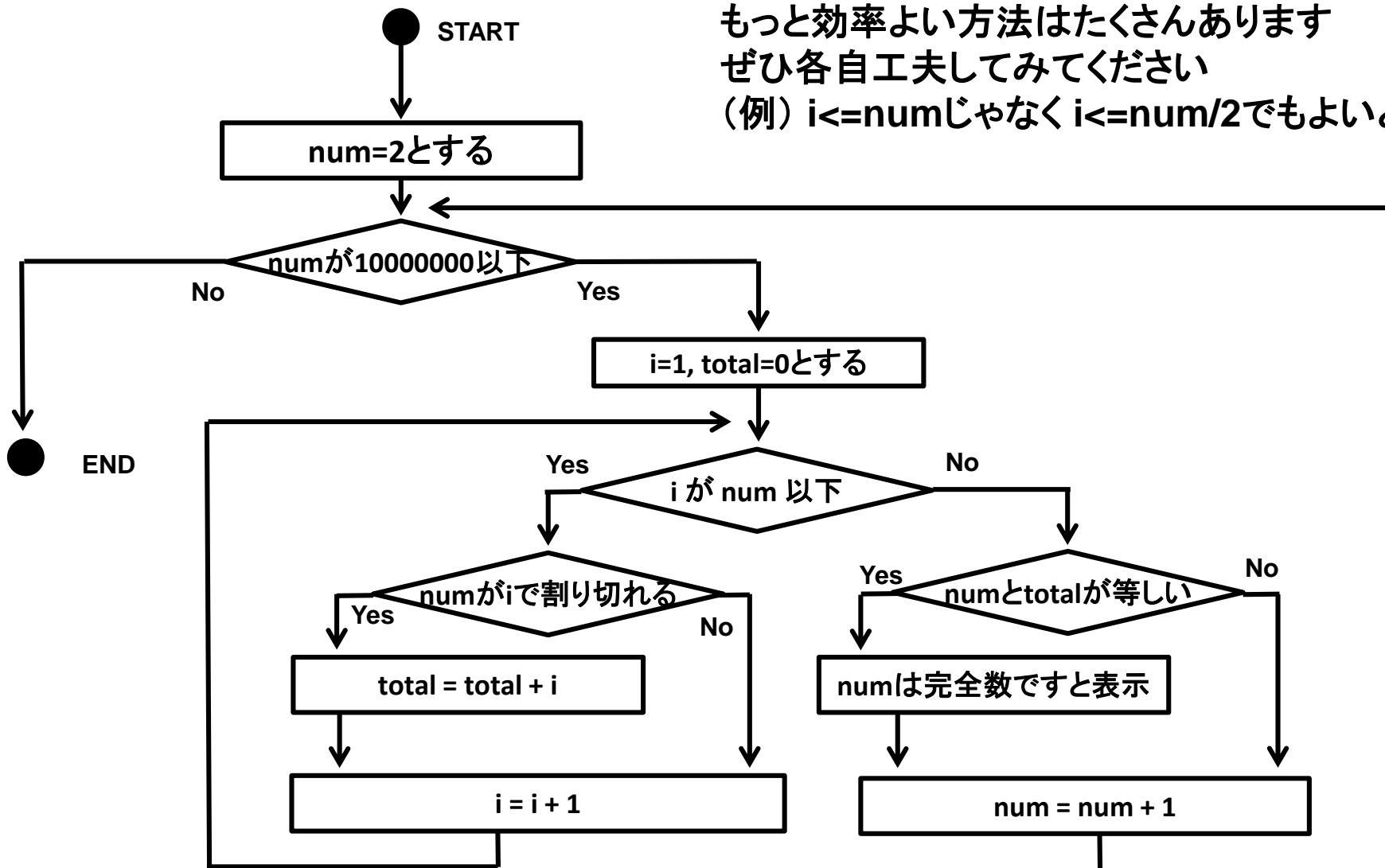
ある数字について、約数をすべて求め、その約数の和がその数字と一致するかどうか判定しよう。

[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

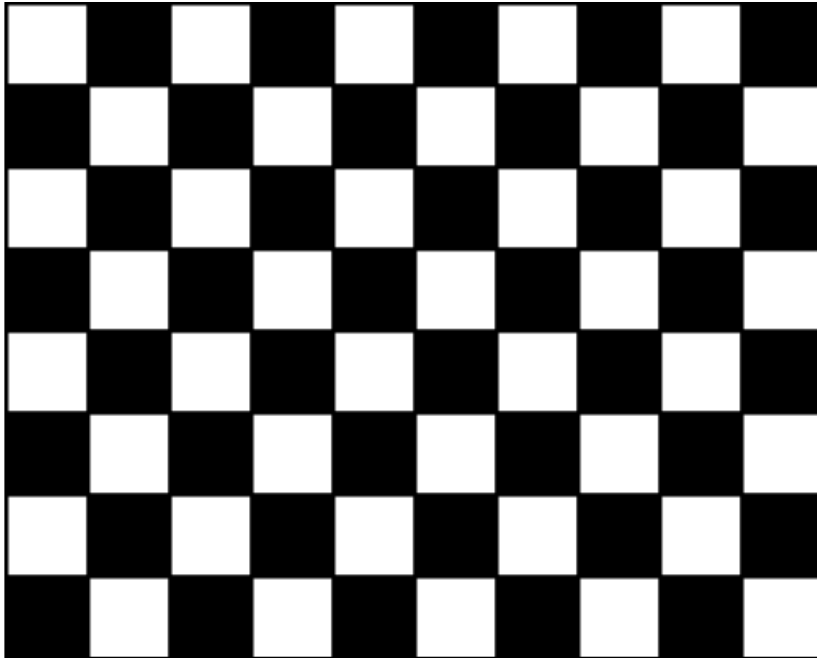
もっと効率よい方法はたくさんあります
ぜひ各自工夫してみてください
(例) $i \leq \text{num}$ じゃなく $i \leq \text{num}/2$ でもよいとか



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500x400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]

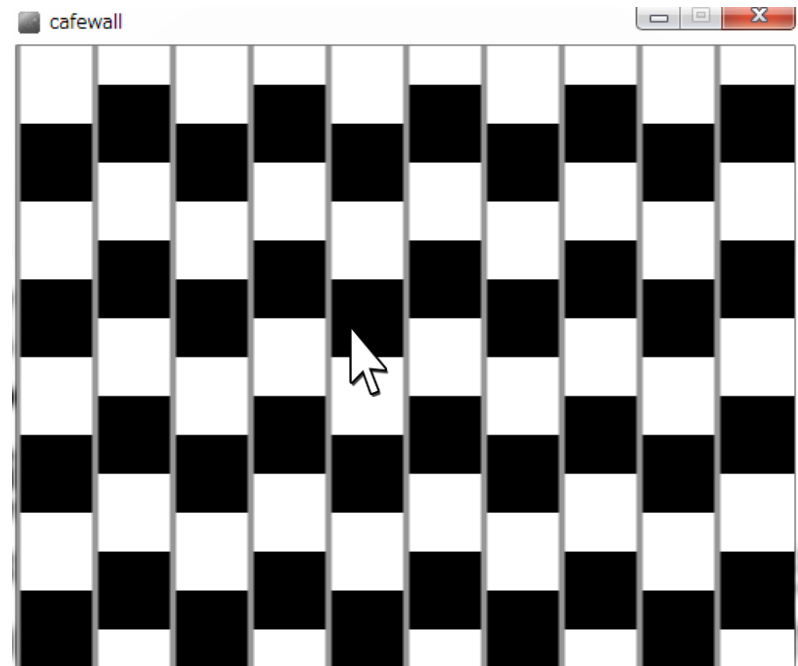
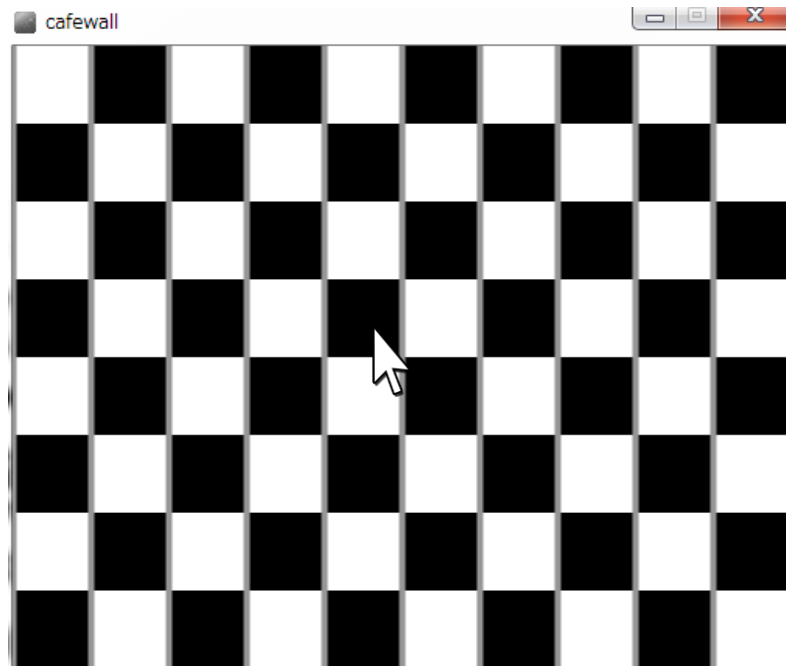
if文の分岐によって白または黒で塗り潰されるようにしてみよう。各マス目が白または黒になる条件とはなんだろうか？

プログラミング演習I (第7回) 課題

縦線が斜めに見える？

発展① スケッチ名: cafewall

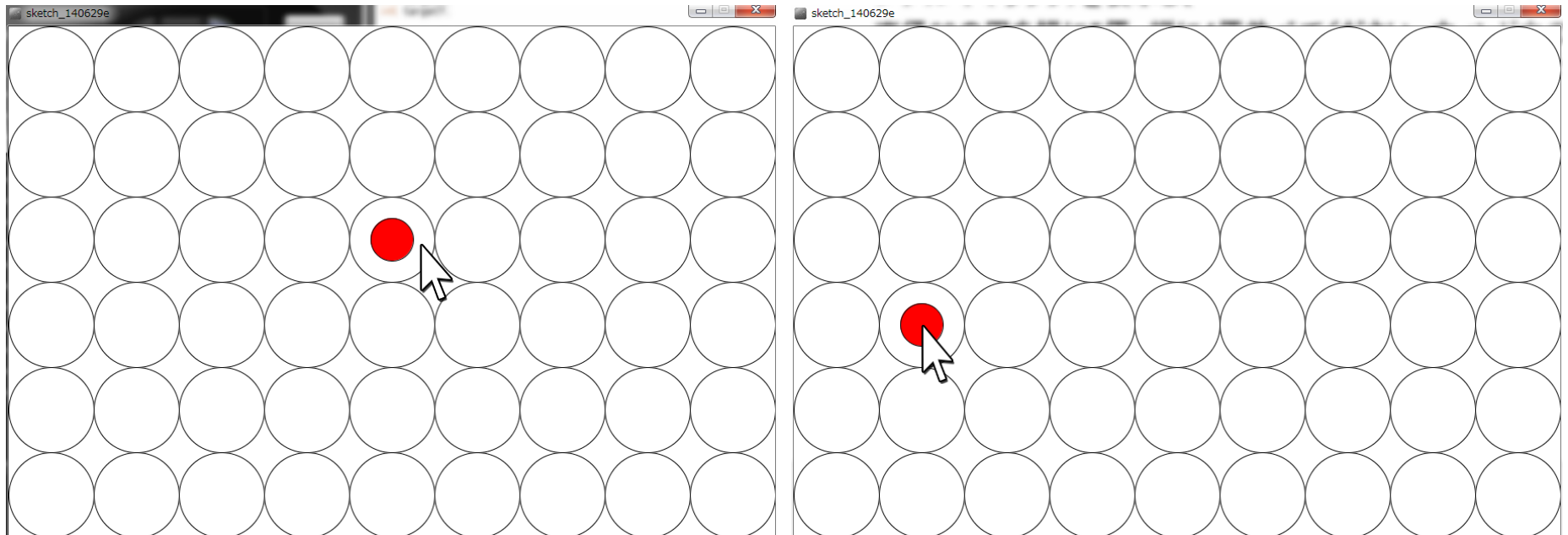
- 基本3のプログラムを改造して、まず1列おきに太さ4の灰色の縦線を入れるようにせよ。次に、偶数列について位置を25ピクセル縦に動かし、「カフェウォール錯視」のプログラムを作ってください。
- クリックする度に元の状態、カフェウォール錯視の状態と切り替るようにせよ



プログラミング演習I (第7回) 課題

• 発展② スケッチ名 : mogura

- 画面内を円で敷き詰め、任意の円に小さな赤丸(か、適当なオリジナルキャラクタ)を表示し、その赤丸が含まれる白丸をクリックすると、赤丸が別の任意の場所に移動できるようにせよ。
- 直径100の円を横に9個、縦に6個並べてください。ウィンドウのサイズは900x600とします。また、円が画面内にぴったり収まるようにしてください。また、Bashoを利用して現在の変数の状態を示せ。



Bashoの利用方法

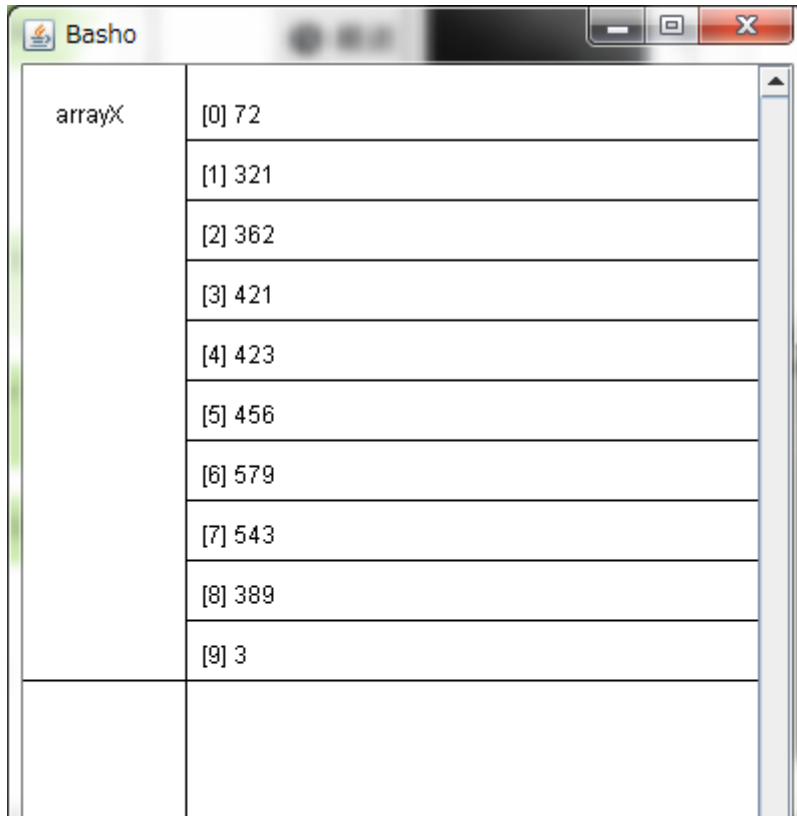
- まずは、下記URLからBashoをダウンロード
 - <https://github.com/kvvzr/Basho>
 - Basho.pdeを抜き出そう！
- 何かプログラムを書いて、そのプログラムの隣に(タブの部分に)Basho.pdeをドラッグアンドドロップしよう！
- おまじない(黄色の行のもの)を2行書こう
- プログラムを実行したら別ウィンドウで変数の状態が表示される！

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



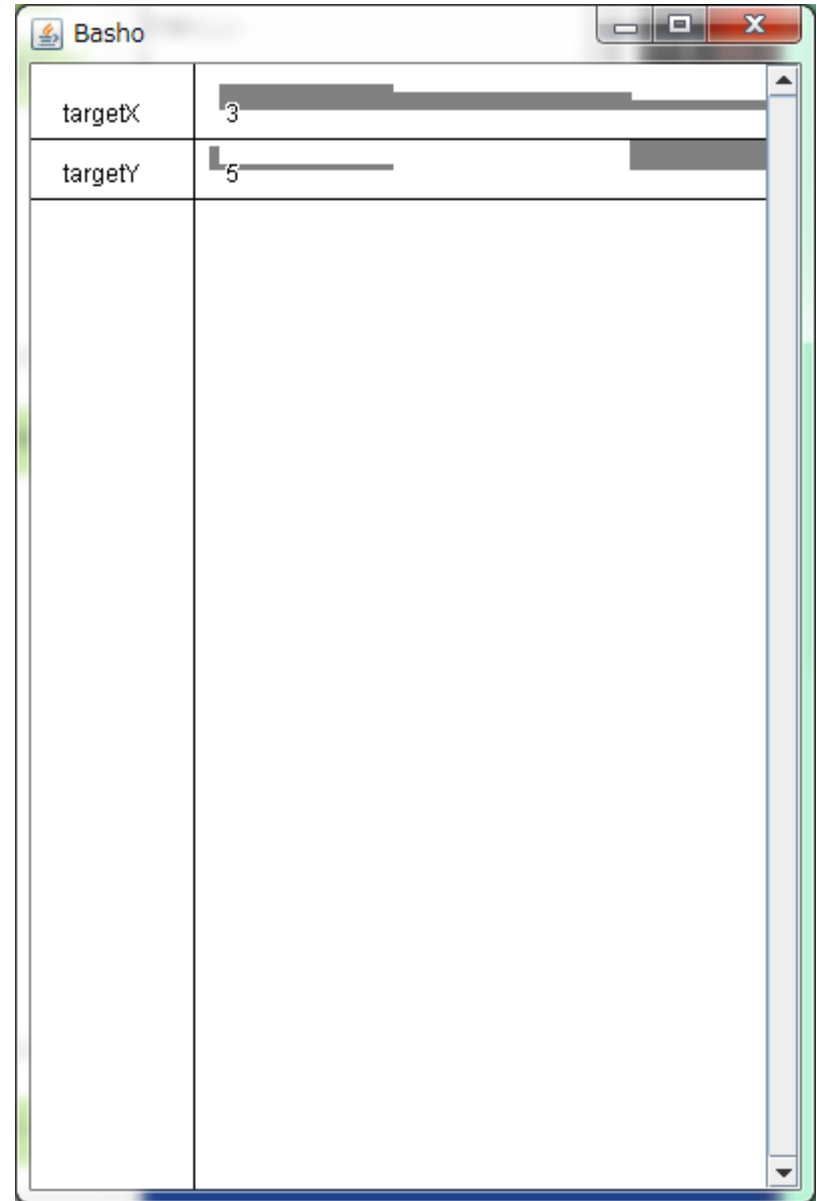
Bashoでどう表示されるか？

- 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 two columns. The left column is labeled "arrayX" and the right column contains the following values:

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 left column contains the labels "targetX" and "targetY", and the right column contains the values "3" and "5" respectively.

targetX	3
targetY	5