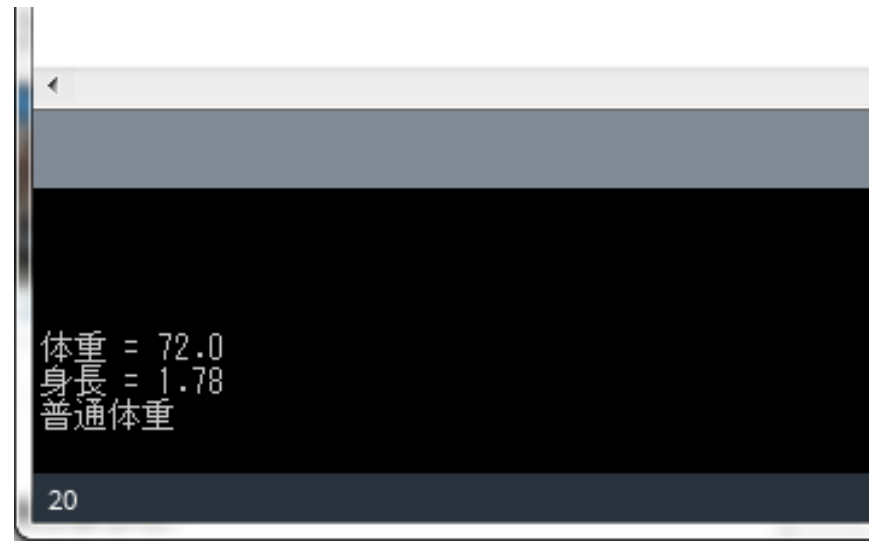


プログラミング演習I（第10回）課題

• 基本① スケッチ名: BMI

- 体重と身長を入力とし, BMI値を返す関数を作成せよ
- ただし, BMIは $\text{体重} \div (\text{身長} \times \text{身長})$ 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」, 18.5以上25未満の場合は「普通体重」, 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ないこと
- BMI関数の中ではprintしないこと



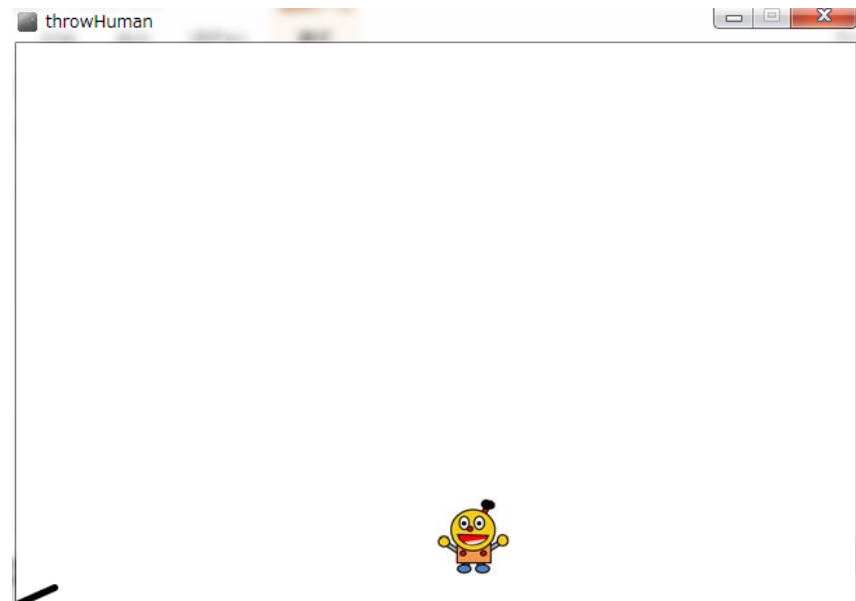
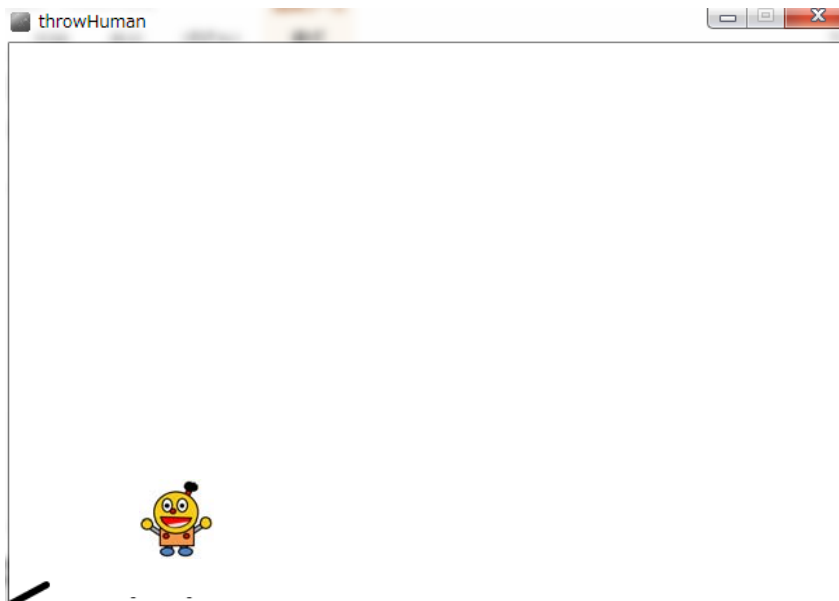
```
体重 = 72.0  
身長 = 1.78  
普通体重
```

20

プログラミング演習I（第10回）課題

• 基本② スケッチ名: jumpChara

- キャラクタを描画する関数を作成し(参照: 参考資料), その関数を利用してキャラクタがジャンプする(斜方投射される)プログラムを作れ
- 600x400のウィンドウ左下からマウスカーソルがある方向にキャラクタ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。
- 右端まで来ると跳ね返るようにせよ(右端の判定はある程度でよい)



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

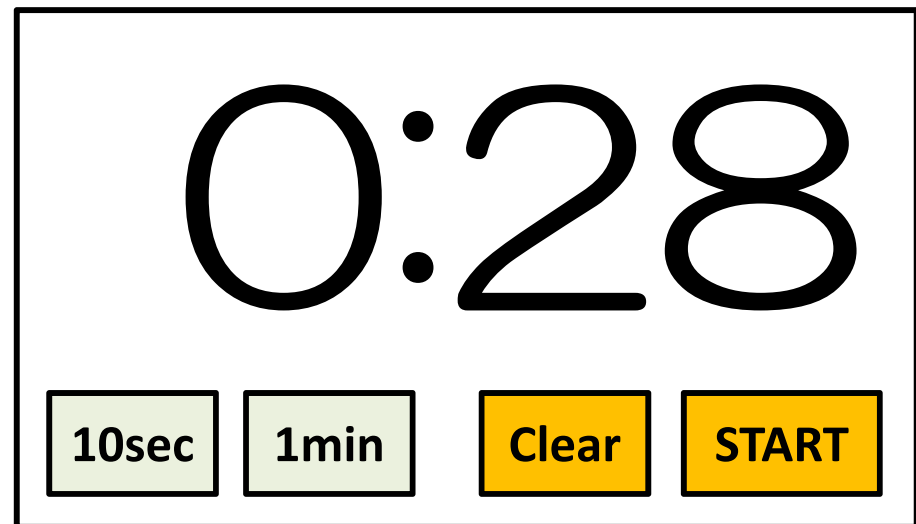
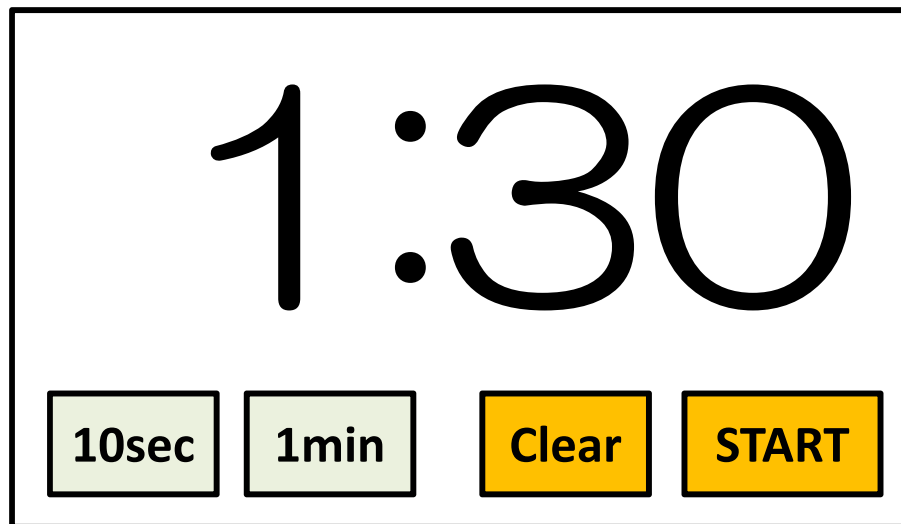
• 基本③ スケッチ名 : DoReMi

- 配布したプログラムの幹音(ドレミファソラシド)となる周波数を求める関数 `getFrequency` を作成し、キーボード操作の上下によって音程を上下させるプログラムを完成させよ。
- 関数の引数は幹音のIDとし、返り値はその周波数の値(float)とせよ。
 - ド 261.6Hz
 - レ 293.7Hz
 - ミ 329.6Hz
 - ファ 349.2Hz
 - ソ 392.0 Hz
 - ラ 440.0
 - シ 493.9Hz
- 幹音のIDが0のときは261.6Hzのド、1のときは293.7Hzのレとなるようにすること。
- なお、1オクターブ上がるとそれぞれ周波数は2倍に、1オクターブ下がると周波数は1/2倍になる。
- 少なくとも3オクターブ分の結果を返すようにせよ。

プログラミング演習I（第10回）課題

• 発展① スケッチ名 : PresentationTimer

- 画面に時間(10sec、1min)を模したボタンが表示され、それらをクリックするごとに合計時間が増えていくプログラムを作成してください。
- クリアを押すと0:00に戻るようにし、スタートボタンを押すとカウントダウンを開始し、時間が来ると画面を赤色にしたり、音を鳴らしたりせよ。
- 現在の時間を0時0分0秒からみて何秒にあたるかということを、秒単位で取得する関数を作成し、利用せよ。
- また、その関数を利用してSTARTボタンを押した時に現在の秒数を取得し、そこからの差で経過時間を計算できるようにせよ。



P演習 I 発表会スケジュール表

時間	研究室	人数	注意事項 ・発表は18:10開始 ・研究室から研究室へのトランジション時間は2分 ・研究室メンバ間のトランジション時間はなし ・個人の発表開始時間と終了時間とが厳密に決まっています (リレーのように研究室内でトランジションの練習が必要！！)
18:10:00	福地	6	
18:11:30			
18:13:00			
18:14:30			
18:16:00			
18:17:30			
18:21:00	嵯峨山	4	
18:22:30			
18:24:00			
18:25:30			
18:29:00	橋本	6	
18:30:30			
18:32:00			
18:33:30			
18:35:00			
18:36:30			
18:40:00	中村	6	
18:41:30			
18:43:00			
18:44:30			
18:46:00			
18:47:30			
18:51:00	小松	6	
18:52:30			
18:54:00			
18:55:30			
18:57:00			
18:58:30			
19:02:00	阿原	7	
19:03:30			
19:05:00			
19:06:30			
19:08:00			
19:09:30			
19:11:00			
19:14:30	渡邊	6	
19:16:00			
19:17:30			
19:19:00			
19:20:30			
19:22:00			
19:33:00	鹿喰	6	
19:34:30			
19:36:00			
19:37:30			
19:39:00			
19:40:30			

19:44:00	斎藤	6	
19:45:30			
19:47:00			
19:48:30			
19:50:00			
19:51:30			
19:55:00	菊池	7	
19:56:30			
19:58:00			
19:59:30			
20:01:00			
20:02:30			
20:04:00			
20:07:30	小林	7	
20:09:00			
20:10:30			
20:12:00			
20:13:30			
20:15:00			
20:16:30			
20:20:00	荒川	7	
20:21:30			
20:23:00			
20:24:30			
20:26:00			
20:27:30			
20:29:00			
20:32:30	鈴木	7	
20:34:00			
20:35:30			
20:37:00			
20:38:30			
20:40:00			
20:41:30			
20:44:00	五十嵐	6	
20:45:30			
20:47:00			
20:48:30			
20:50:00			
20:51:30			
20:55:00	宮下	6	
20:56:30			
20:58:00			
20:59:30			
21:01:00			
21:02:30			