

## 復習1-1, 1-2

- 出力結果はどうか？

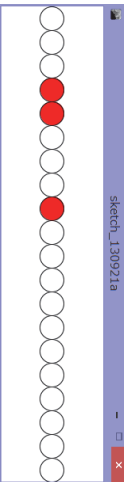
```
1 int x, y, z;
2 x = 10;
3 y = 20;
4 println( x + y );
5 z = x + y;
6 z = z * 5;
7 println( z );
8 println( x / z );
9
```

- ウインドウの右半分にはカーソルがある場合に背景の色を赤色にし、左半分にはカーソルがある場合に背景の色を白色にするには？

```
1 void setup(){
2   size( 400, 400 );
3 }
4
5 void draw(){
6   if( A ){
7     background( 255, 255, 255 );
8   } else {
9     background( B );
10  }
11 }
```

## 復習1-4(配列)

- 下記のような表示のプログラムを作るにはどうしたらよいか、右記プログラムの穴をうめよ



```
int [] light = new int [20];
void setup(){
  size( 600, 100 );
  for( int x=0; x<20; x++){
    light[x] = 0;
  }
  light[ A ] = 1;
  light[ B ] = 1;
  light[ C ] = 1;
}
void draw(){
  background( 255, 255, 255 );
  int x = 0;
  while( D ){
    if( light[x] == 1 ){
      fill( 255, 0, 0 );
    } else {
      fill( 255, 255, 255 );
    }
    ellipse( E, 50, 30, 30 );
    x++;
  }
}
```

## 復習1-3(繰り返し)

- プログラムの問題を指摘せよ
- 左側のような表示を行いたいが、動作しない。問題を指摘せよ。

sketch\_140920b



```
1 size( 400, 400 );
2 background( 255 );
3 int i = 0;
4 while( i < 400 ){
5   i = 0;
6   line( i, 0, i, 400 );
7   i = i + 10;
8 }
9 while( i < 400 ){
10  i = 0;
11  line( 0, i, 400, i );
12  i = i + 10;
13 }
```

## 復習1-5(関数/メソッド)

- 整数の値を引数として渡すと、約数の数を返す関数を作成せよ

```
int getNumberOfDivison( int num ){
  int i=1;
  int count = A;
  while( i<=num ){
    if( B ){
      count++;
    }
    i++;
  }
  return C;
}
```

## 課題1-1

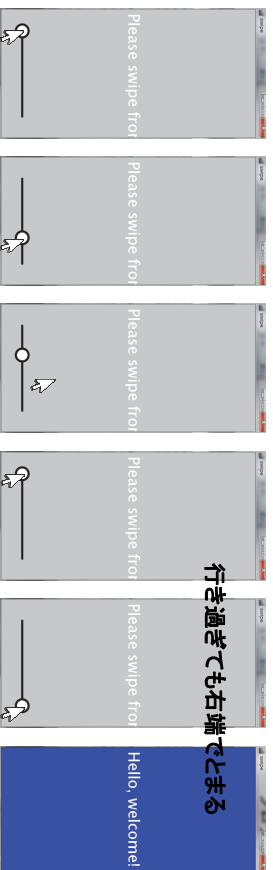
- Fizz-Buzzを行うプログラムを作成せよ
  - Fizz-Buzzとは、3で割り切れる場合は「Fizz」、5で割り切れる場合は「Buzz」、両方で割り切れる場合は「Fizz Buzz」を数の代わりに発言するというもの
  - 下記のように1~200までについて、連続的に表示するようにせよ(1つずつ改行してもよい)
- 【出力例】
- ```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, Fizz
Buzz, 16, 17, Fizz, 19, Buzz, Fizz, 22, 23, Fizz, Buzz, 26, Fizz,
28, 29, Fizz Buzz, 31, 32, Fizz, 34, Buzz, Fizz, ...
```

## 宿題1-1

- 復習1-5で作成した約数の数を返す関数を活用し、引数として指定した数字が素数かどうかを判定する関数(素数判定関数)を作成せよ
- (ヒント) 約数の数が2だったら1を返し、2出なかったら0を返す関数を作る
- その関数(メソッド)を利用して、2から各自の学番号(8桁)までのすべての整数について素数かどうかを判定し、素数の場合はその値を `println` で標準出力するプログラムを作成せよ

## 宿題1-2

- 縦長のウインドウの下部に左の方に丸型のものを用意し、それを右側にスライド(スワイプ)させると、ロックが解除され何らかの画面に遷移する仕組みを実現せよ(例えば、作成したキヤラクタが描画されている画面など)
- なお、右端まで移動せずに手を離れた場合は、5ピクセル/フレームの速度で最初の位置に戻るようにせよ。また、指定の位置より右や左にはみ出ないようにせよ。



## 宿題1-3

- コラッツ予想とは、下記のルールに従うとすべての自然数が最終的に1になるのではないかという予想である。また、本予想は証明されていないが、 $3 \times 2^53$ までは反例が無いことが確かめられている。ルールは下記のとおり
  - ある数が偶数なら2で割る
  - ある数が奇数なら3を掛けて1を足す
  - 計算結果が1になるまで上記の計算を繰り返し返す
- ここで、2から100までの数字において、すべての値の変化を右下のように表示せよ。また、その最後にスワップ数も表示せよ。なお、そのために下記の仕様を満たす関数 `Collatz` を作成せよ
- $n$  が偶数の時  $Collatz(n) = \frac{n}{2}$
  - $n$  が奇数の時  $Collatz(n) = 3n + 1$

```
(2) => 1 (1 steps)
(3) => 10->5->16->8->4->2->1 (7 steps)
(4) => 2->1 (2 steps)
(5) => 16->8->4->2->1 (5 steps)
(6) => 9->10->5->16->8->4->2->1 (8 steps)
(7) => 22->11->5->17->52->26->13->40->20->10->5
(8) => 4->2->1 (3 steps)
(9) => 28->14->7->22->11->34->17->52->26->13->40
(10) => 5->16->8->4->2->1 (6 steps)
(11) => 34->17->52->26->13->40->20->10->5->16->8
(12) => 6->3->10->5->16->8->4->2->1 (9 steps)
(13) => 40->20->10->5->16->8->4->2->1 (9 steps)
(14) => 7->22->11->34->17->52->26->13->40->20->1
(15) => 46->23->70->35->106->53->160->80->40->20
(16) => 8->4->2->1 (4 steps)
(17) => 52->26->13->40->20->10->5->16->8->4->2->1
(18) => 17->26->13->20->10->5->16->8->4->2->1
```