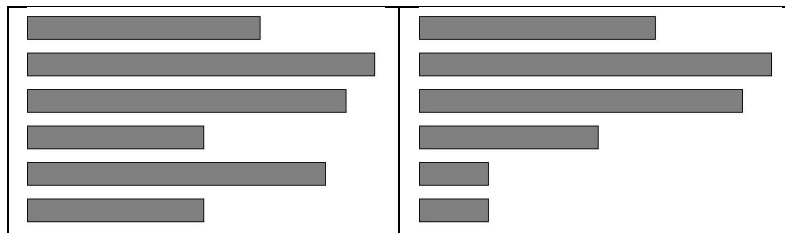


2013 年度情報メディア基盤ユニット 7 月 5 日分課題と宿題

授業関連資料は <http://www.sato-lab.jp/imfu2012> からダウンロード出来ます。授業中に配布したプリントに誤りを見つけた際には、修正版をのせてあります。問 2 以降は、出来たら先生か TA の人に確認をしてもらって下さい。【自己確認問題】は先生や TA の人にチェックしてもらう必要はありません。出来たら先生か TA の人に確認をもらって下さい。問題は難易度順に並んでいるわけではありません。

1. 【自己確認】本文中のサンプルプログラムを実行して見て下さい。
2. 次のプログラムは、乱数で決めた 4 つの値とマウスの X 座標の値に応じて、5 つの四角形と、この 5 つの値の中で最小値を利用して描かれた四角形を描くプログラムです。つまり、一番下の四角形の長さが、常に一番短いものとなっています。空欄を埋めて、プログラムを完成させて下さい。



未完成プログラム

```
float[] x = new float[ (a) ];
void setup(){
  size(400,250);
  for( (b) i=0;i<x.length-1;i++){
    x[i] = random(0.3*width,width);
  }
}
void draw(){
  background(255);
  stroke(0);
  fill(128);
  x[x. (c) - (d) ] = mouseX;
  float m = myMin(x);
  for(int i=0;i< (e) ;i++){
    rect(0,40*i+10,x[i],25);
  }
  rect(0,40* (f) +10,m,25);
}
float myMin( (g) x){
  float tentativeMin = x[0];
  for(int i=1;i<x. (h) ;i++){
    if(tentativeMin (i) x[i]){
      tentativeMin = (j) ;
    }
  }
}
```

```
return __ (k) __ ;
}
```

3. 【目コピ問題】 次のプログラムは、乱数で決めた 4 つの値とマウスの X 座標を用いて 5 つの四角形を描くものです。この 5 つの値の中で最小値を利用して描かれた四角形の色は赤色とします。空欄を埋めて、プログラムを完成させて下さい。



未完成プログラム

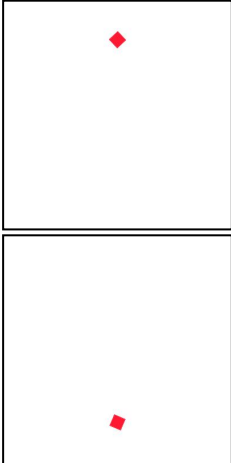
```
float[] y = new float[__(a)__);

void setup() {
  size(400, 200);
  for (int i=0;i<__(b)__;i++) {
    y[i] = random(0.3*width, width);
  }
}

void draw() {
  background(255);
  stroke(0);
  y[__(c) ] = mouseX;
  int minPos = findMinPos(__(d)__);
  for (int j=0;j<__(e)__;j++) {
    fill(128);
    if (__(f) == minPos) {
      fill(__(g) , __(h) , __(i) );
    }
    rect(0, 40*j+10, y[j], 25);
  }
}

int findMinPos(float[] x) {
  int tentativePos = __(j)__;
  for (int i=1;i<__(k)__;i++) {
    if (__(l) > x[i]) {
      tentativePos = i;
    }
  }
  return __(m) ;
}
```

4. 【目コピ問題】 次のプログラムは、1 個の正方形が回転しながら上から下に移動するようなプログラムです。空欄を埋めて、プログラムを完成させて下さい。また、式「rot = rot+PI/90;」の 90 の値を変更するとどのような変化が起きるか説明して下さい。

未完成のプログラム	実行例
<pre> float y; // 正方形の中心のY座標値 float rot; // 正方形の回転角度 (ラジアン) void setup(){ size(400,400); smooth(); y = 0; rot = 0; } void draw(){ background(255); stroke(255,25,50); fill(255,25,50); y = y+1; rot = rot+PI/90; pushMatrix(); translate((a) , (b)); rotate((c)); rectMode((d)); rect(0,0,20,20); popMatrix(); } </pre>	

5. 【目コピ問題】 次のプログラムは問 4 のプログラムを変更して作成したものです。正方形が一番下まで移動したら、上に戻り、再び下のほうに移動し、さらに下から上に戻る際に、落下する速さを乱数で変更しています。

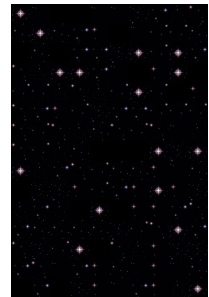
未完成のプログラム
<pre> float y; // 正方形の中心のY座標値 float rot; // 正方形の回転角度 (ラジアン) float v; // 正方形の移動速度 void setup(){ size(400,400); smooth(); y = 0; rot = 0; v = random(0.5,1.5); } void draw(){ background(255); stroke(255,25,50); fill(255,25,50); rot = rot+PI/90; y = y+ (a) ; if((b)){ y = 0; </pre>

```

    v = (c) ;
}
pushMatrix();
translate( (d) , (e) );
rotate( (g) );
rectMode( (g) );
rect(0,0,20,20);
popMatrix();
}

```

6. 【目コピ問題】 次の未完成プログラムは、10 枚の 32X32 の大きさの PNG ファイルをランダムに表示することで、宇宙空間のような表示結果を得るプログラムです。読み込む画像ファイル名は space0.png～space9.png となっており、bgID の各要素にはどの画像ファイルを表示するかの示す 0～9 までの数字のどれかが入っている。なお、画像ファイルは <http://www.sato-lab.jp/imfu2013/sampledata> に入っている。



未完成のプログラム

```

PImage[] bgImages; // 画像情報
int[][] bgID; // どの画像を表示するかを決めている配列
void setupBackground(){
    bgImages = new PImage[10];
    for(int i=0;i < 10;i++){
        bgImages[i] = loadImage("space"+i+".png");
    }
    bgID = new int[ (a) ][ (b) ];
    for(int i=0;i< (c) ;i++){
        for(int j=0;j< (d) ;j++){
            bgID[i][j] = (e) (random( (f) ));
        }
    }
}
void setup(){
    size(320,480);
    setupBackground();
}

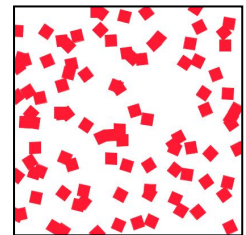
void draw(){
    background(255);
    for(int x=0;x< (g) ;x++){
        for(int y=0;y< (h) ;y++){
            image(bgImages[bgID[ (i) ][ (j) ]], (k) *x, (l) *y);
        }
    }
}
}

```

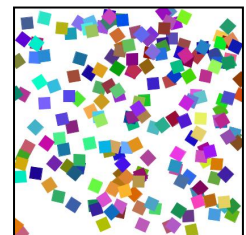
7. 【目コピ問題】 次の未完成プログラムは、問 6 のプログラムに、マウスをクリックしたら、一下に表示されている画像が一番上に移動し、残りのものは一段ずつ下に移動するようなものです。Setup 関数や draw 関数には変更はありません。問 6 に void mouseClicked 関数を付け加えることで完成させて下さい。

未完成のプログラム
<pre> void mouseClicked(){ int[] tmp = new int[(a)]; for(int i=0;i< (b) ;i++){ tmp[i] = bgID[i][(c)]; } for(int y= (d) ;y> (e) ;y--){ for(int x=0;x< (f) ;x++){ bgID[x][(g)] = bgID[x][(h)]; } } for(int x=0;x< (i) ;x++){ bgID[x][(j)] = (k) ; } } </pre>

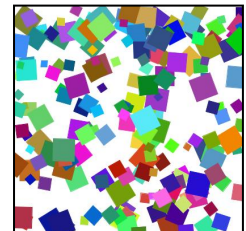
8. 【目コピ問題】 問 5 のプログラムを変更して、沢山 (100 個以上) の四角形が回転しながら上から下に移動するようなプログラムを作成して下さい。



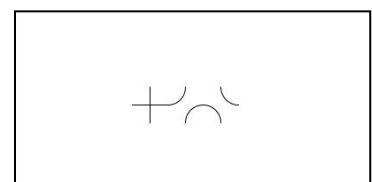
9. 【目コピ問題】 問 8 のプログラムを変更して、描く正方形の色を乱数で決めるように変更したものを作成してください。



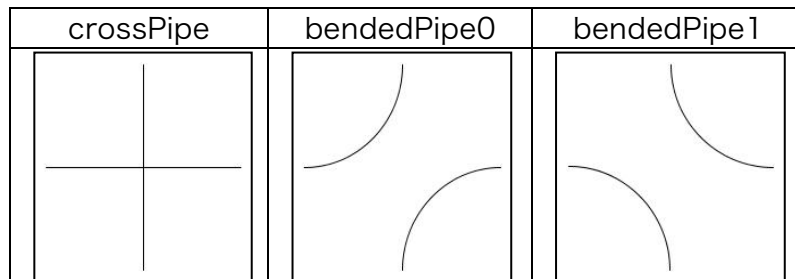
10. 【目コピ問題】 問 9 のプログラムを変更して、描く正方形の辺の長さを乱数で決めるように変更したものを作成してください。



11. 【目コピ問題】 次のプログラムは、十字のような形や 2 つの四分の 1 円を組み合わせて描くものである。関数 crossPipe, bendedPipe0, bendedPipe1 は以下のような形を

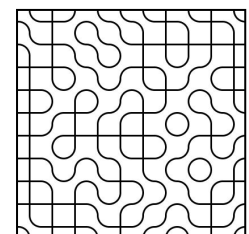


描く関数である。プログラム中の空欄を埋めて、プログラムを完成させてください。なお、関数 `bendedPipe1` は、自分で関数を完成させて下さい。



未完成プログラム
<pre> (a) crossPipe(float x,float y,float l){ (b) (x+l/2,y,x+l/2,y+l); (c) (x,y+l/2,x+l,y+l/2); } (d) bendedPipe0(float x,float y,float l){ arc(x,y,l,l,0,HALF_PI); (e) (x+l,y+l,l,l, (f) , (g)); } void bendedPipe1(float x,float y,float l){ // この関数は自力で完成させ下さい。 } void setup(){ size(400,200); smooth(); } void draw(){ background(255); stroke(0); crossPipe(mouseX,mouseY,40); bendedPipe0(mouseX+1*40,mouseY,40); bendedPipe1(mouseX+2*40,mouseY,40); } </pre>

12. 【目コピ問題】 次のプログラムは、問 11 で作成した 3 つの関数を利用して作成したものです。int 型の 2 次元配列 `board` を宣言し、この変数に 0~2 までの int 型の乱数を記憶させます。そして、要素の値が 0 なら `crossPipe`、1 なら `bendedPipe0`、2 なら `bendedPipe1` を使って、描画を行います。空欄を埋めて、プログラムを完成させてください。



未完成プログラム
<pre> int boardSize = 10; (a) board = new int[boardSize][boardSize]; </pre>

```

    (b) crossPipe(float x,float y,float l){
        (c) (x+l/2,y,x+l/2,y+l);
        (d) (x,y+l/2,x+l,y+l/2);
    }
    (e) bendedPipe0(float x,float y,float l){
        arc(x,y,l,l,0,HALF_PI);
        (f) (x+l,y+l,l,l, (g) , (h) );
    }

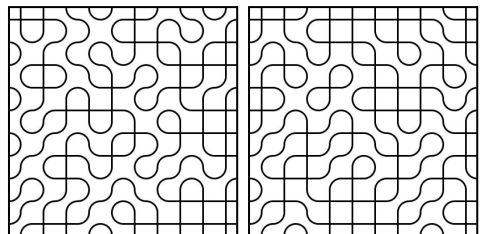
void bendedPipe1(float x,float y,float l){
    // この関数は自力で完成させ下さい。
}

void setup(){
    size(400,400);
    smooth();
    for(int i=0;i<board.length;i++){
        for(int j=0;j<board[i].length;j++){
            board[i][j] = (i) ;
        }
    }
}

void draw(){
    background(255);
    stroke(0);
    strokeWeight(3);
    for(int i=0;i<board.length;i++){
        (j) y=40*i;
        for(int j=0;j<board[i].length;j++){
            (k) x=40*j;
            if(board[i][j]==0){
                (l) (x,y,40);
            }else if(board[i][j] == 1){
                (m) (x,y,40);
            }else{
                (n) (x,y,40);
            }
        }
    }
}
}
}

```

13. 【目コピ問題】 問 12 のプログラムを変更して、マウスをクリックすると、2次元配列 board の要素の値を乱数で変更し、表示が変化するようにしたものです。空欄を埋めて、プログラムを完成させてください。



未完成のプログラム
<pre> int boardSize = 10; (a) board = new int[boardSize][boardSize]; (b) crossPipe(float x,float y,float l){ </pre>

```

    (c) (x+l/2,y,x+l/2,y+l);
    (d) (x,y+l/2,x+l,y+l/2);
}
(e) bendedPipe0(float x,float y,float l){
    arc(x,y,l,l,0,HALF_PI);
    (f) (x+l,y+l,l,l, (g) , (h) );
}

void bendedPipe1(float x,float y,float l){
    // この関数は自力で完成させ下さい。
}

void setupBoard(int[][] b){
    for(int i=0;i<(i);i++){
        for(int j=0;j<(j);j++){
            b[i][j] = (k);
        }
    }
}

void setup(){
    size(400,400);
    smooth();
    setupBoard(board);
}

void draw(){
    background(255);
    stroke(0);
    strokeWeight(3);
    for(int i=0;i<board.length;i++){
        (l) y=40*i;
        for(int j=0;j<board[i].length;j++){
            (m) x=40*j;
            if(board[i][j]==0){
                (n) (x,y,40);
            }else if(board[i][j] == 1){
                (o) (x,y,40);
            }else{
                (p) (x,y,40);
            }
        }
    }
}

void mouseClicked(){
    (q) (board);
}

```

14. 【目コピ問題】 問 12 のプログラムを変更して、マウスをクリックすると、その場所の絵を crossPipe->bendedPipe0->bendedPipe1 の順に変更するようなプログラムとしてください。面倒なので、mouseClicked 関数の部分だけをのせています。

未完成のプログラム(mouseClicked 関数)その 1

```

void mouseClicked(){
    int x=mouseX/40;
    int y=mouseY/40;
    if(board[ (a) ][ (b) ] == 0){

```



```

board[ (a) ][ (b) ] = (c) ;
}else if(board[ (a) ][ (b) ] == 1){
board[ (a) ][ (b) ] = (d) ;
}else{
board[ (a) ][ (b) ] = (e) ;
}
}
}

```

未完成のプログラム(mouseClicked 関数)その 2

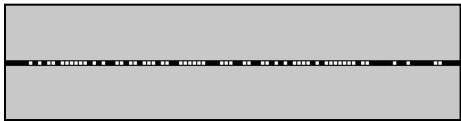
```

void mouseClicked(){
int x=mouseX/40;
int y=mouseY/40;
board[ (a) ][ (b) ] =(board[ (c) ][ (d) ] + (e) ) % (f) ;
}

```

15. 【目コピ問題】 int 型配列 cells の各要素には、0 または 1 という値が保存されている。値が 0 の時には細胞が死んでいる、値が 1 の時には細胞が生きていることを表しているとします。ある細胞 A が死んでいるとします。このとき、左隣の細胞が生きているときには、うれしくなり、次の世代では、細胞 A は生き返ります。そうでなければ、次の世代でも細胞 A は死んだままです。また、細胞 A が生きているとします。このときは、両隣の細胞が死んでいると、寂しくなり次の世代では死んでしまいます。また、両隣が生きていると、息苦しくなり次の世代では死んでしまいます。そうでなければ、次の世代でも細胞 A は生き続けます。配列の添え字番号が小さいものが左にあるとします。また、添え字番号が 0 の要素の左隣は配列の最後の要素とし、配列の最後の要素の右隣は添え字番号が 0 の要素とします。

次のプログラムは、このような規則のもとで、世代毎の表示を行っているプログラムです。生きている細胞は黒色の正方形、死んでいる細胞は白色の正方形で表しています。空欄を埋めて、プログラムを完成させてください。なお、初期世代(最初の cells の各要素の値)は乱数で決めるものとします。



世代変化の規則の矢印 (→) の左側は現在の細胞 A の周りの状況を表しています。赤字の部分が細胞 A です。矢印 (→) の右側は、次の世代での細胞 A の状況を示しています。

世代変化ルール

000→0	010→0
100→1	110→1
001→0	011→0
101→1	111→0 (ここが変更)

未完成のプログラム

```
int nCells = 100;
(a) cells = new int[nCells];
void setupCells(){
  for(int i=0;i<nCells;i++){
    cells[i] = int( (b) );
  }
}
void evolve(){
  int[] ng = new int[nCells];
  for(int i=0;i<nCells;i++){
    int left = i-1;
    int right = i+1;
    if(left == -1){
      left = (c) ;
    }
    if(right == nCells){
      right = (d) ;
    }
    if(cells[i] == 0){
      ng[i]= (e) ;
      if( (f) ){
        ng[i] = 1;
      }
    }else{
      ng[i] = 1;
      if( (g) ){
        ng[i] = 0;
      }else if( (h) ){
        ng[i] = 0;
      }
    }
  }
  for(int i=0;i<nCells;i++){
    cells[i] = ng[i];
  }
}
void display(int[] c,float y){
  stroke(0);
  for(int i=0;i<nCells;i++){
    if(c[i] == 1){
      fill( (i) );
    }else{
      fill( (j) );
    }
    rect(4*i,y,4,4);
  }
}
void setup(){
  size(400,100);
  smooth();
  setupCells();
}
void draw(){
  background(150);
```

```
    evolve();
    display(cells,height/2-4/2);
}
```

16. 次のプログラムは、問 15 のプログラム中の関数 evolve を次のように、剰余演算（%、余りを求める）を使ったもの書き換えたものです。空欄を埋めて、この関数を完成させてください。

未完成の evolve 関数
<pre>void evolve(){ int[] ng = new int[nCells]; for(int i=0;i<nCells;i++){ int left = (i-1+nCells) % (a) ; int right = (i+(b)) % (c) ; if(cells[i] == 0){ ng[i]=0; if(cells[left] == 1){ ng[i] = 1; } }else{ ng[i] = 1; if(cells[right] == 1 && cells[left] ==1){ ng[i] = 0; }else if(cells[right] == 0 && cells[left] == 0){ ng[i] = 0; } } } for(int i=0;i<nCells;i++){ cells[i] = ng[i]; } }</pre>

17. 武部さんは、問 15 のプログラム中の関数 evolve で配列 ng を使わなくても大丈夫だと思い、evolve 関数を次のように書き換えました。この evolve 関数を使って、プログラムを実行するとどうなりますか？なぜ不具合が起きるのかを説明してください。

未完成の evolve 関数
<pre>void evolve(){ for(int i=0;i<nCells;i++){ int left = i-1; int right =i+1; if(left == -1){ left = nCells-1; } if(right == nCells){ right = 0; } } }</pre>

```

if(cells[i] == 0){
    cells[i]=0;
    if(cells[left] == 1){
        cells[i] = 1;
    }
}else{
    cells[i] = 1;
    if(cells[right] == 1 && cells[left] ==1){
        cells[i] = 0;
    }else if(cells[right] == 0 && cells[left] == 0){
        cells[i] = 0;
    }
}
}
}
}

```

18. 【工夫問題】 問 15 の世代変化のルールを表を眺めていると、二進数が見てくるように思えます。つまり、下の表のようになります。そこで、二進数として見たときの値を添え字の番号として、配列に次世代の状態を保存することで evolve 関数を実装出来るように思えます。この方針で作ったものが、次のプログラムです。空欄を埋めて、関数を完成させてください。配列 rules に、世代変化ルールを保存しています。変数 rules の要素の値を変更して、プログラムを実行して見てください。各自、面白い変化をするルールを見つけてください。

世代変化ルール

0=000→0	2=010→0
4=100→1	6=110→1
1=001→0	3=011→0
5=101→1	7=111→0 (ここが変更)

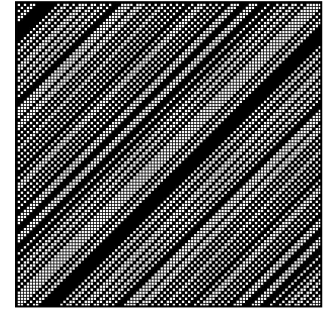
未完成の evolve 関数

```

void evolve(){
    int[] rules = {0, (a) , (b) , (c) , (d) , (e) , (f) , (g) };
    int[] ng = new int[nCells];
    for(int i=0;i<nCells;i++){
        int left = (i-1+nCells) % (h) ;
        int right = (i+(i) ) % (j) ;
        int v = (k) *cells[left]+ (l) *cells[i]+cells[right];
        ng[i] = rules[v];
    }
    for(int i=0;i<nCells;i++){
        cells[i] = ng[i];
    }
}
}

```

19. 【工夫問題】 問 15 のプログラムでは、1 世代分しか表示されない。過去の世代も表示されるようなプログラムを作成してください。



20. 【工夫問題】 問 10 のプログラムを改良し、遠近感が出るようにして下さい。たとえば、

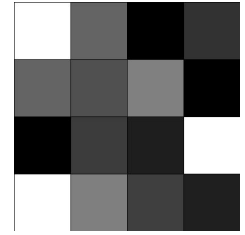
- 小さな多角形は遠くにあると考え、大きな多角形の前に表示されない、
- 小さな多角形の色は大きな多角形の色よりも薄くなる、
- 小さな多角形の移動速度は大きな多角形の移動速度よりも少し遅くなる。などの効果を追加してください。

21. 【工夫問題】 問 6 のプログラムを改良し、一ドットごとに背景がスクロールするようなプログラムを作成して下さい。

宿題

1. 次の表のよう、ウインドウを 4X4 の灰色に塗り分けるものです。空欄を埋めて、プログラムを完成させて下さい。

255	100	0	50
100	80	128	0
0	60	30	255
255	127	63	31



未完成のプログラム

```
int[][] gray = {{255, __(a)__, __(b)__, __(c)__),
                { __(d)__, __(e)__, __(f)__, __(g)__),
                { __(h)__, __(i)__, __(j)__, __(k)__),
                { __(l)__, __(m)__, __(n)__, 31}}};

void setup(){
  size(400,400);
}

void draw(){
  stroke(0);
  for(int i=0;i< __(o)__;i++){
    for(int j=0;j< __(p)__;j++){
      fill(gray[ __(q) ][ __(r) ]);
      rect(100* __(s)__,100* __(t)__,100,100);
    }
  }
}
```