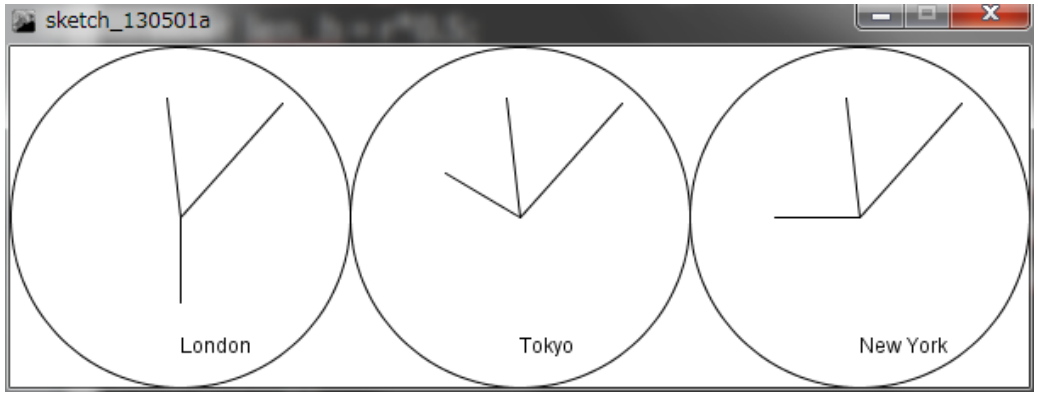
# プログラミング演習I （第10回） 課題

- **基本①　スケッチ名： basic_AnalogueClock**
  - LondonとTokyoとNew Yorkの時計を表示するアナログ時計を横に並べてみましょう（時差は各自チェックしましょう！）
  - アナログ時計を描く関数（ただし，この関数は下記のように整数値で中心座標(x,y)，半径(r)，時分秒(h,m,s)と，文字列で場所(location)を引数とし，時差については引数で加算減算せよ）を作成し，その関数を利用して3つの時計を描画するようにせよ

```
void analogueClock(int x, int y, int r, int h, int m, int s, String location);
```

```
// 中心(200,200)に半径80で，時差がないTokyoのアナログ時計を表示する場合
analogueClock(200, 200, 80, hour(), minute(), second(), "Tokyo" );
// 中心(600,200)に半径60で，時差1時間のSydneyのアナログ時計を表示する場合
analogueClock(150, 200, 60, hour()+1, minute(), second(), "Sydney" );
```



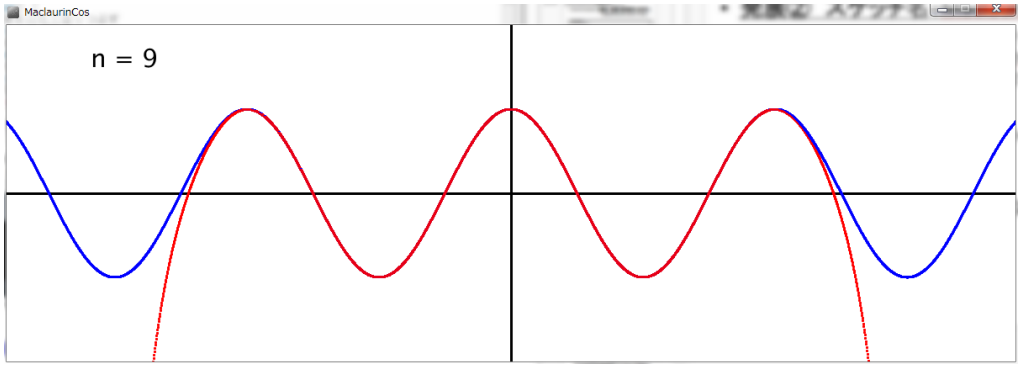
# プログラミング演習I（第10回） 課題

- **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、cosxと近似できることを確認せよ。

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

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

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



# プログラミング演習I （第10回） 課題

- **基本③　スケッチ名：basic_LifeGame**
  - 誕生、生存、過疎、過密によってセルが生まれたり死んだりするライフゲームを関数で作ろう。
  - ライフゲームでは、対象とするセルの周囲8マスが生きているか死んでるかを数え、その結果に応じて次のターンで、対象となるセルを生きている状態にするか、死んでいる状態にするかを決めるものである（次ページ参照)。
  - このライフゲームでは、セルが生きている場合は緑色の四角形を、死んでいる場合は黒色の四角形を描画することで表現する。
  - 配布する basic_LifeGame.pde の checkNextDeadOrAlive 関数は、その判定を行うものである。この関数を完成させてライフログゲームを完成させよ
  - ただし，checkNextDeadOrAlive 関数は，調べたいセルの(x, y)座標を引数とし，関数の返り値として，次の状態が「生」の場合は1，「死」の場合は0を返すものとせよ
  - 下記URLの安定状態が幾つか観測されたら成功

    http://ja.wikipedia.org/wiki/ライフゲーム

# プログラミング演習I （第10回） 課題

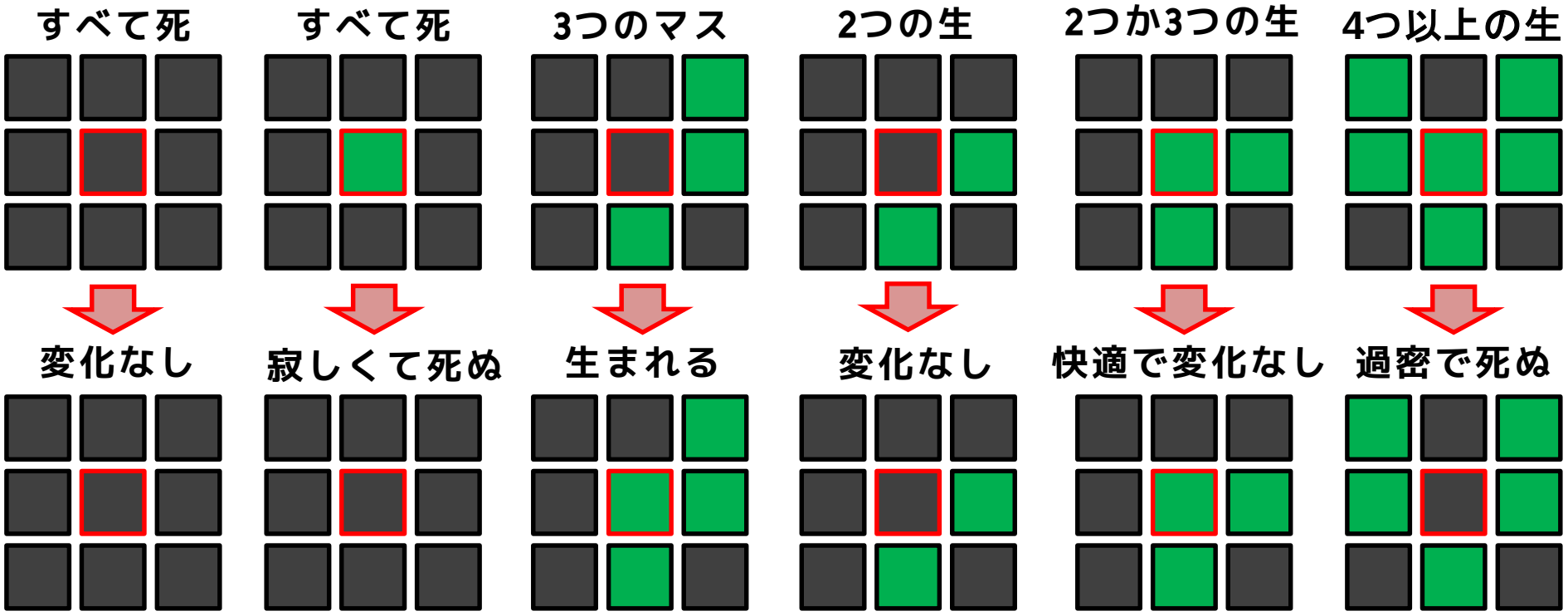
あるマス（赤フレーム）の縦・横・斜めの8マスの生死の状態（生の数）に注目する

【誕生】 死んでいるセルに隣接する生きたセルがちょうど3つならば次世代が誕生

【生存】 生きているセルに隣接する生きたセルが2つか3つならば次世代でも生存

【過疎】 生きているセルに隣接する生きたセルが1つ以下ならば過疎により死滅

【過密】 生きているセルに隣接する生きたセルが4つ以上ならば過密により死滅

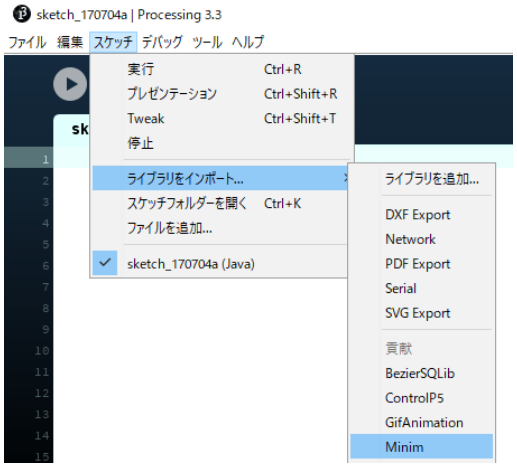


# プログラミング演習I （第10回） 課題

- **発展①　スケッチ名：advanced_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
  - 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。音はある程度聞こえていればOK。

# 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

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