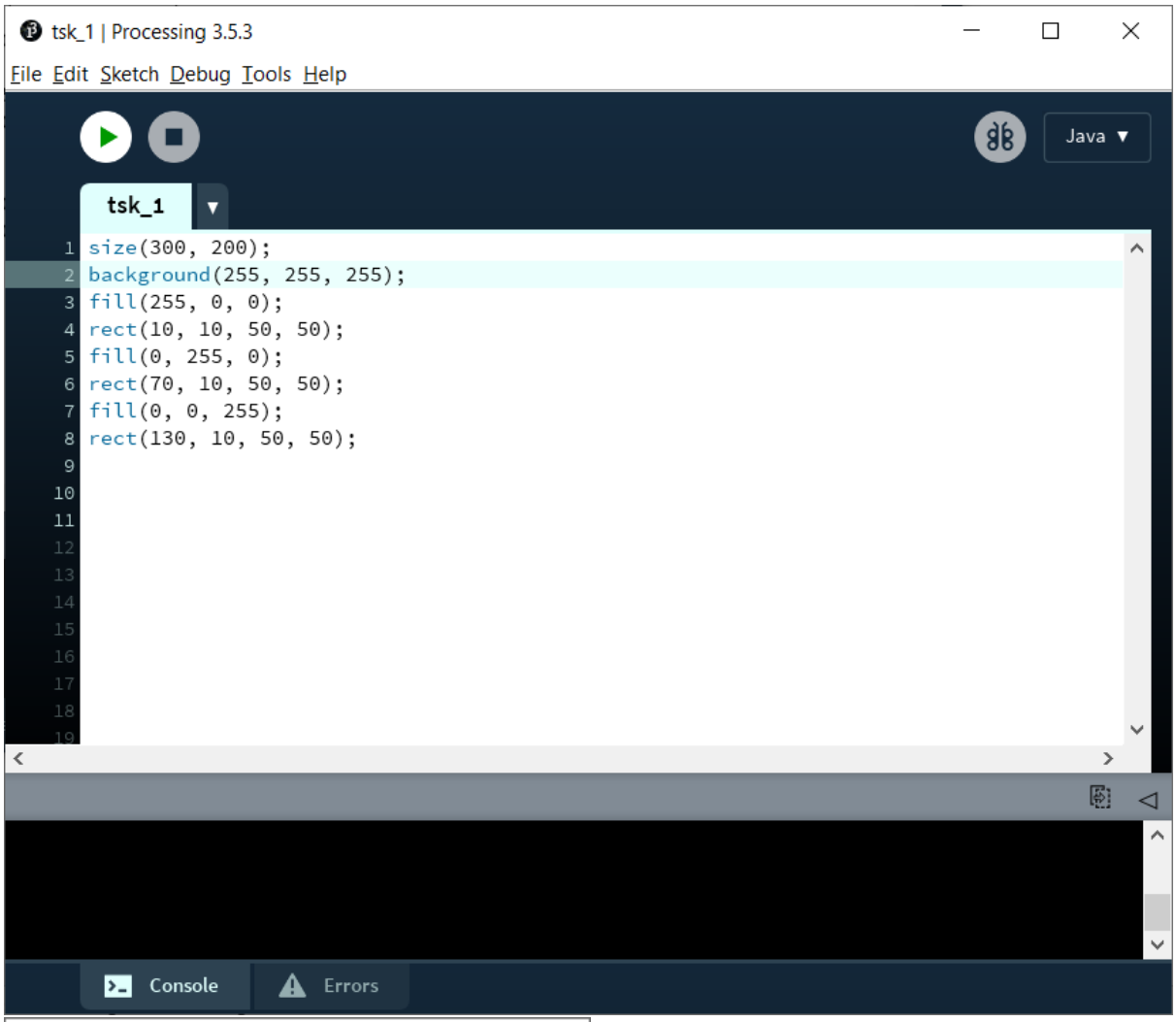
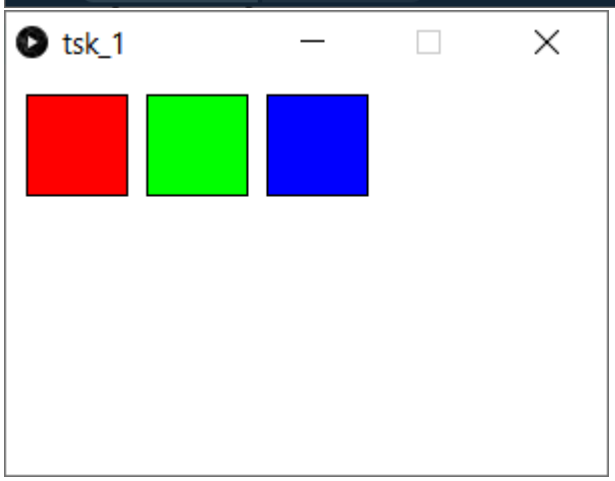
Lab 1

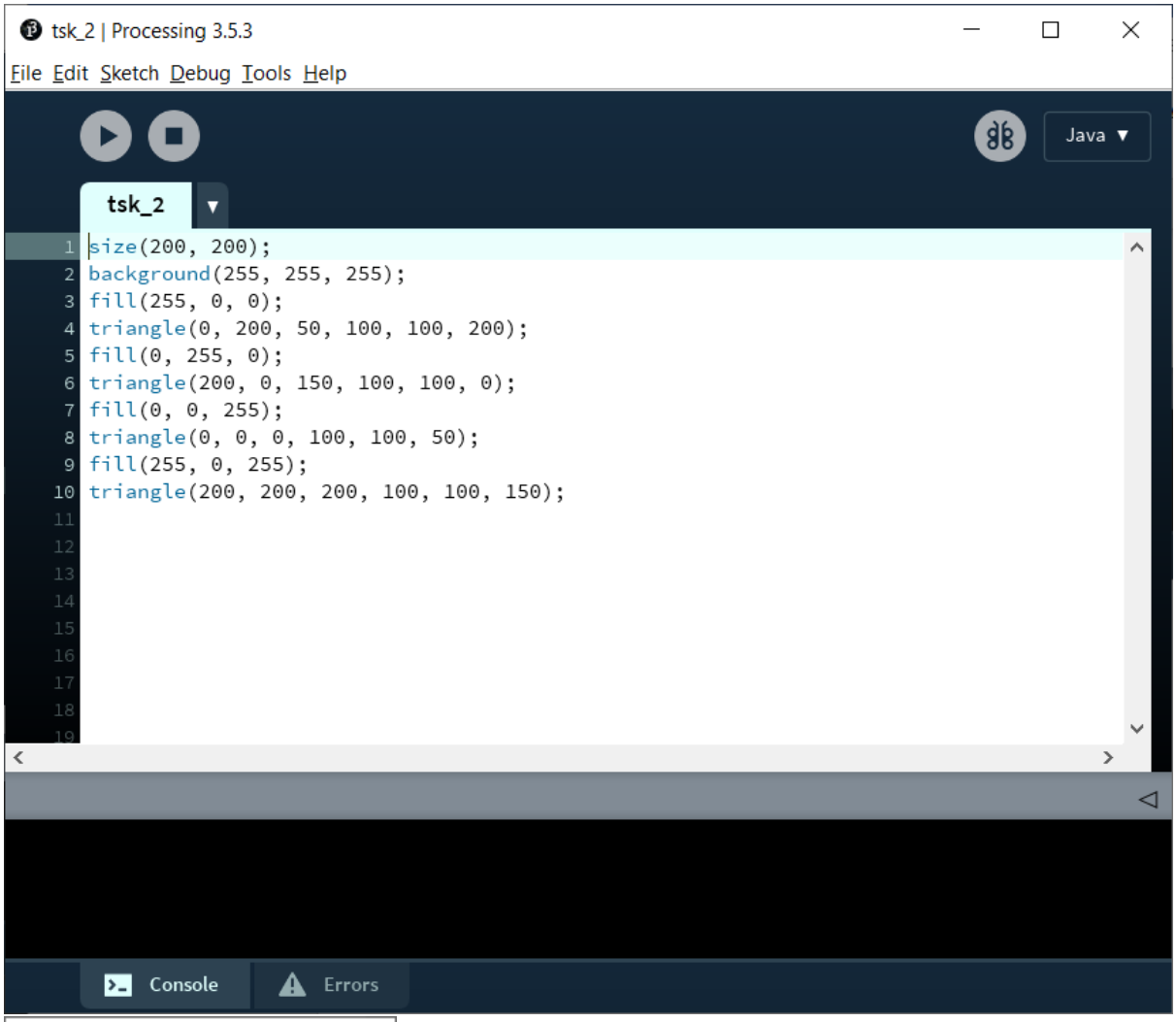
1.

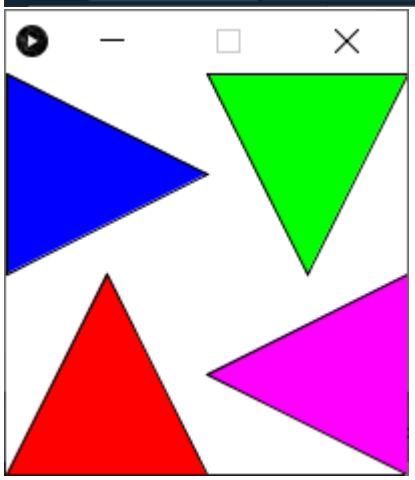

tsk_1 | Processing 3.5.3
File Edit Sketch Debug Tools Help
Java
tsk_1
size(300, 200);
background(255, 255, 255);
fill(255, 0, 0);
rect(10, 10, 50, 50);
fill(0, 255, 0);
rect(70, 10, 50, 50);
fill(0, 0, 255);
rect(130, 10, 50, 50);
Console
Errors


tsk_1

2.

2.1.


tsk_2 | Processing 3.5.3
File Edit Sketch Debug Tools Help
Java
tsk_2
size(200, 200);
background(255, 255, 255);
fill(255, 0, 0);
triangle(0, 200, 50, 100, 100, 200);
fill(0, 255, 0);
triangle(200, 0, 150, 100, 100, 0);
fill(0, 0, 255);
triangle(0, 0, 0, 100, 100, 50);
fill(255, 0, 255);
triangle(200, 200, 200, 100, 100, 150);
Console
Errors



2.2.

tsk_2_2 | Processing 3.5.3

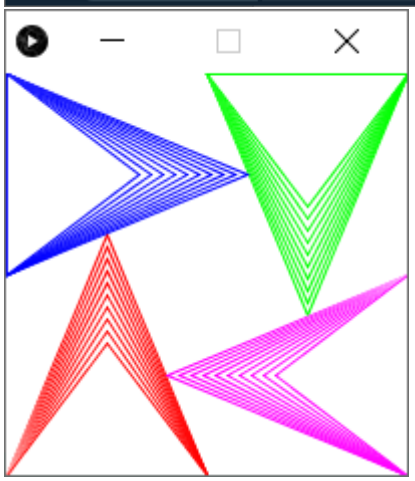
File Edit Sketch Debug Tools Help

Java

tsk_2_2

```
size(200, 200);
background(255);
stroke(255, 0, 0);
for (int i = 0; i < 10; i++) {
triangle(0, 200, 50, 80+(i*6), 100, 200);
}
stroke(0, 255, 0);
for (int i = 0; i < 10; i++) {
triangle(200, 0, 150, 120-(i*6), 100, 0);
}
stroke(0, 0, 255);
for (int i = 0; i < 10; i++) {
triangle(0, 0, 120-(i*6), 50, 0, 100);
}
stroke(255, 0, 255);
for (int i = 0; i < 10; i++) {
triangle(200, 200, 80+(i*6), 150, 200, 100);
}
```

Console Errors



3.

tsk_3 | Processing 3.5.3

File Edit Sketch Debug Tools Help

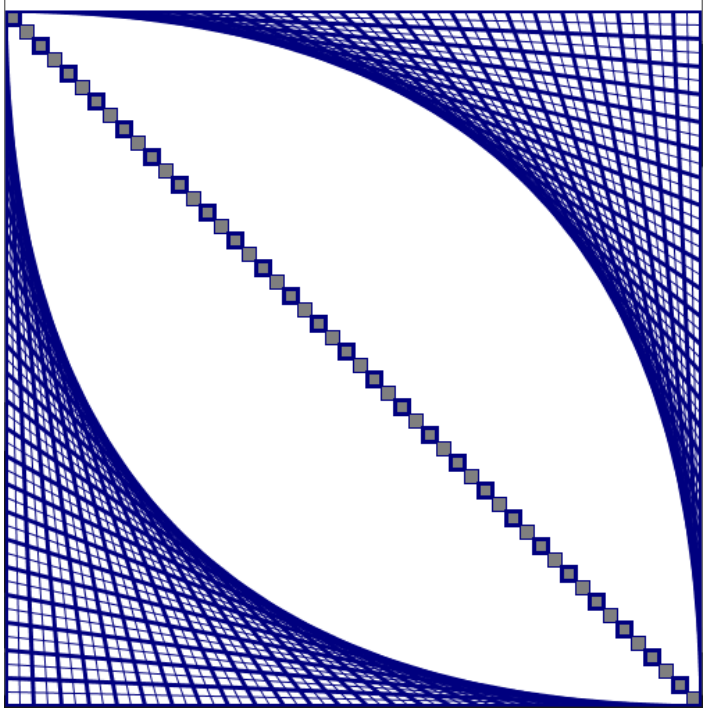
Java

tsk_3

```
size(501, 501);
background(255);
stroke(0, 0, 127);
fill(127);

for (int i = 0; i < 51; i++) {
strokeWeight(3);
line(0, i*10, i*10, 500);
line(i*10, 0, 500, i*10);
rect(i*10, i*10, 10, 10);

i++;
strokeWeight(1);
line(0, i*10, i*10, 500);
line(i*10, 0, 500, i*10);
rect(i*10, i*10, 10, 10);
}
```

Console Errors

tsk_3



Lab 2

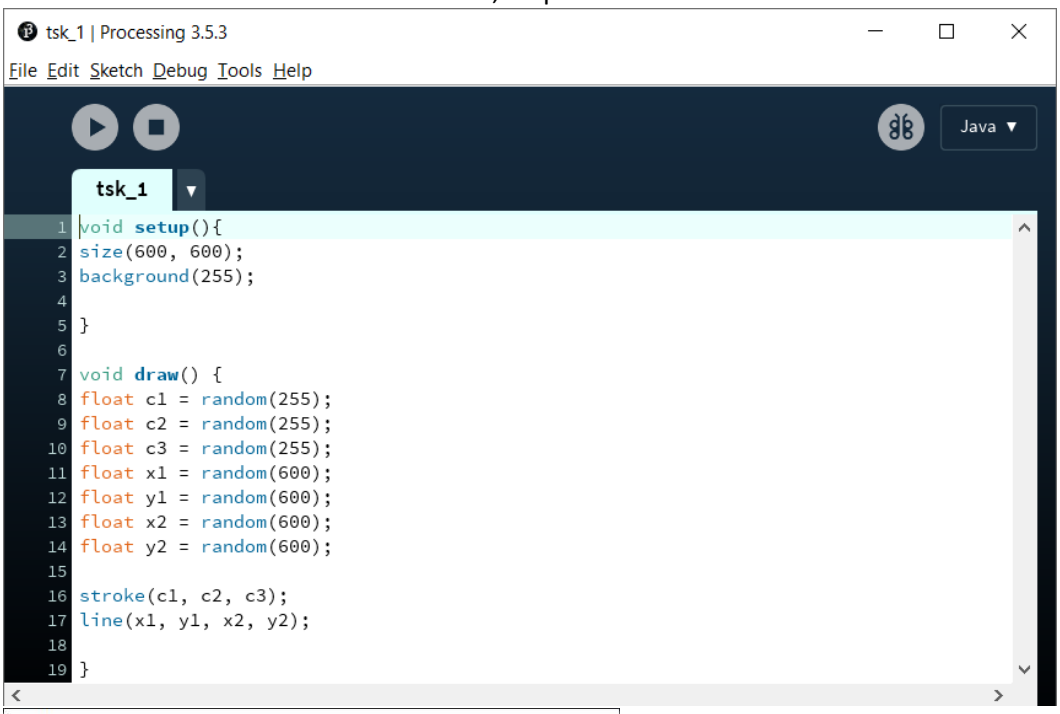
1.

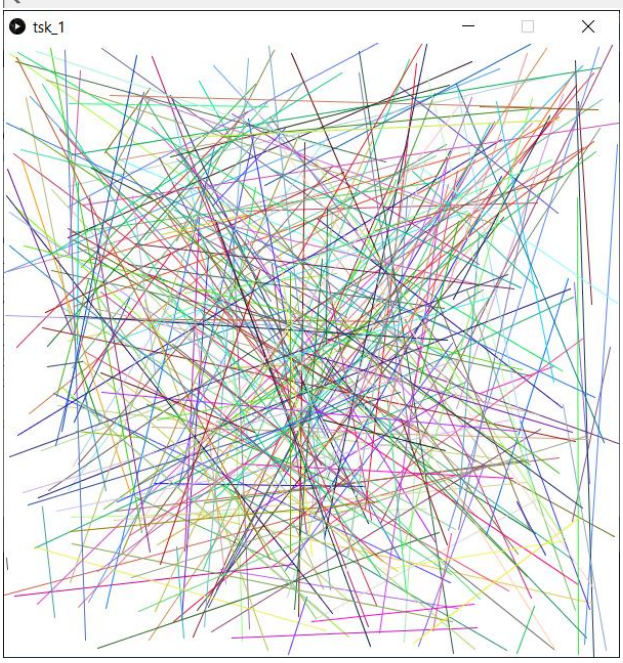
1.1.

1.1.1.It gradually updates the screen because the draw function is being called repeatedly

1.1.2.

1.1.3.Create a counter and once it reaches 100, stop the draw function





1.2.

tsk_1_2 | Processing 3.5.3

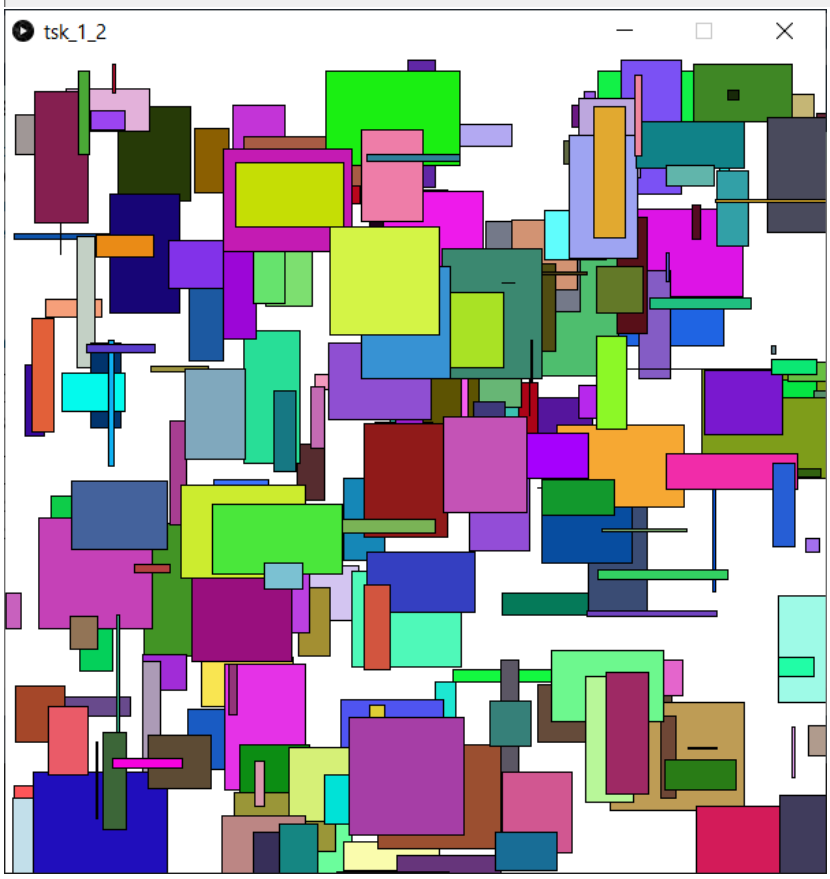
File Edit Sketch Debug Tools Help

Java

tsk_1_2

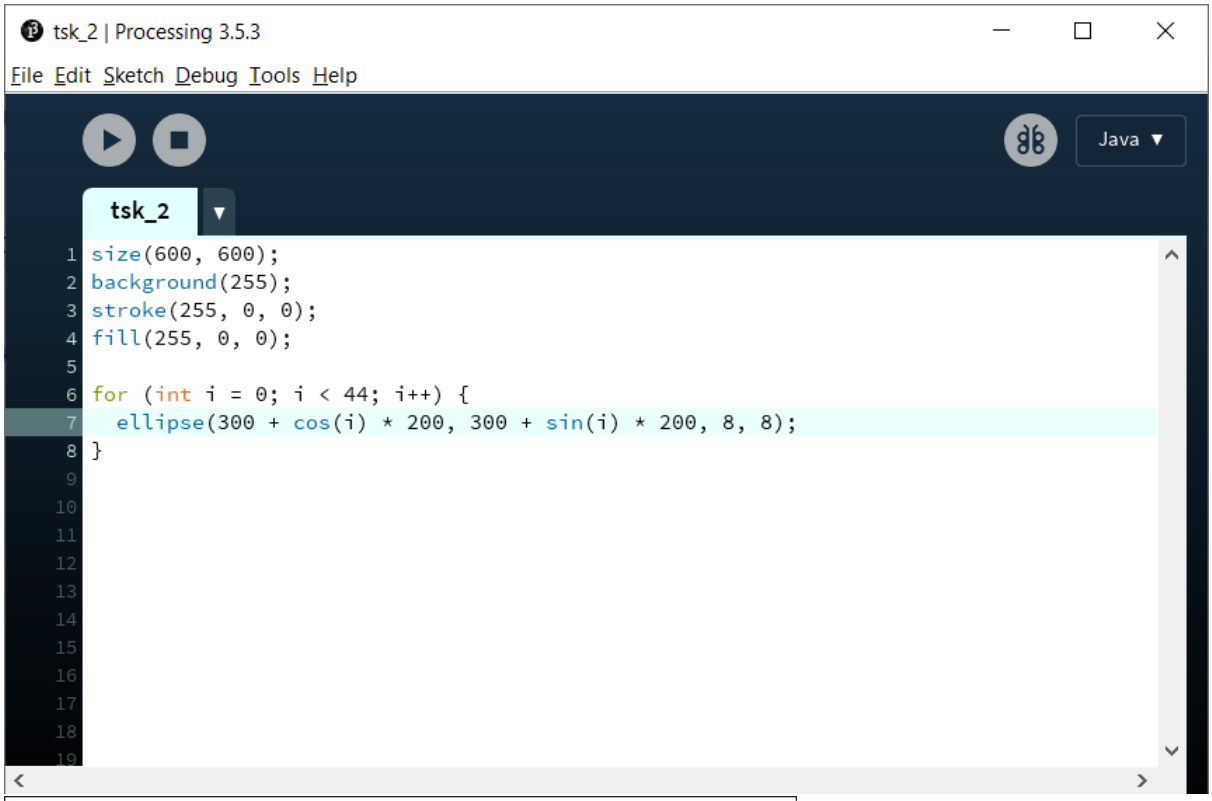
```
void setup(){
size(600, 600);
background(255);
rectMode(CENTER);

}

void draw() {
float c1 = random(255);
float c2 = random(255);
float c3 = random(255);
float x1 = random(600);
float y1 = random(600);
float x2 = random(100);
float y2 = random(100);

fill(c1, c2, c3);
rect(x1, y1, x2, y2);

}
```

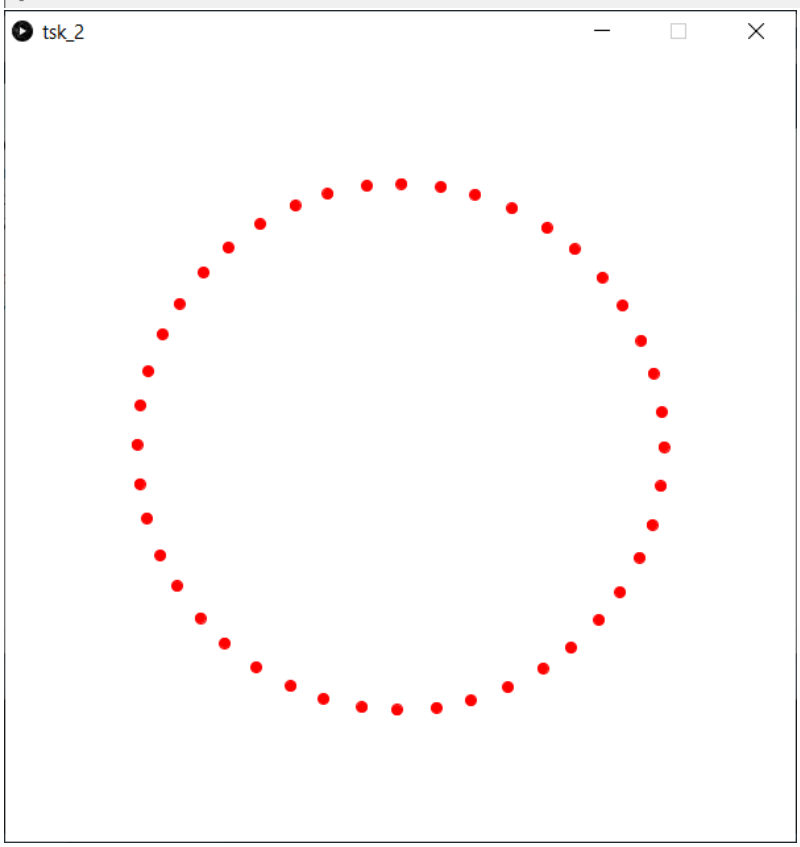
tsk_1_2



2.

2.1.


tsk_2 | Processing 3.5.3
File Edit Sketch Debug Tools Help
Java
tsk_2
size(600, 600);
background(255);
stroke(255, 0, 0);
fill(255, 0, 0);
for (int i = 0; i < 44; i++) {
ellipse(300 + cos(i) * 200, 300 + sin(i) * 200, 8, 8);
}

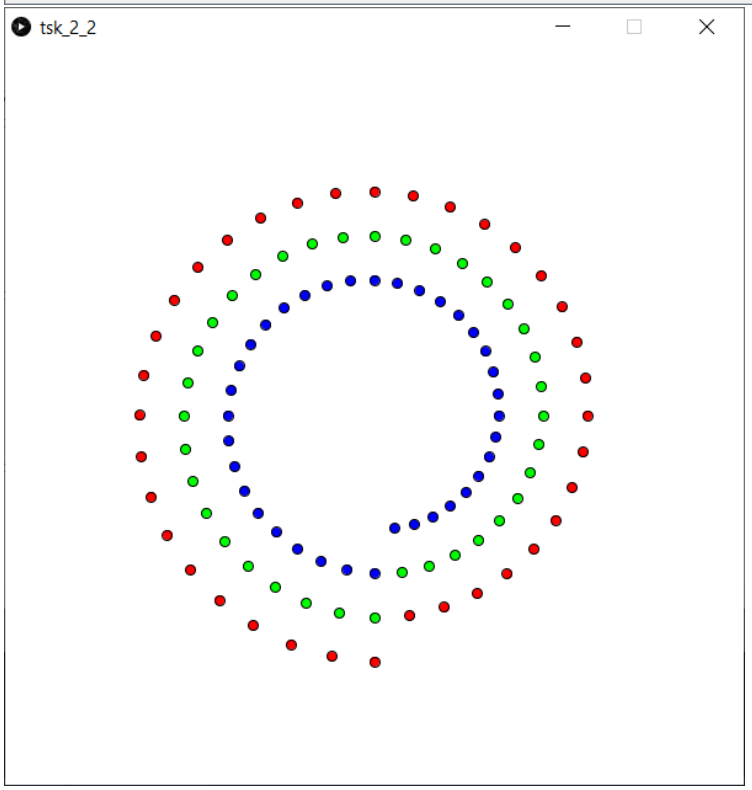

tsk_2

2.2.

tsk_2_2

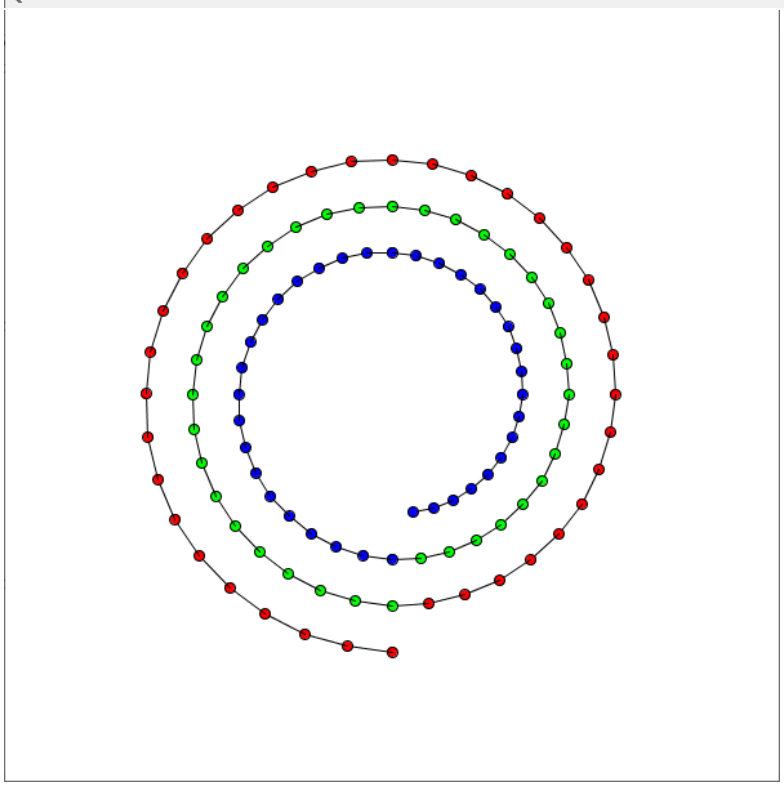
```
size(600, 600);
background(255);

float x = 0;
float y = 0;

fill(255, 0, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-i);
y = 300 + sin(radians(i*10+90)) * (200-i);
ellipse(x, y, 8, 8);
}

fill(0, 255, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+36));
y = 300 + sin(radians(i*10+90)) * (200-(i+36));
ellipse(x, y, 8, 8);
}

fill(0, 0, 255);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+72));
y = 300 + sin(radians(i*10+90)) * (200-(i+72));
ellipse(x, y, 8, 8);
}
```

tsk_2_2



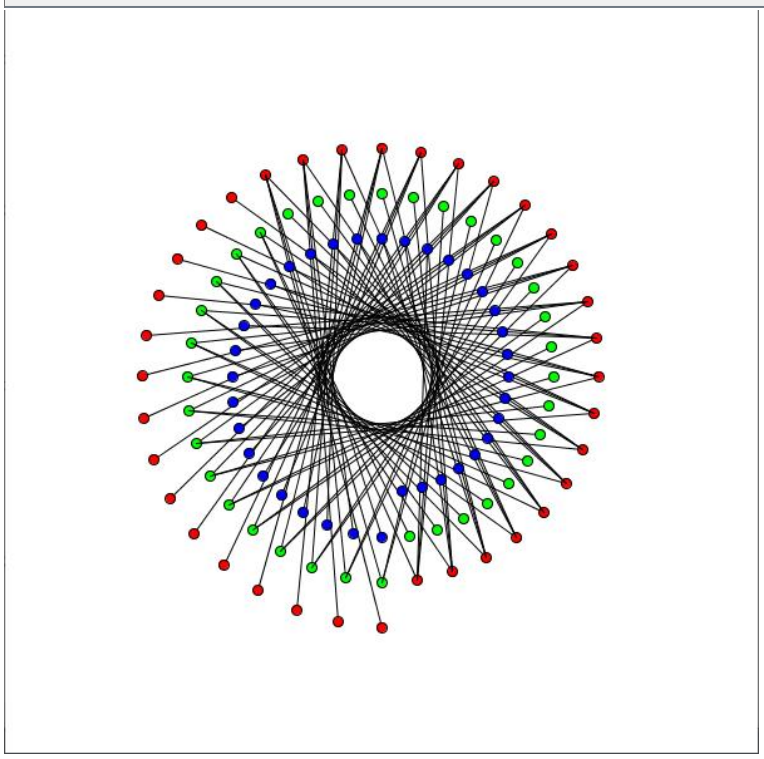
2.3.

```
float x = 0;
float y = 0;
float xPrev = 300 + cos(radians(90)) * (200);
float yPrev = 300 + sin(radians(90)) * (200);

fill(255, 0, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-i);
y = 300 + sin(radians(i*10+90)) * (200-i);
line(x, y, xPrev, yPrev);
ellipse(x, y, 8, 8);
xPrev = x;
yPrev = y;
}

fill(0, 255, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+36));
y = 300 + sin(radians(i*10+90)) * (200-(i+36));
line(x, y, xPrev, yPrev);
ellipse(x, y, 8, 8);
xPrev = x;
yPrev = y;
}

fill(0, 0, 255);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+72));
y = 300 + sin(radians(i*10+90)) * (200-(i+72));
line(x, y, xPrev, yPrev);
ellipse(x, y, 8, 8);
xPrev = x;
yPrev = y;
}
```



2.4.

```
size(600, 600);
background(255);

float x = 0;
float y = 0;
float xPrev = 0;
float yPrev = 0;

fill(255, 0, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-i);
y = 300 + sin(radians(i*10+90)) * (200-i);
  int j = i+15;
  xPrev = 300 + cos(radians(j*10+90)) * (200-j);
  yPrev = 300 + sin(radians(j*10+90)) * (200-j);
  line(x, y, xPrev, yPrev);
ellipse(x, y, 8, 8);
}

fill(0, 255, 0);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+36));
y = 300 + sin(radians(i*10+90)) * (200-(i+36));
  int j = i+15;
  xPrev = 300 + cos(radians(j*10+90)) * (200-j);
  yPrev = 300 + sin(radians(j*10+90)) * (200-j);
  line(x, y, xPrev, yPrev);
ellipse(x, y, 8, 8);
xPrev = x;
yPrev = y;
}

fill(0, 0, 255);
for (int i = 0; i < 36; i++) {
x = 300 + cos(radians(i*10+90)) * (200-(i+72));
y = 300 + sin(radians(i*10+90)) * (200-(i+72));
if (i + 15 < 36){
  int j = i+15;
  xPrev = 300 + cos(radians(j*10+90)) * (200-j);
  yPrev = 300 + sin(radians(j*10+90)) * (200-j);
  line(x, y, xPrev, yPrev);
}
ellipse(x, y, 8, 8);
xPrev = x;
yPrev = y;
}
```

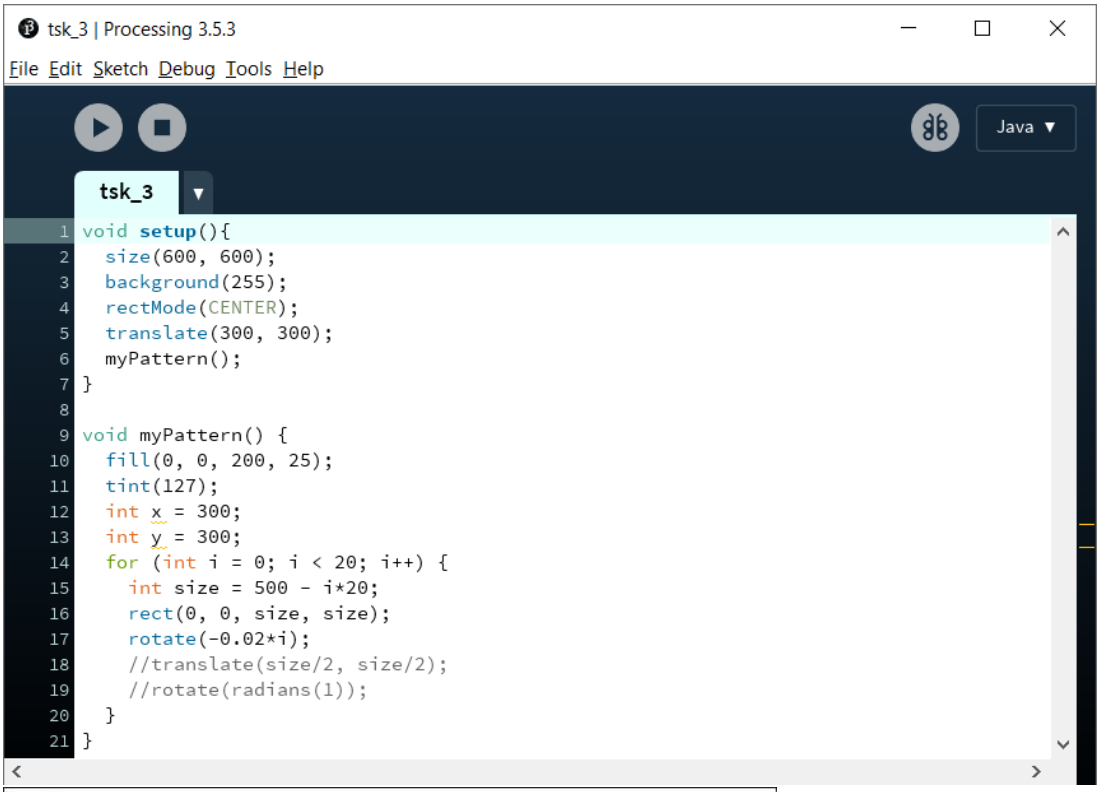


3.

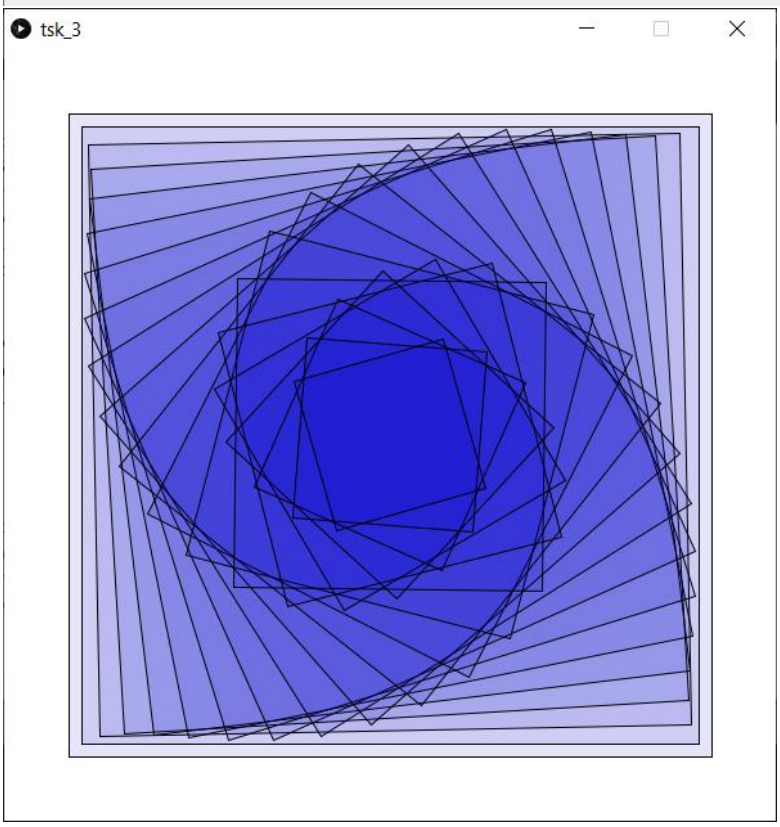
3.1.

3.1.1. It looks darker in the centre because there are more transparent layers stacked there so the colours stack to make a deeper blue

3.1.2. With draw() the squares are all a deep blue colour because they are drawn multiple times on top of one another which stacks the colours across all the squares

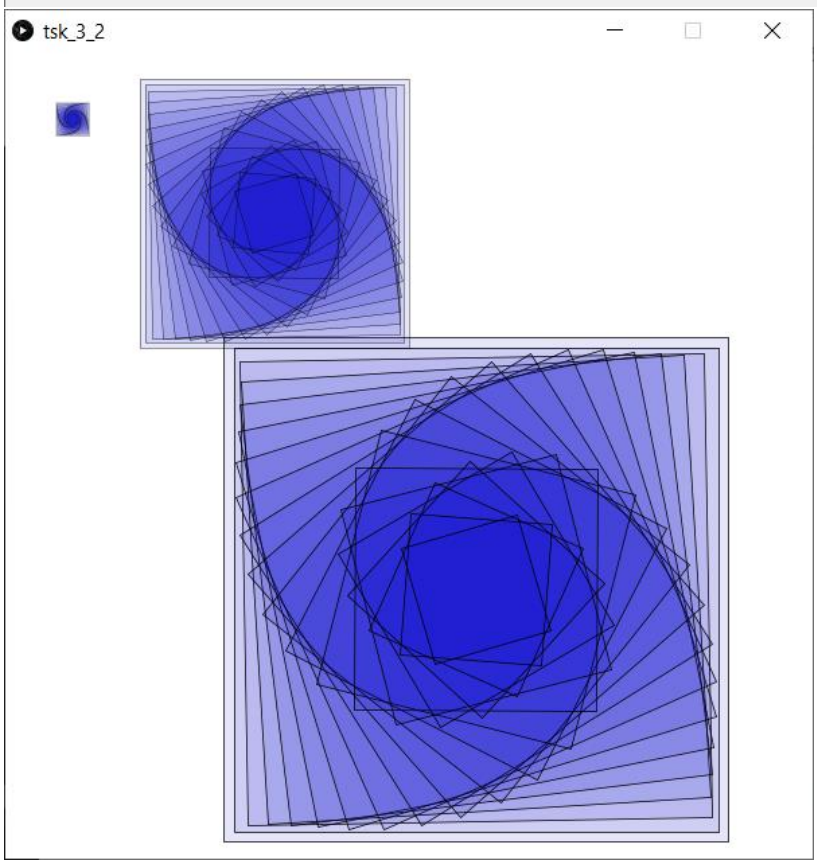



```
void setup(){
  size(600, 600);
  background(255);
  rectMode(CENTER);
  translate(300, 300);
  myPattern();
}

void myPattern() {
  fill(0, 0, 200, 25);
  tint(127);
  int x = 300;
  int y = 300;
  for (int i = 0; i < 20; i++) {
    int size = 500 - i*20;
    rect(0, 0, size, size);
    rotate(-0.02*i);
    //translate(size/2, size/2);
    //rotate(radians(1));
  }
}
```



3.2.

```
void setup(){
  size(600, 600);
  background(255);
  rectMode(CENTER);

  pushMatrix();
  translate(50, 50);
  scale(0.05);
  myPattern();
  popMatrix();

  pushMatrix();
  translate(350, 400);
  scale(0.75);
  myPattern();
  popMatrix();

  pushMatrix();
  translate(200, 120);
  scale(0.4);
  myPattern();
  popMatrix();
}

void myPattern() {
  fill(0, 0, 200, 25);
  tint(127);
  for (int i = 0; i < 20; i++) {
    int size = 500 - i*20;
    rect(0, 0, size, size);
    rotate(-0.02*i);
  }
}
```



Lab 3

1.

tsk_1 | Processing 3.5.3

File Edit Sketch Debug Tools Help

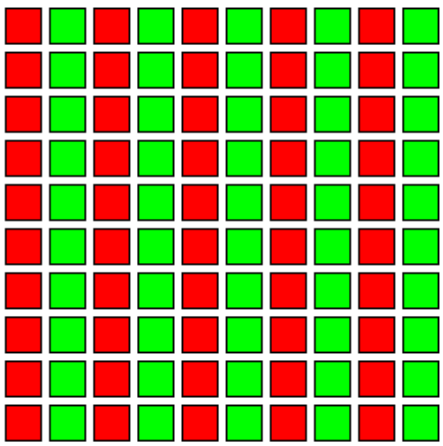
Java

tsk_1

```
void setup() {
  size(400, 400);
  background(255);
}

void draw() {
  for (int i = 1; i <= 10; i++) {
    fill(255, 0, 0);
    for (int j = 1; j <= 10; j++) {
       rect(25*i, 25*j, 20, 20);
    }
    i++;
    fill(0, 255, 0);
    for (int j = 1; j <= 10; j++) {
       rect(25*i, 25*j, 20, 20);
    }
  }
}
```

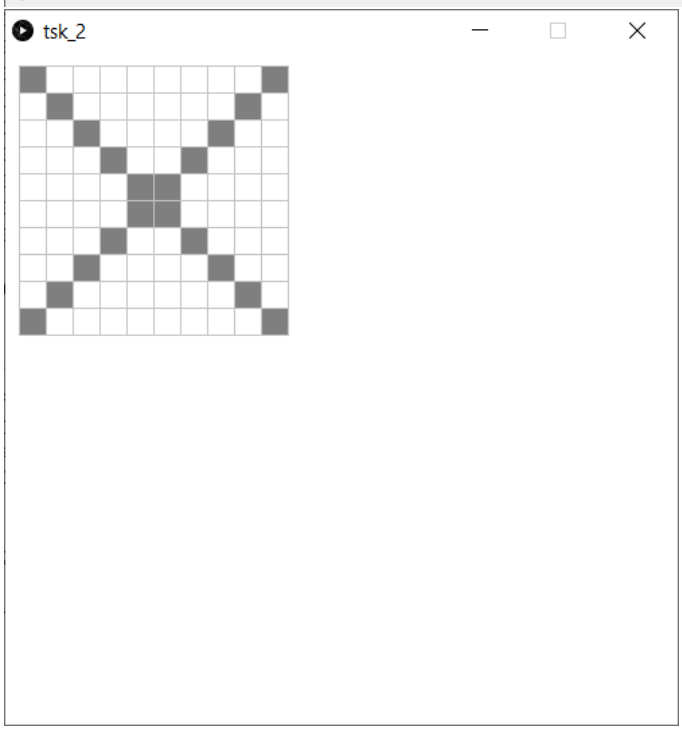
Console Errors

tsk_1



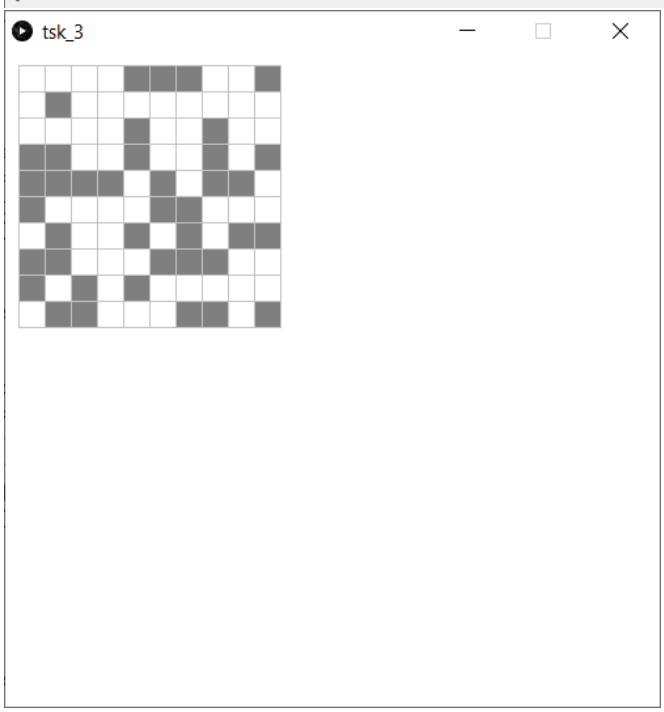
2.

```
void setup() {
 size(500, 500);
 background(255);
 stroke(195);
 int [][] myData ={
  {1,0,0,0,0,0,0,0,0,1},
  {0,1,0,0,0,0,0,0,1,0},
  {0,0,1,0,0,0,0,1,0,0},
  {0,0,0,1,0,0,1,0,0,0},
  {0,0,0,0,1,1,0,0,0,0},
  {0,0,0,0,1,1,0,0,0,0},
  {0,0,0,1,0,0,1,0,0,0},
  {0,0,1,0,0,0,0,1,0,0},
  {0,1,0,0,0,0,0,0,1,0},
  {1,0,0,0,0,0,0,0,0,1}
  };
  myDraw(myData);
}

void myDraw(int[][] array){
  for (int i = 0; i < 10; i++) {
    for (int j = 0; j < 10; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}
```

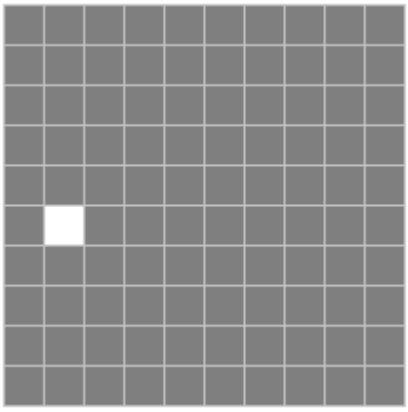


3.

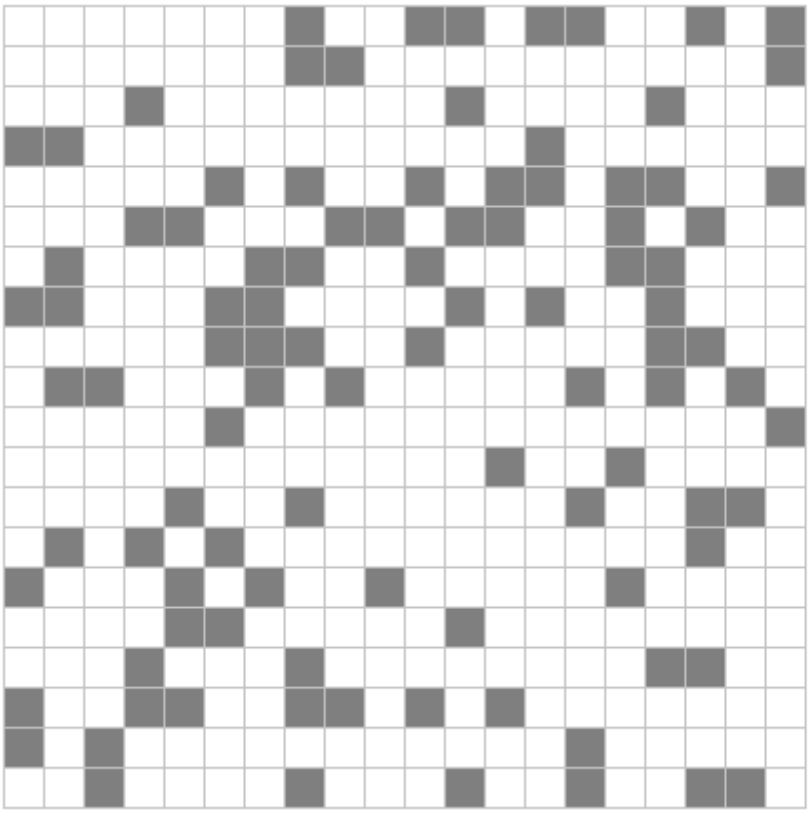
```
void setup() {
 size(500, 500);
 background(255);
 stroke(195);
 int[][] myData = new int[10][10];
 myClearData(myData);
 myRandomSet(myData, 40);
 myDraw(myDraw);
}

void myDraw(int[][] array){
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}

int[][] myClearData(int[][] array) {
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++){
      array[i][j] = 0;
    }
  }
  return array;
}

int[][] myRandomSet(int[][] array, int count) {
  for (int i = 0; i < count && i < array.length*array.length; i++) {
    int x = int(random(array.length));
    int y = int(random(array.length));
    if (array[x][y] == 0) {
      array[x][y] = 1;
    }
    else {
     i--;
    }
  }
  return array;
}
```



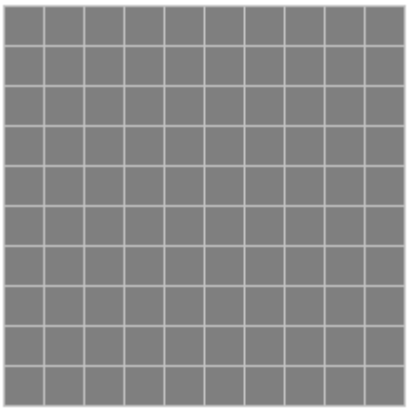
3.1.



3.2.



3.3.



4.

```
void setup() {
 size(1050, 1050);
 background(255);
 stroke(195);
 int[][] myData = new int[50][50];
 myClearData(myData);
 myRandomSet(myData, 1000);
 myDraw(myData);
}

void myDraw(int[][] array){
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}

int[][] myClearData(int[][] array) {
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++){
      array[i][j] = 0;
    }
  }
  return array;
}

int[][] myRandomSet(int[][] array, int count) {
  for (int i = 0; i < count && i < array.length*array.length; i++) {
    int x = int(random(array.length));
    int y = int(random(array.length));
    if (array[x][y] == 0) {
      array[x][y] = 1;
    }
    else {
     i--;
    }
  }
  return array;
}
```

Lab 4

1.

1.1. i) The grid is filled; ii) A square in the bottom right corner of the last 20 squares is filled in; iii) All but the rightmost column is filled in, as it starts at -1 – outside the grid

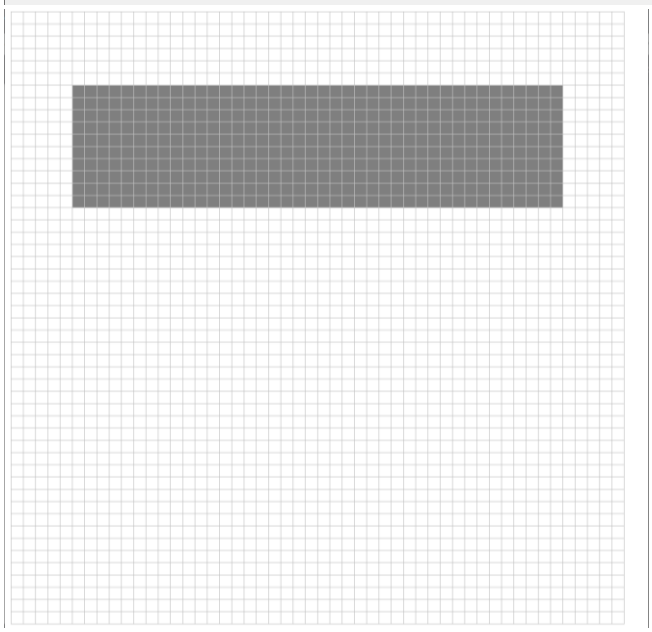
tsk_1 | Processing 3.5.3

File Edit Sketch Debug Tools Help

Java

tsk_1

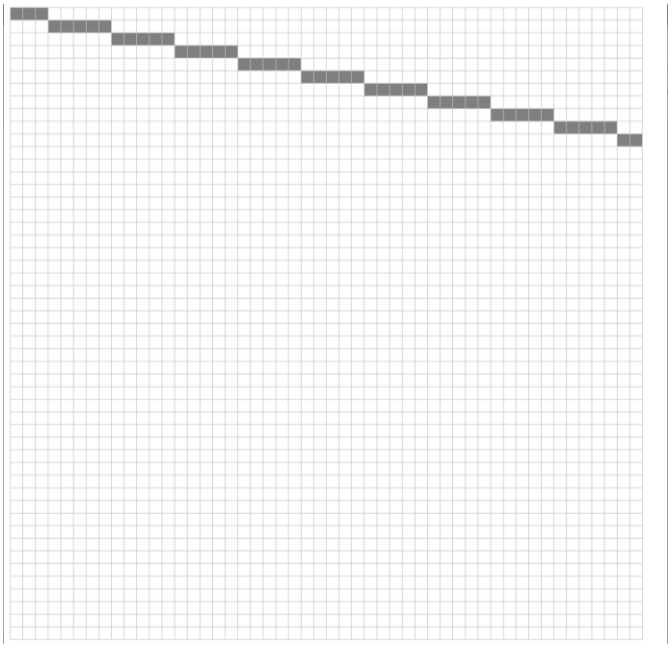
```
void setup() {
 size(1050, 1050);
 background(255);
 stroke(195);
 int[][] myNewData = new int[50][50];
 myClearData(myNewData);
 myRect(myNewData, 5, 6, 40 , 10);
 myDraw(myNewData);
}

void myDraw(int[][] array){
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}

int[][] myClearData(int[][] array) {
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++){
      array[i][j] = 0;
    }
  }
  return array;
}

int[][] myRect(int[][] array, int xPos, int yPos, int wid, int hei) {
  for (int i = xPos; i < xPos + wid && i < array.length*array.length; i++) {
    for (int j = yPos; j < yPos + hei && j < array.length*array.length; j++) {
      if (j >= 0 && i >= 0) {
        array[j][i] = 1;
      }
    }
  }
  return array;
}
```



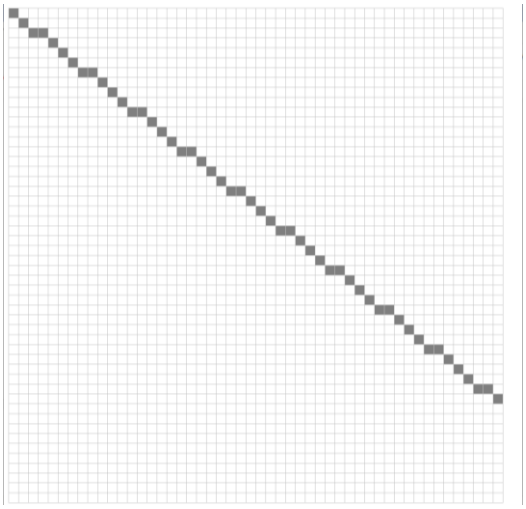
2.

```
void setup() {
 size(1050, 1050);
 background(255);
 stroke(195);
 int[][] myNewData = new int[50][50];
 myClearData(myNewData);
 myLine(myNewData, 0, 0, 50 , 10);
 myDraw(myNewData);
}

void myDraw(int[][] array){
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}

int[][] myClearData(int[][] array) {
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++){
      array[i][j] = 0;
    }
  }
  return array;
}

int[][] myLine(int[][] array, int x1, int y1, int x2, int y2) {
  double xn = x2 - x1;
  double yn = y2 - y1;
  double m = yn/xn;
  double c = y1 - m * x1;
  for (int i = x1; i < x2; ++i) {
    double j = m * i + c;
    if (i < array.length && (int) Math.round(j) < array.length) {
      array[(int) Math.round(j)][i] = 1;
    }
  }

  return array;
}
```

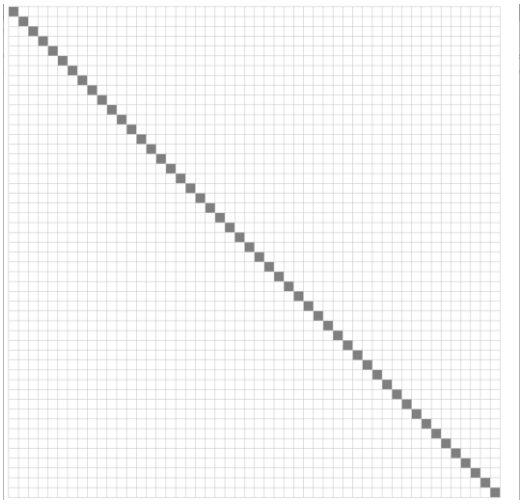
2.1. A diagonal line is drawn from (0, 0) to (50, 10)



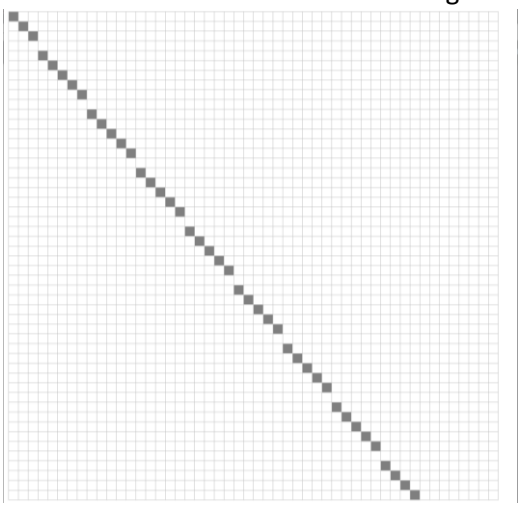
2.2. A diagonal line is drawn from (0, 0) to (50, 40)



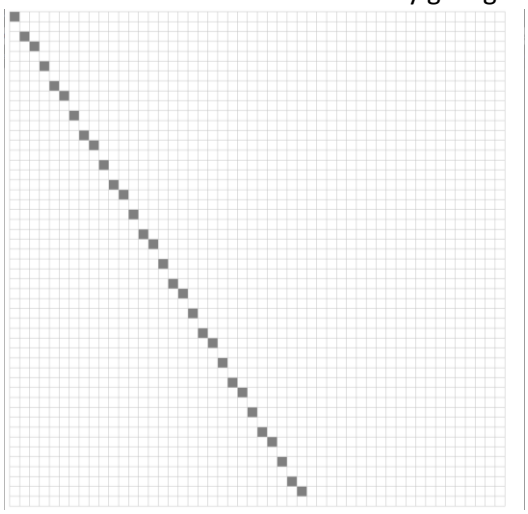
2.3. A diagonal line is drawn from (0, 0) to (50, 50) this is straighter than the others as it has a direct diagonal path between the two coordinates



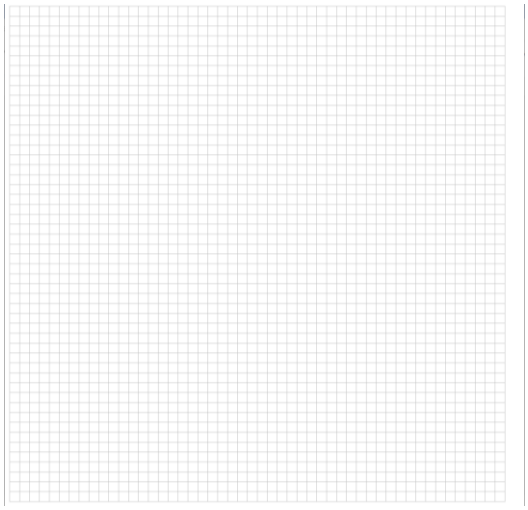
2.4. A diagonal line is drawn from (0, 0) to a point outside the grid. This causes the line to be stretched and so some of the points are missing. This could be fixed by giving the Y-axis calculations a more accurate rounding



2.5. A diagonal line is drawn from (0, 0) to (30, 50). Because the Y-axis calculations have a decimal value rounded to the nearest whole number, some of the points are missing from the line. This could also be fixed by giving the Y-axis calculations a more accurate rounding



2.6. The line starts drawing at the edge of the grid, but the algorithm used does not account for a line going in a negative direction, so no line is drawn. This could be fixed by determining which point is the smallest, and then drawing from that point



3.

3.1. It doesn't look like a good circle as there are many missing pixels and some erroneous ones. I could improve the circle by giving a more accurate rounding or using a finer/larger grid

3.2. The problem with the naïve circle algorithm is that circles are not very well represented with such large pixel sizes – they require finer detail for their curved sides

```
void setup() {
 size(1050, 1050);
 background(255);
 stroke(195);
 int[][] myNewData = new int[50][50];
 myClearData(myNewData);
 myCircle(myNewData, 20, 20, 10);
 myDraw(myNewData);
}

void myDraw(int[][] array){
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++) {
      if (array[j][i] == 1) {
        fill(127);
      }
      else {
        fill(255);
      }
      rect(10+20*i, 10+20*j, 20, 20);
    }
  }
}

int[][] myClearData(int[][] array) {
  for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array.length; j++){
      array[i][j] = 0;
    }
  }
  return array;
}

int[][] myCircle(int[][] array, int xPos, int yPos, float rad) {
  for (int i = 0; i < array.length; i++) {
        array[(int)(yPos+sin(i)*rad)][(int)(xPos+cos(i)*rad)] = 1;
      }
  return array;
}
```