

組	番	氏名	点数

設問 1

1~300 までの Fizz-Buzz を行うプログラムを作成している。ただ、思ったように結果が出力されない。下記のプログラムにある問題を指摘せよ。修正内容は、プログラムに直接書き込め。

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である。

(ヒント) 誤りは 5 箇所程度ある (類似誤りは 1 つの誤りとする)

```
int num = 1;
while( num <= 300 ){
    if( num / 3 == 0 ){
        println( "Fizz" );
    } if( num / 5 == 0 ){
        println( "Buzz" );
    } if( num / 15 == 0 ){
        println( "Fizz Buzz" );
    } else {
        println( num );
    }
}
```

設問 2

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 になるのではという予想である。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す
- 2 から 100 までの数について、その数と全てのステップを「[3]->10->5->16->8->4->2->1」のように表示するように穴を埋めよ。

```
int Collatz(  ) {
    if ( n % 2 == 0 ) {
        return ;
    }
    return ;
}

void setup() {
    int number = 2;
    while ( number <= 100 ) {
        int retNum = number;
        print( "[" + number + "]" );
        while (  ) {
            ;
            print( "->" + retNum );
        }
        print( "¥n" );
        number++;
    }
}
```

解答欄

①

②

③

④

⑤

(裏へ続く)

設問 3

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 (isPrimeNumber 素数判定関数) を作成せよ. 1 からその数までの約数の数が 2 だったら 1 を返し, 2 でなかったら 0 を返す関数を作って利用する. また, その関数 (メソッド) を利用して, 2 から 10000 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. ただし, count は約数の数を数える変数とする.

解答欄
①
②
③
④
⑤
⑥
⑦

```
int isPrimeNumber ( int num ) {
    int i = 1;
    int count = ①;
    while( i <= num ) {
        if( ② ) {
            ③;
        }
        i++;
    }

    if( ④ ) {
        return 0;
    }
    return 1;
}

void setup() {
    int i = ⑤;
    while( i <= 10000 ) {
        if( ⑥ ) {
            println( i + " is prime number." );
        }
        ⑦;
    }
}
```

組	番	氏名	点数

設問 1

1~300 までの Fizz-Buzz を行うプログラムを作成している。ただ、思ったように結果が出力されない。下記のプログラムにある問題を指摘せよ。修正内容は、プログラムに直接書き込め。

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である。

(ヒント) 誤りは 5 箇所程度ある (類似誤りは 1 つの誤りとする)

```
int SayFizzBuzz( int num ){
    if( num % 3 == 1 || num % 5 == 1 ){
        println( "Fizz Buzz" );
    } else if( num % 3 == 1 ){
        println( "Fizz" );
    } else if( num % 5 == 1 ){
        println( "Buzz" );
    } else {
        println( num );
    }
}

void setup() {
    int i = 1;
    while( i <= 300 ){
        SayFizzBuzz( num );
    }
}
```

設問 2

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 になるのではという予想である。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す
- 2 から 100 までの数について、その数と全てのステップを「[3]->10->5->16->8->4->2->1」のように表示するように穴を埋めよ。

```
int Collatz(  ) {
    if ( n % 2 == 0 ) {
        return ;
    }
    return ;
}

void setup() {
    int number = 2;
    while ( number <= 100 ) {
        int retNum = number;
        print( "[" + number + "]" );
        while (  ) {
            retNum = ;
            print( "->" + retNum );
        }
        print( "\n" );
        number++;
    }
}
```

解答欄
①
②
③
④
⑤

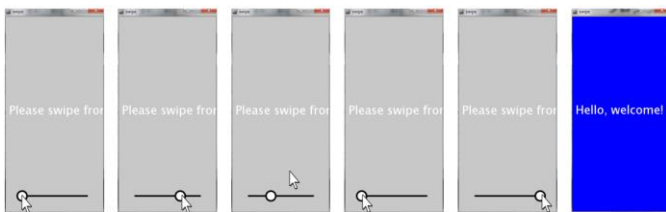
(裏へ続く)

設問 3

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、それを左から右にスライド（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。

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。

本プログラムでは、ロックの状態を管理するための変数として `lock` を、丸型のものを選択状態を管理するための変数として `selected` というものを用意している。



解答欄

①

②

③

④

⑤

⑥

⑦

```
int lock = ①;
int selected = 0;
int cx = 50, cy = 550;

void setup() {
  size( 300, 600 );
  strokeWeight( 5 );
  textSize( 36 );
}

void draw() {
  background( 200 );
  if ( ② ) {
    background( 0, 0, 255 );
    text( "Hello, welcome!", 10, 300 );
  } else {
    text( "Please swipe!", 10, 300 );
    line( 50, 550, 250, 550 );
    if ( selected == 1 ) {
      cx = ③;
    } else {
      cx = ④;
    }
    if ( cx < 50 ) { cx = 50; }
    if ( cx > 250 ) { cx = 250; }
    ellipse( cx, cy, 30, 30 );
  }
}

void mousePressed() {
  if( dist(cx, cy, mouseX, mouseY) <= 15 ) {
    selected = ⑤;
  }
}

void mouseReleased() {
  selected = ⑥;
  if ( cx == 250 ) {
    lock = ⑦;
  }
}
```