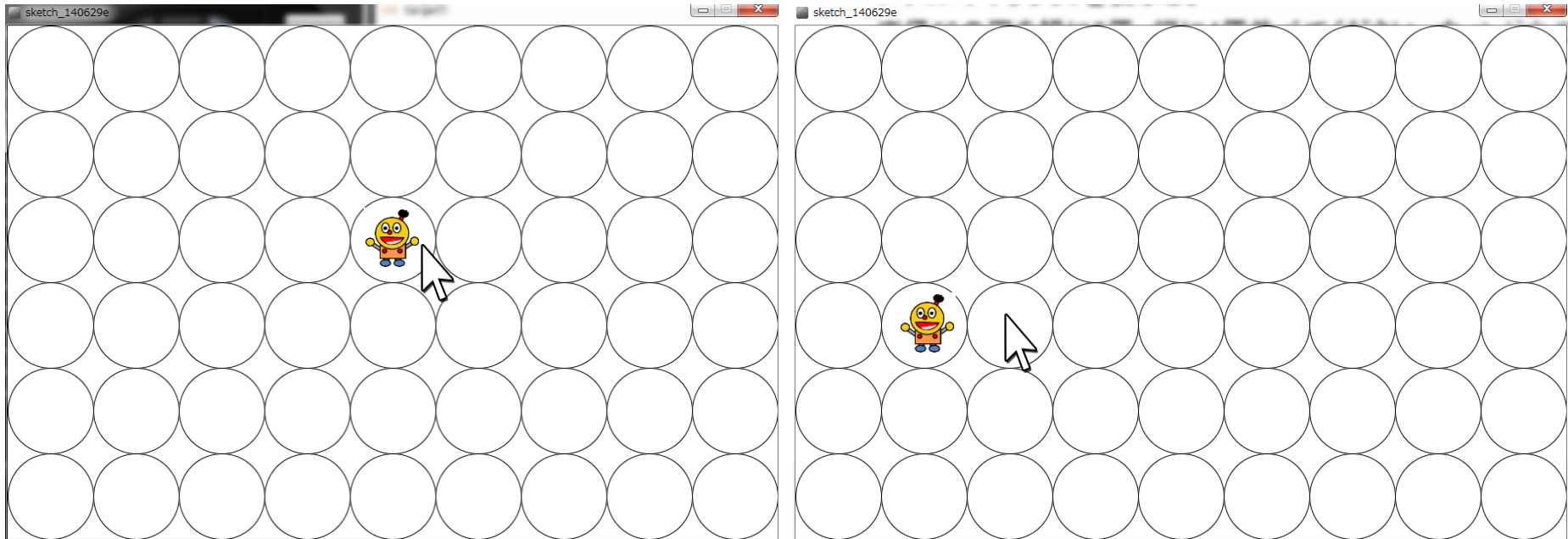


プログラミング演習I (第10回) 課題

• 基本① スケッチ名 : moguraChara

- キャラクタを描画する関数を作成し(参照: 参考資料)、その関数を利用したもぐらたたきを作成せよ。ただし、画面内を円で敷き詰め、任意の円に小さなキャラクタを表示し、そのキャラクタが含まれる白丸をクリックすると、キャラクタ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。



自分のキャラを表示する関数を作ろう

```
void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
```

```
int offset_x = 190;
int offset_y = 200;
```

← offset_x と offset_y はキャラの中心座標。
← 自分のキャラクタに合わせて値を変えること。

```
pushMatrix();
translate( x, y );
scale( scale_factor );
rotate( radians( angle_deg ) );
translate( -offset_x, -offset_y );
```

ここにchara1で作った、キャラクタを描画する処理をコピーしてください。
ただし、size()とbackground()は入れないこと。

```
popMatrix();
}
```

drawCharacter() の使い方

drawCharacter(X座標, Y座標, 回転角度, 拡大率);

回転角度の単位は【度】、拡大率は1.0で等倍。

```
void setup() {
  size( 600, 400 );
}

void draw() {
  background(255);

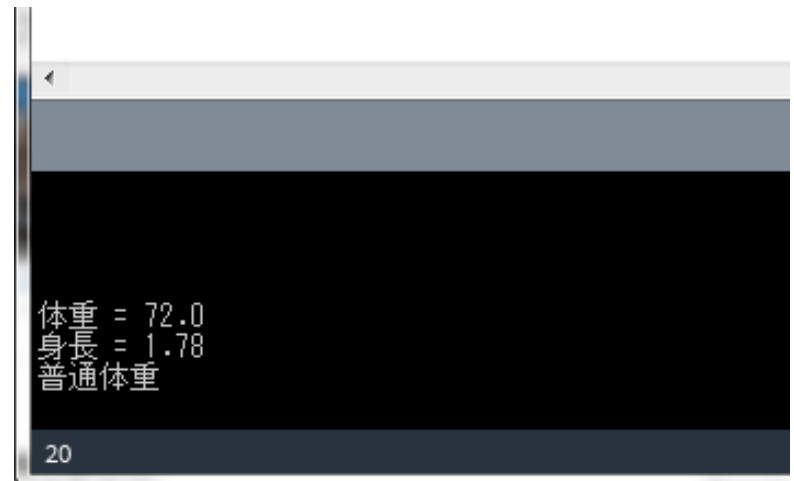
  // 例 : マウスカーソルがある位置に45度傾けて0.5倍でキャラクタを描画
  drawCharacter( mouseX, mouseY, 45.0, 0.5 );
}

void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
  .....
}
```

プログラミング演習I (第10回) 課題

• 基本② スケッチ名: BMI

- BMIを計算して結果を返す関数を作成せよ
- ただし, BMIは体重÷(身長×身長)で求めることができる
 - なお, 体重は単位がkgで, 身長は単位がmとなる
- 自分または友人の身長と体重とBMI関数を利用し, 身長と体重を表示した後, BMIが18.5未満の場合は痩せ型を, 18.5以上25未満の場合は普通体重を, 25以上の場合は肥満体重と表示するようにせよ.
- グローバル変数を用いず, BMI関数の中ではprintしないこと

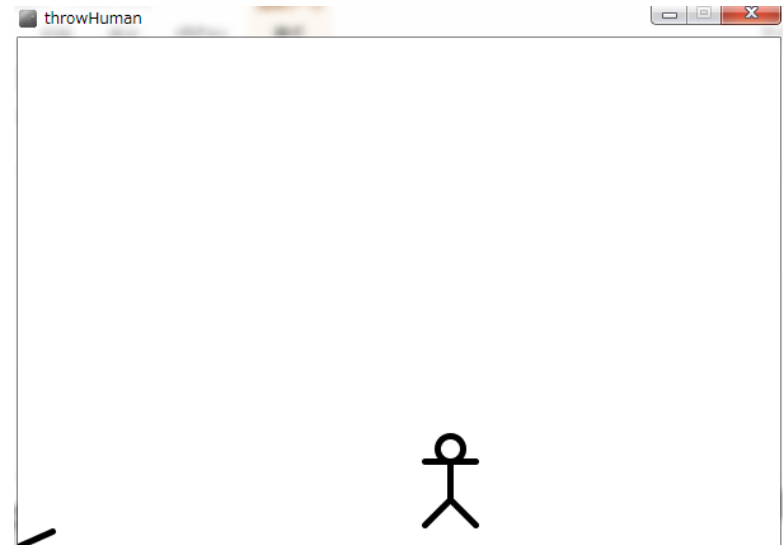
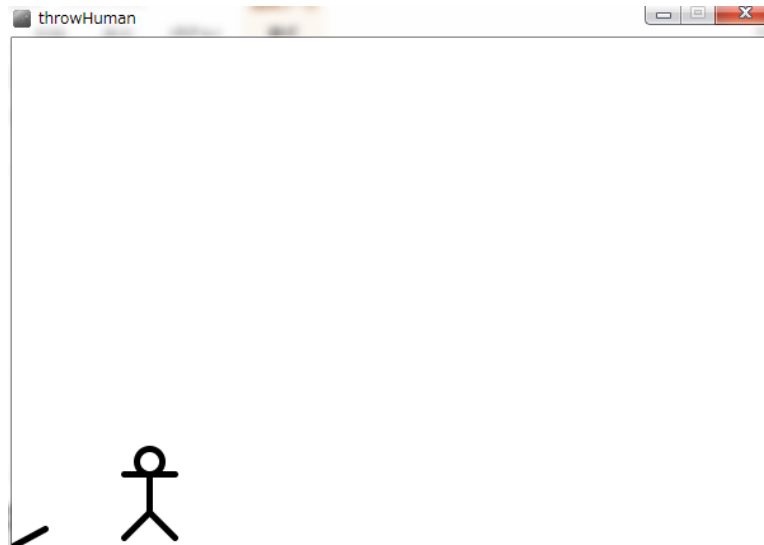


```
体重 = 72.0  
身長 = 1.78  
普通体重  
20
```

プログラミング演習I (第10回) 課題

• 基本③ スケッチ名 : jumpHuman

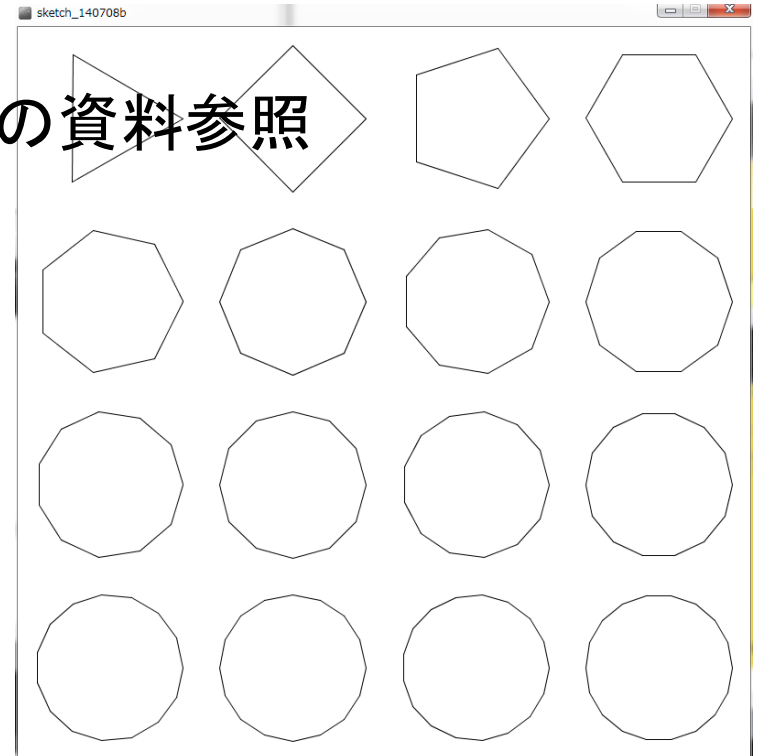
- 棒人間を描画する関数を作成し, その関数を利用して棒人間がジャンプする(斜方投射される)プログラムを作ってください。
- 600x400のウィンドウ左下からマウスカーソルがある方向に棒人間を発射するジャンプ台のプログラムを作ってください。クリックされたタイミングでジャンプ(投射)されるようにすること。また、クリック時のジャンプ台からカーソルまでの距離を初速とせよ。
- なお, 右端まで来ると跳ね返るようにせよ。



プログラミング演習I (第6回) 課題

• 発展① スケッチ名: regular_polygon

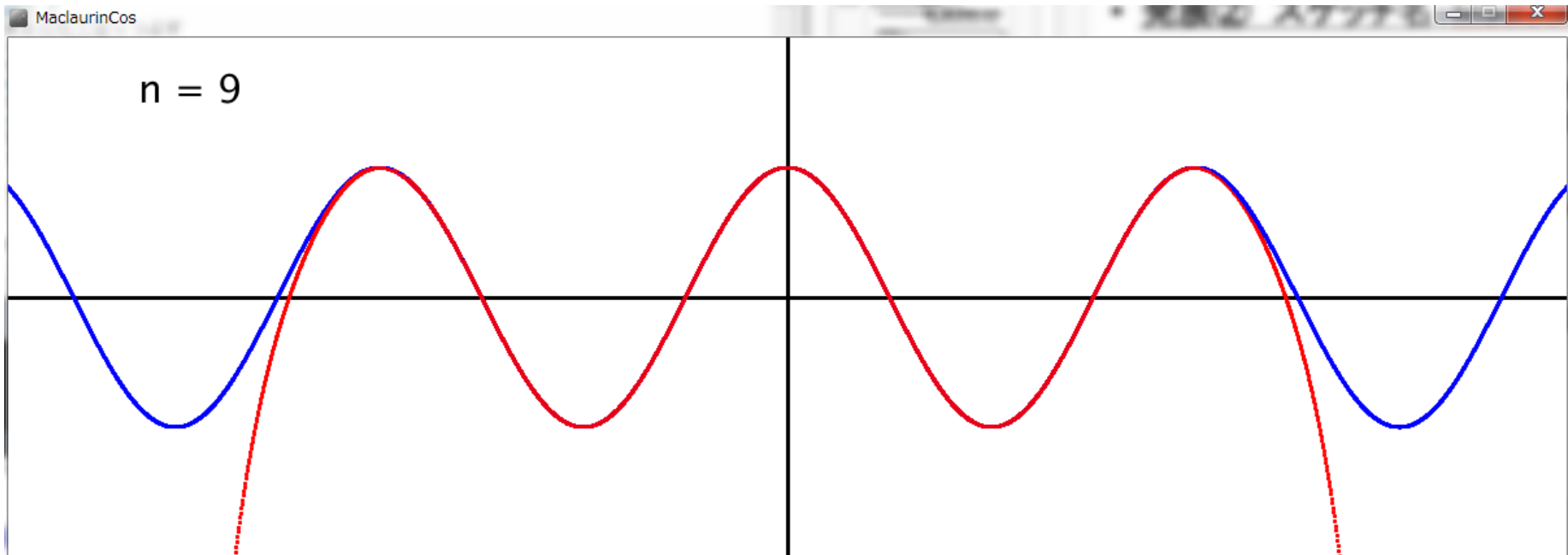
- 中心座標(cx , cy)と, 半径 R , 角数 num を指定することによって正多角形を描画するRegularPolygonという関数を作成せよ
- また, そのRegularPolygonという関数を利用することで, 800x800のウィンドウ上に正三角形から正十八角形まで並べるようにせよ
- 正多角形の描画については過去の資料参照



プログラミング演習I (第10回) 課題

• 発展② スケッチ名: Maclaurin

- マクローリン展開(テイラー展開の $a=0$ のもの)を利用すると、様々な数式の多項式近似を行うことができる。これを用いて $\cos x$ と近似できることを図示せよ。
- $\cos x$ のマクローリン展開については次ページで説明する。骨組みを配布するので、それに従い関数を作成せよ。



プログラミング演習I (第10回) 課題

• 発展② スケッチ名: Maclaurin

– $\cos x$ は下記のような式にマクローリン展開可能である。このマクローリン展開を行うための関数Maclaurinを完成させよ。ただし、Maclaurin関数は、入力をxの値と項数(n)とし、その時の値を返すようにせよ。

$$- \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

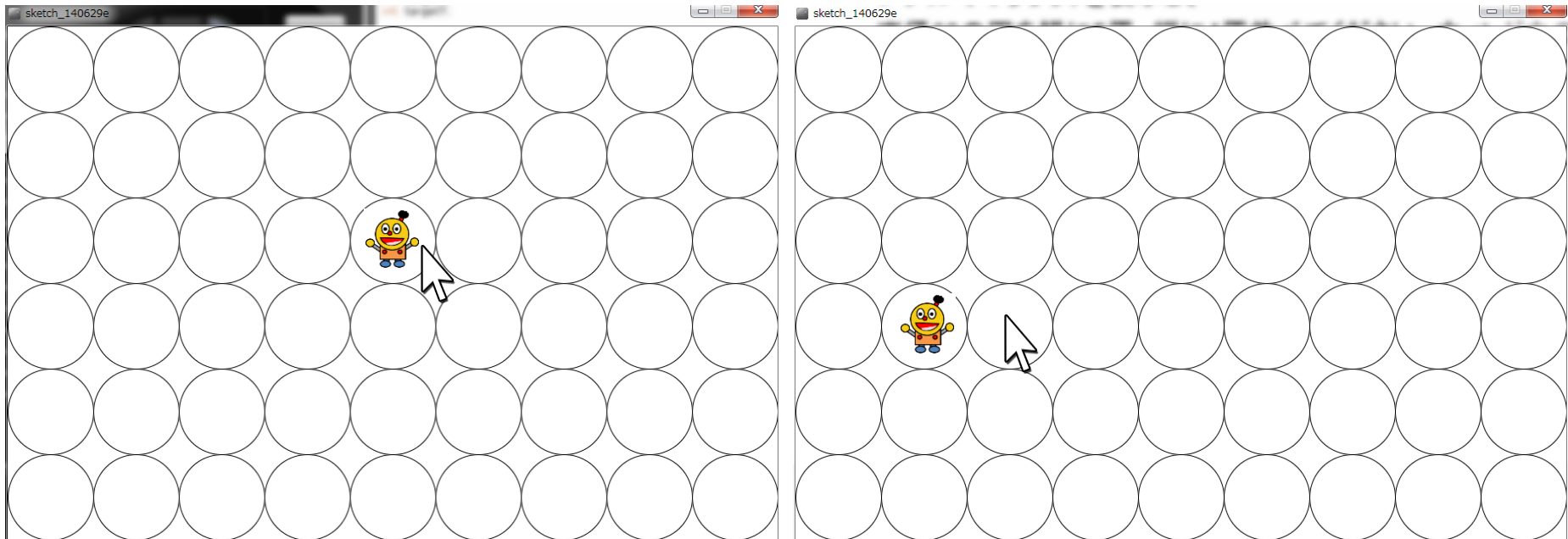
$$- f(x, n) = (-1)^n \frac{x^{2n}}{(2n)!} \text{ とすると}$$

$$- \text{Maclaurin}(x, n) = \sum_{i=0}^n f(x, i) \text{ となる}$$

プログラミング演習I (第10回) 課題

• 基本① スケッチ名 : moguraChara

- キャラクタを描画する関数を作成し(参照: 参考資料)、その関数を利用したもぐらたたきを作成せよ。ただし、画面内を円で敷き詰め、任意の円に小さなキャラクタを表示し、そのキャラクタが含まれる白丸をクリックすると、キャラクタが別の任意の場所へ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。



自分のキャラを表示する関数を作ろう

```
void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
```

```
int offset_x = 190;
int offset_y = 200;
```

← offset_x と offset_y はキャラの中心座標。
← 自分のキャラクタに合わせて値を変えること。

```
pushMatrix();
translate( x, y );
scale( scale_factor );
rotate( radians( angle_deg ) );
translate( -offset_x, -offset_y );
```

ここにchara1で作った、キャラクタを描画する処理をコピーしてください。
ただし、size()とbackground()は入れないこと。

```
popMatrix();
```

```
}
```

drawCharacter() の使い方

drawCharacter(X座標, Y座標, 回転角度, 拡大率);

回転角度の単位は【度】、拡大率は1.0で等倍。

```
void setup() {
  size( 600, 400 );
}

void draw() {
  background(255);

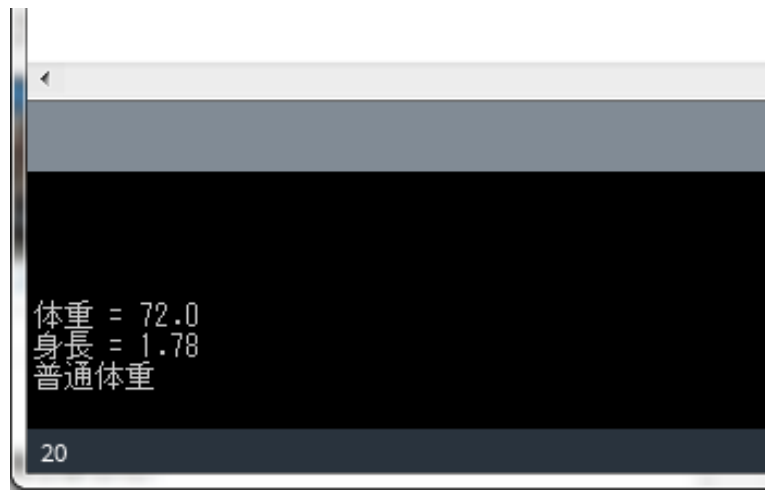
  // 例 : マウスカーソルがある位置に45度傾けて0.5倍でキャラクタを描画
  drawCharacter( mouseX, mouseY, 45.0, 0.5 );
}

void drawCharacter(float x, float y, float angle_deg, float scale_factor ) {
  .....
}
```

プログラミング演習I (第10回) 課題

• 基本② スケッチ名: BMI

- BMIを計算して結果を返す関数を作成せよ
- ただし, BMIは体重÷(身長×身長)で求めることができる
 - なお, 体重は単位がkgで, 身長は単位がmとなる
- 自分または友人の身長と体重とBMI関数を利用し, 身長と体重を表示した後, BMIが18.5未満の場合は痩せ型を, 18.5以上25未満の場合は普通体重を, 25以上の場合は肥満体重と表示するようにせよ
- グローバル変数を用いず, BMI関数の中ではprintしないこと

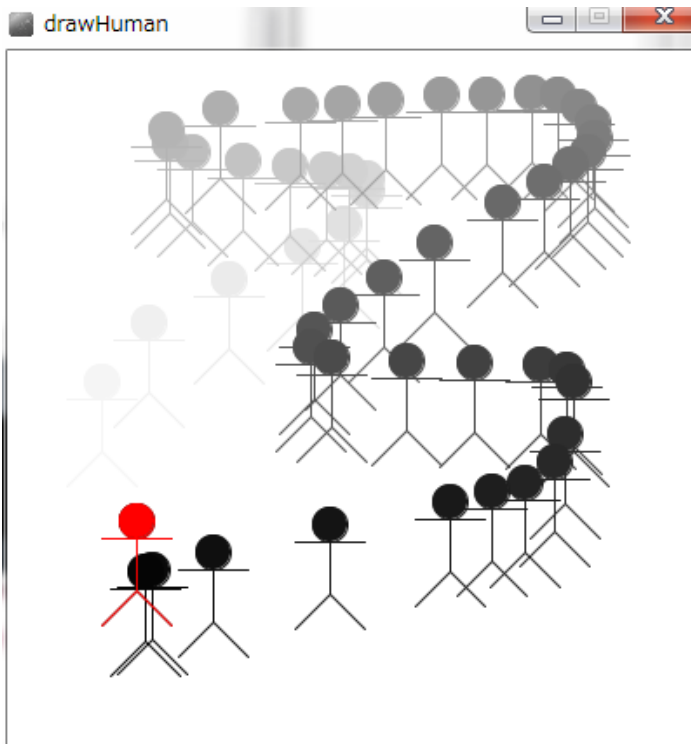


```
体重 = 72.0  
身長 = 1.78  
普通体重  
20
```

プログラミング演習I (第10回) 課題

• 基本③ スケッチ名: drawHuman

- 配列と繰り返しを使ってマウスカーソルに追尾する棒人間を50人描いてみましょう。
- ただし、先頭の棒人間は赤色で描画し、徐々に薄くなるようにせよ。



[step1]

まずは予習テキストにある棒人間を描くプログラムを動かしてみよう。

[step2]

棒人間を描く関数を使って縦横に並べるには、2重の繰り返し文(for文またはwhile文)を使えばよい。座標の指定を工夫してみよう。

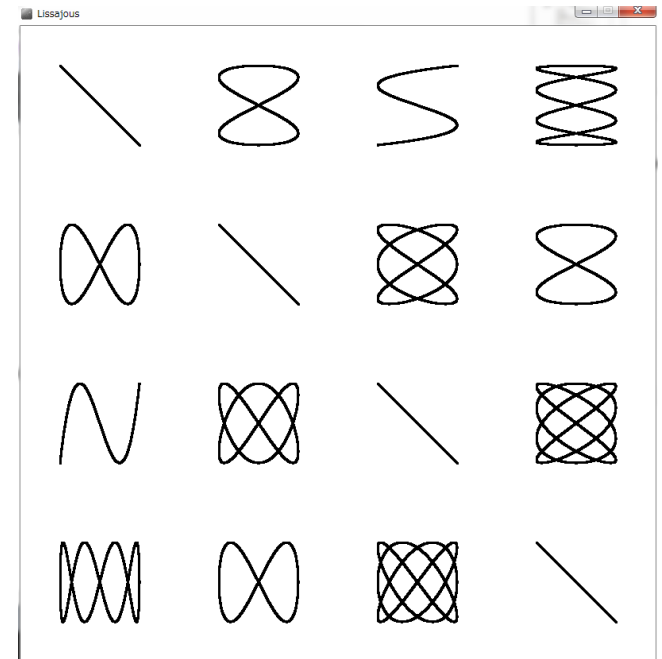
[step2]

棒人間を描く関数を使って縦横に並べるには、2重の繰り返し文(for文またはwhile文)を使えばよい。座標の指定を工夫してみよう。

プログラミング演習I (第6回) 課題

• 発展① スケッチ名 : Lissajous

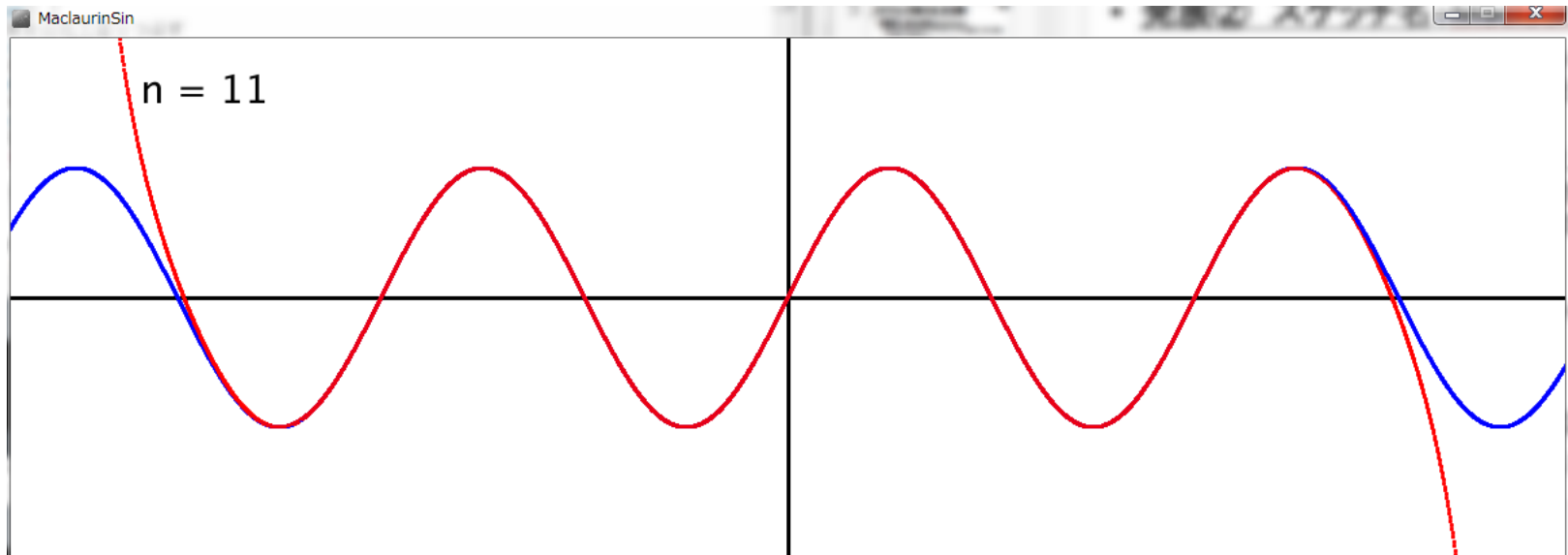
- 800x800のウィンドウ上に, 中心座標(cx , cy)と, 下記の式の a と b と w を指定することでリサージュカーブを表示する `drawLissajous` という関数を作成せよ. w は50とする.
- 横方向に a を1~4と増やせ
- 縦方向に b を1~4と増やせ
 - $x = w \times \sin(at) + cx$
 - $y = w \times \sin(bt) + cy$



プログラミング演習I (第6回) 課題

• 発展② スケッチ名: Maclaurin

- マクローリン展開(テイラー展開の $a=0$ のもの)を利用すると、様々な数式の多項式近似を行うことができる。これを用いて $\sin x$ と近似できることを図示せよ。 $\sin x$ のマクローリン展開については次ページで説明する。骨組みを配布するので、それに従い関数を作成せよ。



プログラミング演習I (第6回) 課題

• 発展② スケッチ名: Maclaurin

– $\sin x$ は下記のような式にマクローリン展開可能である。このマクローリン展開を行うための関数Maclaurinを完成させよ。ただし、Maclaurin関数は、入力をxの値と項数(n)とし、その時の値を返すようにせよ。

$$- \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$$

$$- f(x, n) = (-1)^n \frac{x^{2n+1}}{(2n+1)!} \text{ とすると}$$

$$- \text{Maclaurin}(x, n) = \sum_{i=0}^n f(x, i) \text{ となる}$$