

小テスト補足

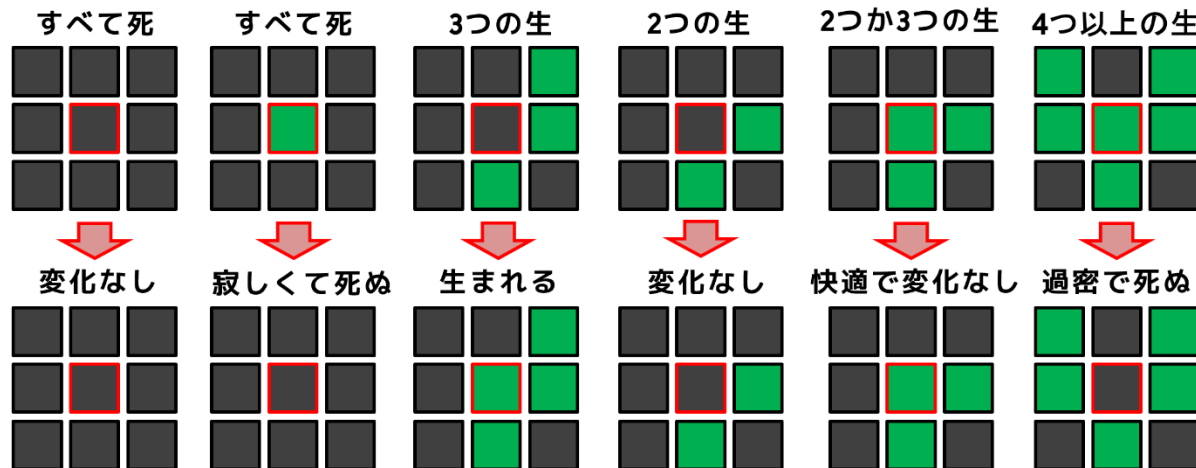
- 小テスト前（順を追って実施します）
 - (1) 他のすべてのアプリを終了せよ
 - (2) 資料配布フォルダの **11_exam** フォルダをデスクトップにコピーせよ
 - (3) フォルダ名を、年-組-番号に変更せよ
 - (4) examAとexamBをProcessingで起動せよ
 - (5) examAに名前と年組番号を記入してキーボードから手を離し、保存せよ
 - (6) examAの名前と年組番号をマウスでコピーせよ
 - (7) examBに作成者と年組番号をマウスでペーストし、プログラムを保存せよ
- 小テスト中
 - 各プログラムは、毎回実行する前には必ず保存するようにせよ
 - 不正が疑われるような行動はしないようにせよ
 - ファイルアクセスの履歴はすべて記録されていますし、前方監視しています
 - 小テスト終了までにプログラムを保存しProcessingを終了せよ
- 小テスト後
 - 指定の時間までに年-組-番号フォルダを提出せよ。その際、キーボードには触れないようにせよ
 - 他人のフォルダに提出しないように注意せよ
 - 保存されておらず中身の無いファイルが提出されている事もありますので注意

第11回 小テスト: examA

- 誕生、生存、過疎、過密によってセルが生き死にするライフゲームを checkNextDeadOrAlive 関数を下記のルールに従い完成させることで作れ
 - 対象とするセルの周囲8マスが生きているか死んでいるかを数え、その結果に応じて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切り替える。
 - 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

あるマス (赤枠) 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て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てきたセルが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てきたセルが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てきたセルが4つ以上ならば過密により死滅



第11回 小テスト: examB

- 引数として入力された年, 月 (ともに整数値) から, その月の最終日を整数値として返す関数 `getLastDay` を作成せよ
 - 1, 3, 5, 7, 8, 10, 12月は31日, 4, 6, 9, 11月は30日が最終日である
 - 2月は通常28日が最終日だが, 年が4で割り切れる場合に29日が最終日となり, さらに100で割り切れる場合に28日が最終日となり, さらに400で割り切れる場合に29日が最終日となる
 - 下記の `setup` をプログラム内に入力し, 標準出力を確認せよ
 - ***年*月の最終日は~という文字列が検証のため増えているのに注意**

```
void setup()
{
  for(int i=1; i<=12; i++)
  {
    println("2018年" + i + "月の最終日は" + getLastDay(2018, i) );
  }
  for(int i=1990; i<=2010; i++)
  {
    println(i + "年2月の最終日は" + getLastDay(i, 2) );
  }
  for(int i=0; i<=2100; i+= 100)
  {
    println(i + "年2月の最終日は" + getLastDay(i, 2) );
  }
}
```

プログラミング演習I (第11回)

この問題を出しますが、頭の中で考えるだけにし、実際にプログラムはしないでくださいね

• 発展② スケッチ名：LangtonsAnt

- ラングトンのアリは、黒と白のマス目（縦横4px）が並んだ広い平面（ウィンドウサイズは800x800px）の中に棲んでおり、アリは東西南北のいずれかの方向を向いている。そして、時計の針が1つ進むと（次の描画で）1マス進んで次のルールに従い行動する
 - **黒いマス目に入ったら左に90度回転する**
 - **白いマス目に入ったら右に90度回転する**
 - **それまでアリがいたセルは白から黒、黒から白に色を変える**
- このとき、最初の黒色がどのようなパターンであっても、最終的にハイウェイと呼ばれる道ができ、アリが脱出していくというものである（必ずハイウェイが出来て逃げるかどうかというのは未解決問題）
- 上記プログラムを完成させて、ラングトンのアリが脱出する様子を観測せよ（脱出したときに配列エラーを起こすのは許容する）

