

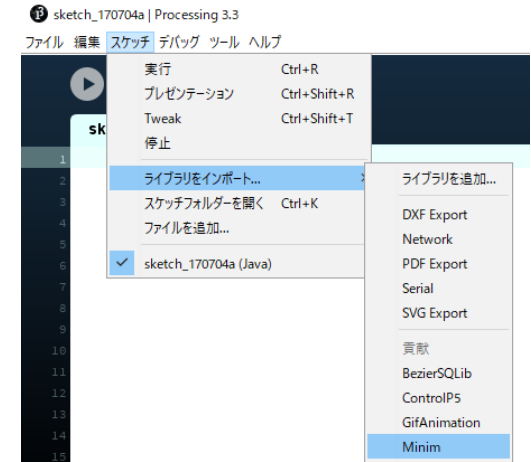
プログラミング演習I (第10回) 課題

• 基本① スケッチ名: basic_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。
- まず準備段階として `minim` を環境に導入せよ (後述)
- 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 392.0Hz, ラ 440.0Hz, シ 493.9Hz
- プログラムとしていじるのは `getFrequency` 内部だけ。今回は `getFrequency` がかなり長くなってもよい。
- 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の周波数は2倍になる。例えば、1オクターブ上のドは $261.6 * 2$ となり、1オクターブ上のレは $293.7 * 2$ Hz となる。
- 少なくとも3オクターブ分 (少なくとも `toneID` が20まで) の結果を返すようにせよ。音はある程度聞こえていればOKとする。

minimの利用方法

- スケッチ > ライブラリをインポート
> Minim で利用できます
 - ない環境ではライブラリを追加しよう！
 - ライブラリを追加で基本的にはできるはず



プログラミング演習I (第10回) 課題

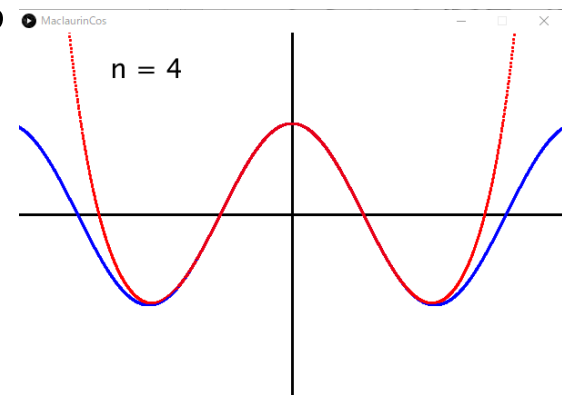
• 基本② スケッチ名: basic_MaclaurinCos

- $\cos x$ は下記の式にマクローリン展開 (テイラー展開の $a=0$ のもの) で多項式近似可能である。このマクローリン展開を行うための関数 `Maclaurin`は、階乗と累乗の組み合わせで表現できる。
- 階乗`calcFactorial`と累乗`calcPower`の関数を完成させることで、 $\cos x$ と近似できることを確認せよ。`calcPower`は`pow`を使わず実現しよう。

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

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

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



プログラミング演習I (第10回) 課題

• 基本課題③ basic_AmicableNumbersFunc

- 友愛数とは、異なる2つの自然数の組で、自分自身を除いた約数の和が、互いに他方と等しくなるもののことである。
 - 220の約数の和が284で、284の約数の和が220となるためこの2つは友愛数である
- ここでまず自分自身を除いた約数の和を求める関数 `calcSumYakusu` を作成せよ。引数は「自分自身を除いた約数の和」を求めたい数、戻り値を自分自身を除いた約数の和とせよ。
- 次に、作成した `calcSumYakusu` 関数を利用して10万までの友愛数のペアを右図のように表示するプログラムを作れ。
 - 関数の定義「`int calcSumYakusu(int num);`」
 - 適切にカッコやカンマ、スペース、改行を入れること
 - 左側の数値が小さくなるようにせよ
- 1分以内にすべての結果が表示されるようにせよ。

```
(220, 284)
(1184, 1210)
(2620, 2924)
(5020, 5564)
(6232, 6368)
(10744, 10856)
(12285, 14595)
(17296, 18416)
(63020, 76084)
(66928, 66992)
```

プログラミング演習I (第10回) 課題

• 発展① スケッチ名 : advanced_Collatz

- 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。まだ本予想は証明されていないが、かなり大きな数まで反例がないことが確かめられている。
- ルールは下記のとおりである
 - 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になるまで上記を繰り返す
- 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下のように標準出力せよ。また、その最後にステップ数も表示せよ。

```
[2]->1 (1 step)
[3]->10->5->16->8->4->2->1 (7 steps)
[4]->2->1 (2 steps)
[5]->16->8->4->2->1 (5 steps)
[6]->3->10->5->16->8->4->2->1 (8 steps)
[7]->22->11->34->17->52->26->13->40->20->10->5->16->8->4->2->1 (16 steps)
[8]->4->2->1 (3 steps)
[9]->28->14->7->22->11->34->17->52->26->13->40->20->10->5->16->8->4->2->1 (19 steps)
[10]->5->16->8->4->2->1 (6 steps)
[11]->34->17->52->26->13->40->20->10->5->16->8->4->2->1 (14 steps)
[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->10->5->16->8->4->2->1 (17 steps)
[15]->46->23->70->35->106->53->160->80->40->20->10->5->16->8->4->2->1 (17 steps)
[16]->8->4->2->1 (4 steps)
```

プログラミング演習I (第10回) 課題

• 発展②スケッチ名 : advanced_CalcPI

- フーリエ級数展開を用いると, 下記の式を用いて円周率の近似値を求めることが可能である

$$\sum_{k=0}^N \frac{(-1)^k}{2k+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

- このNをいくつまで指定するかというのを引数とし, 上記の式に4を掛けた値を返り値として, double型で返す関数 `calcPI` を作成せよ

- 関数の定義 : `double calcPI(int N);`

- また, `calcPI` の引数を10, 100, 1000, 10000, 100000, 1000000, 10000000としていった時の結果 (円周率の近似値) を標準出力せよ

```
N= 10 3.2323157489299774
N= 100 3.151493337005377
N= 1000 3.141592653589793
N= 10000 3.141592653589793
N= 100000 3.141592653589793
N= 1000000 3.141592653589793
N=10000000 3.141592653589793
```