

プログラミング演習I (第11回) 課題

• 基本① basic_AvgMaxMinFunc

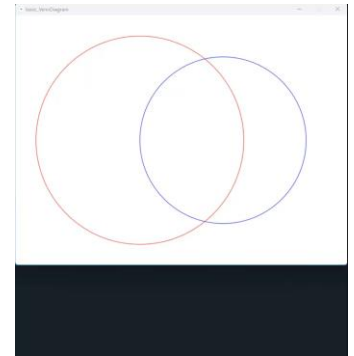
- 起動時に145cmから200cmまでの10人の身長をランダムに取得し、その値をまずは「#1: 173.24161cm」のように、半角シャープ、1から始まる数字、半角コロン、半角スペース、身長の数値、cmという形式で表示せよ。
- 次に、実数値の配列を引数とし、最大値を求めて返すgetMax関数、最小値を求めて返すgetMin関数、平均を求めて返すgetAvg関数を作成し、その関数を利用して最大値、最小値、平均の値を求め、右図のように標準出力せよ。
 - 「cm」表示を忘れないように！
 - 関数内でMax is ... cmと出力しないこと。
- なお、10人の身長をランダムに取得する関数、10人の身長の数値を出力する関数など、他の関数を作って利用しても良い。

```
#1: 173.24161cm
#2: 180.37756cm
#3: 173.16216cm
#4: 182.3948cm
#5: 177.75885cm
#6: 194.30447cm
#7: 148.29167cm
#8: 171.66035cm
#9: 151.28186cm
#10: 153.73384cm
Max is 194.30447cm
Min is 148.29167cm
Avg is 170.62073cm
```

プログラミング演習I (第11回) 課題

• 基本課題② basic_VennDiagram

- 800x600の大きさのウィンドウの背景を白色にし、そのウィンドウに、中心が(300, 300)で直径が500ピクセルの赤枠の円A、中心が(500, 300)で直径が400ピクセルの青枠の円Bを描け。
- 次に、ある座標(x, y)が与えられたときに、その座標が「A only」なら1、「B only」なら2、「A and B」なら3、「not A or B」なら0を返す関数 `judgeAB(int x, int y)` を作成せよ。
- また、この関数を用い、画面上の全ての点について、「A only」なら赤、「B only」なら青、「A and B」なら緑、「not A or B」なら灰色の点を描画せよ。
- また、最後にそれぞれの領域 (A only, B only, A and B, not A or B) 内の点の数を標準出力せよ (この数が面積の近似値となる)
- 最初の丸が表示されず一気に塗りつぶされていても良いが、何度も標準出力しないようにせよ
 - `setup()` だけで実現すると簡単でよいよ!



プログラミング演習I (第11回) 課題

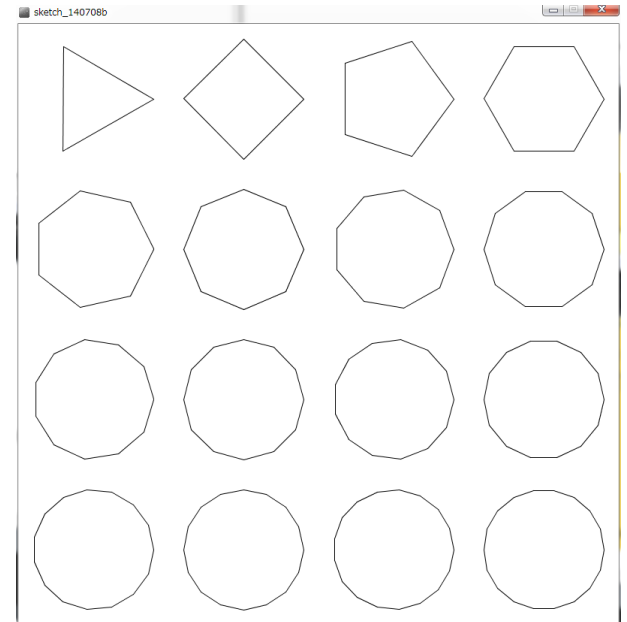
• 基本③ スケッチ名：basic_TsuruKame

- 配布したプログラムを利用して鶴亀算を完成させよ。
- 「鶴は頭が1つ足が2本」「亀は頭が1つ足が4本」である。
- ここで、頭の数と足の数関数の引数として与えたときに、鶴が何羽いるのかを計算して返す
getNumberOfCranes関数を作成せよ
- また、そういった鶴と亀の条件が存在しないときには
-1を返すようにせよ。
- 全てのテストに通るように頑張りましょう！

プログラミング演習I (第11回) 課題

• 発展① advanced_RegularPolygon

- 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 `drawRegularPolygon` という関数を作成せよ。
- また、その `drawRegularPolygon` という関数を利用して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。



プログラミング演習I (第11回) 課題

• 発展② advanced_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マスが活着しているか死んでるか数を数え、その結果に応じて次のターンで、対象となるセルを活着している状態にするか、死んでいる状態にするかを定める。
- このライフゲームでは、drawLifeGame関数で、セルが活着している場合(1のとき)は緑色の四角形を、死んでいる場合(0のとき)は黒色の四角形を描画することで表現する。
- また、checkNextDeadOrAlive 関数は、次のターンにそのセルが活着しているかどうかの判定を行うものであり、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。
- さらに、getCountOfAlive関数は、現在活着しているセルの数を返す関数である。
- checkNextDeadOrAliveとgetCountOfAliveを完成させてライフゲームを完成させよ

プログラミング演習I (第11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

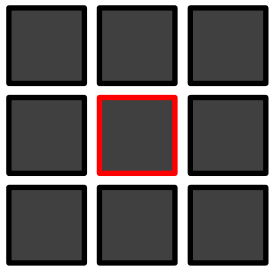
【生存】 生きているセルに隣接する生きたセルが2つか3つならば次世代でも生存

【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

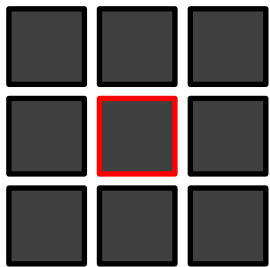
【過密】 生きているセルに隣接する生きたセルが4つ以上ならば過密により死滅

上記以外の場合は現状維持とする

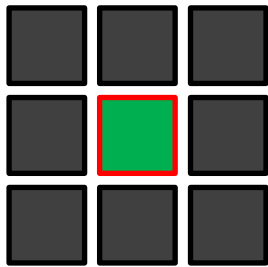
すべて死



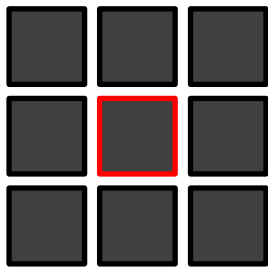
変化なし



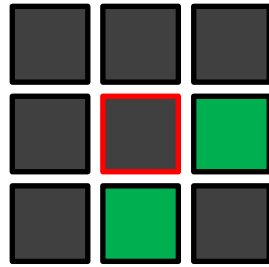
すべて死



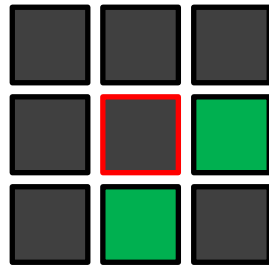
寂しくて死ぬ



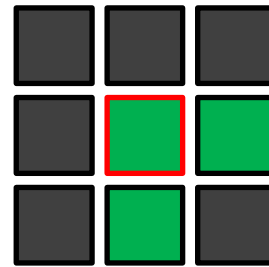
2つの生



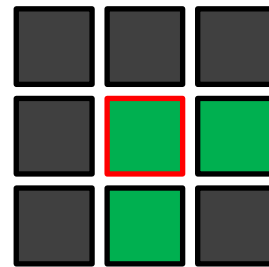
変化なし



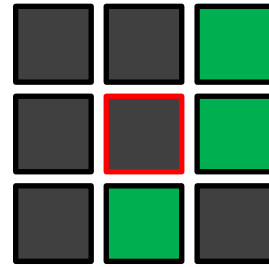
2つの生



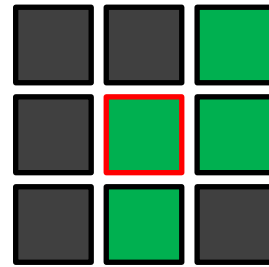
快適で変化なし



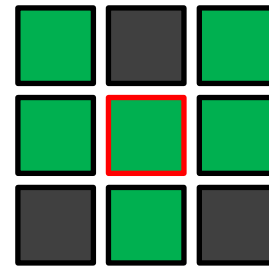
3つの生



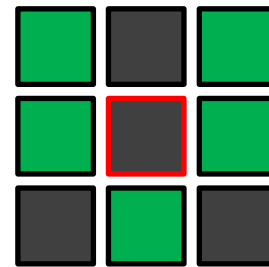
生まれる



5つの生

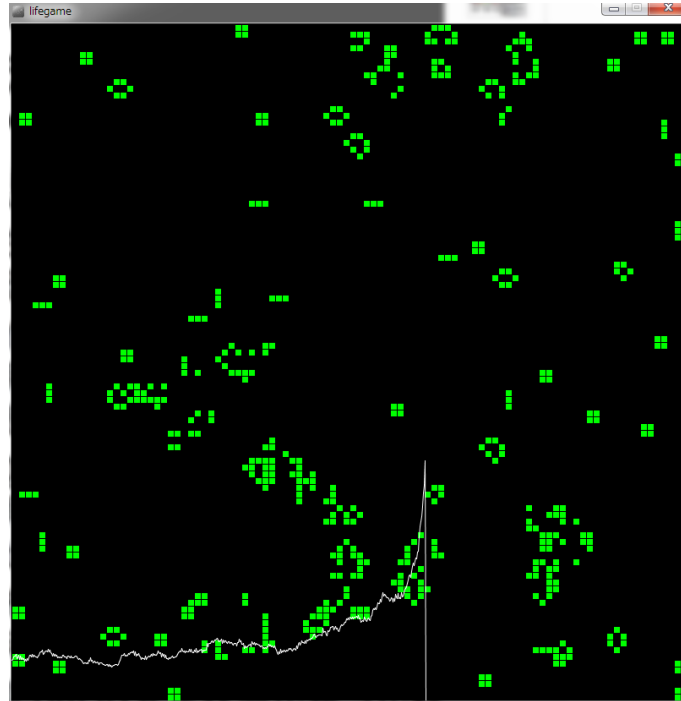
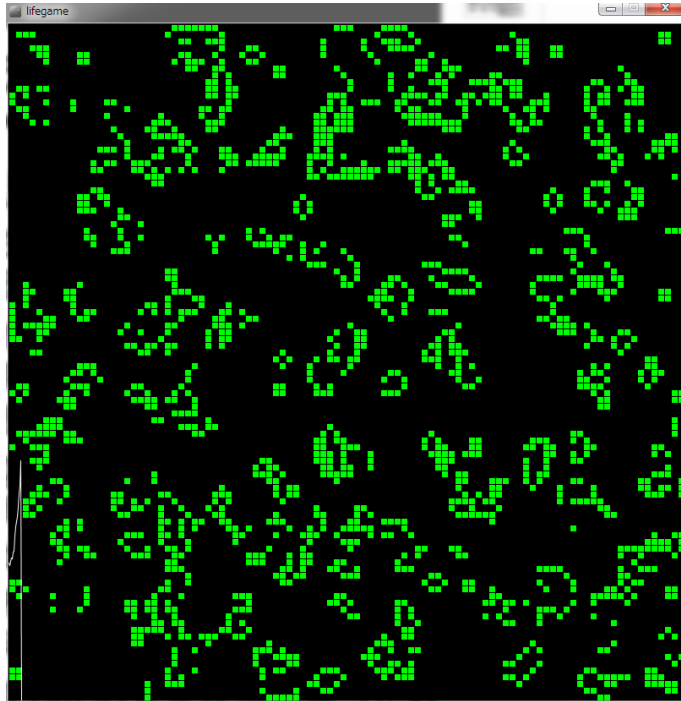


過密で死ぬ



プログラミング演習I (第11回) 課題

- ライフゲームとはこんなもの
 - ある種の生命のシミュレーション
 - 誕生、生存、過疎、過密で生死を繰り返す



下記動画も面白いので参考までに
ライフゲームの世界 <http://www.nicovideo.jp/mylist/34610498>