Lab 2

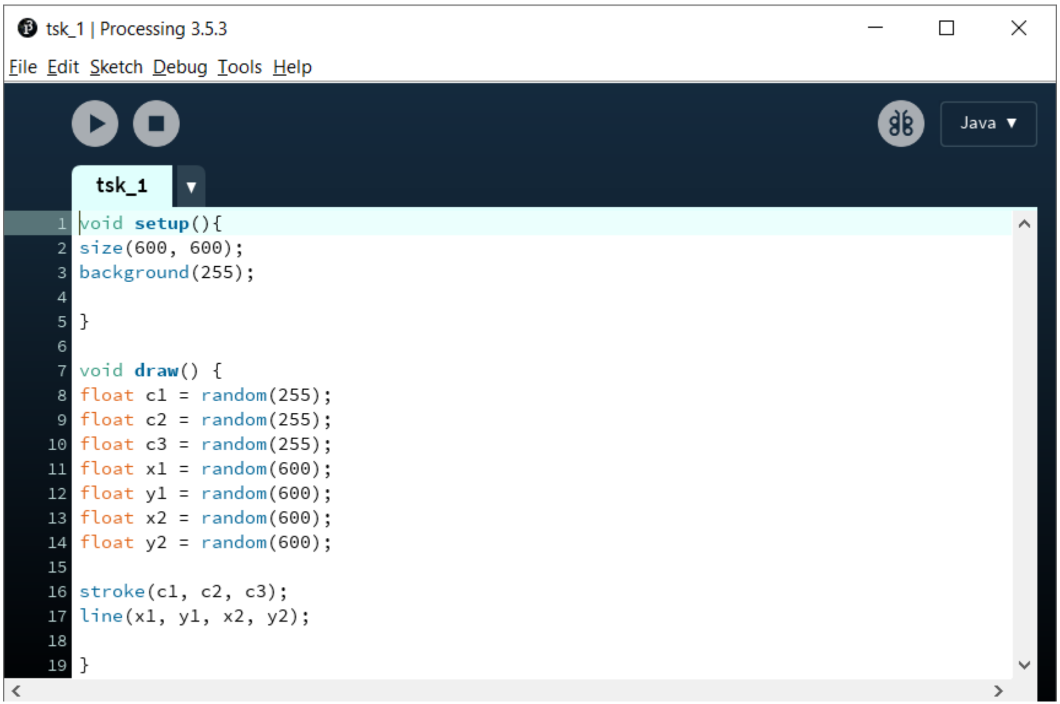
1.

1.1.

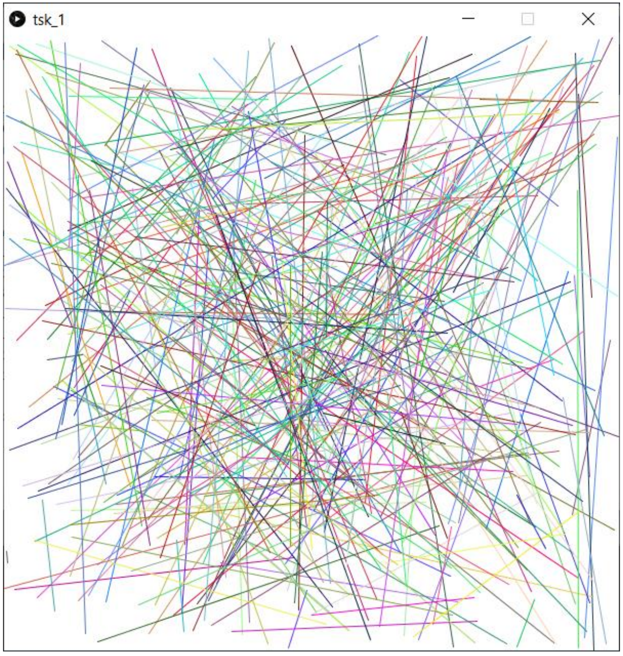
1.1.1. It gradually updates the screen because the draw function is being called repeatedly

1.1.2.

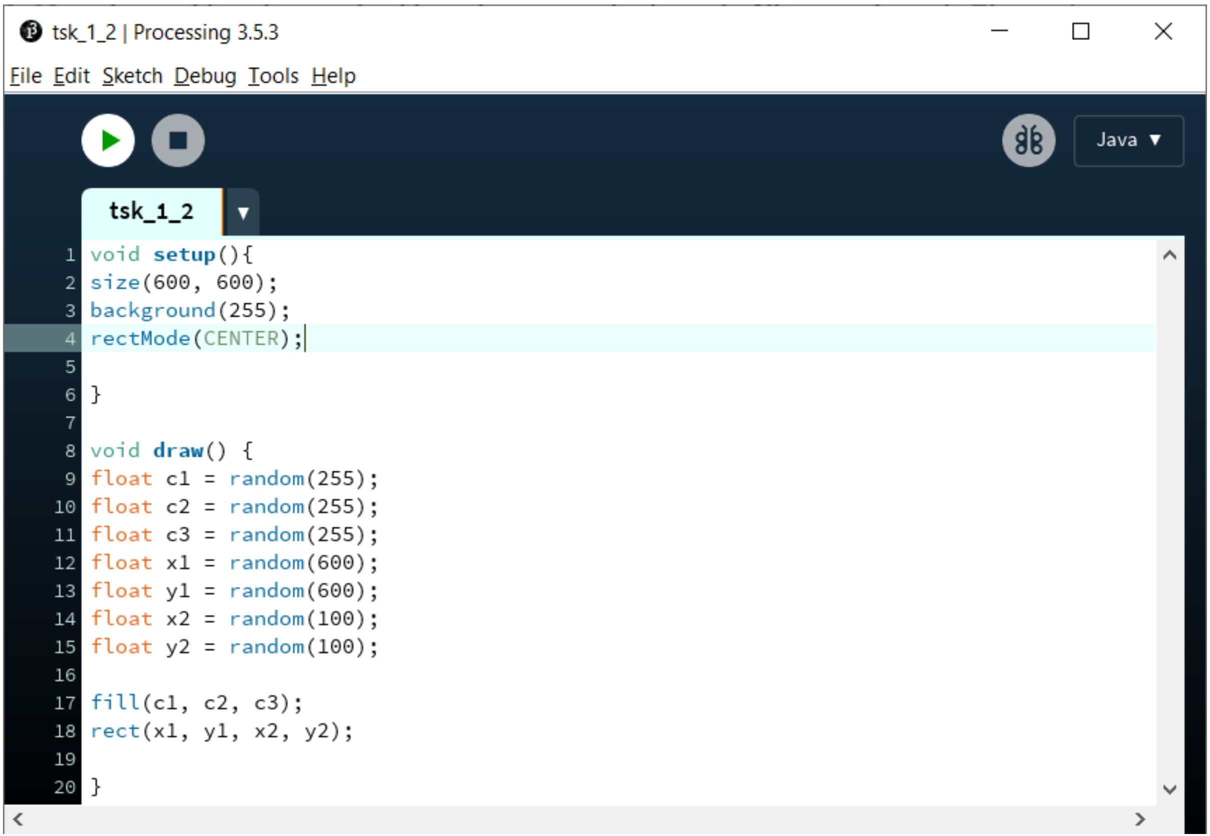
1.1.3. Create a counter and once it reaches 100, stop the draw function



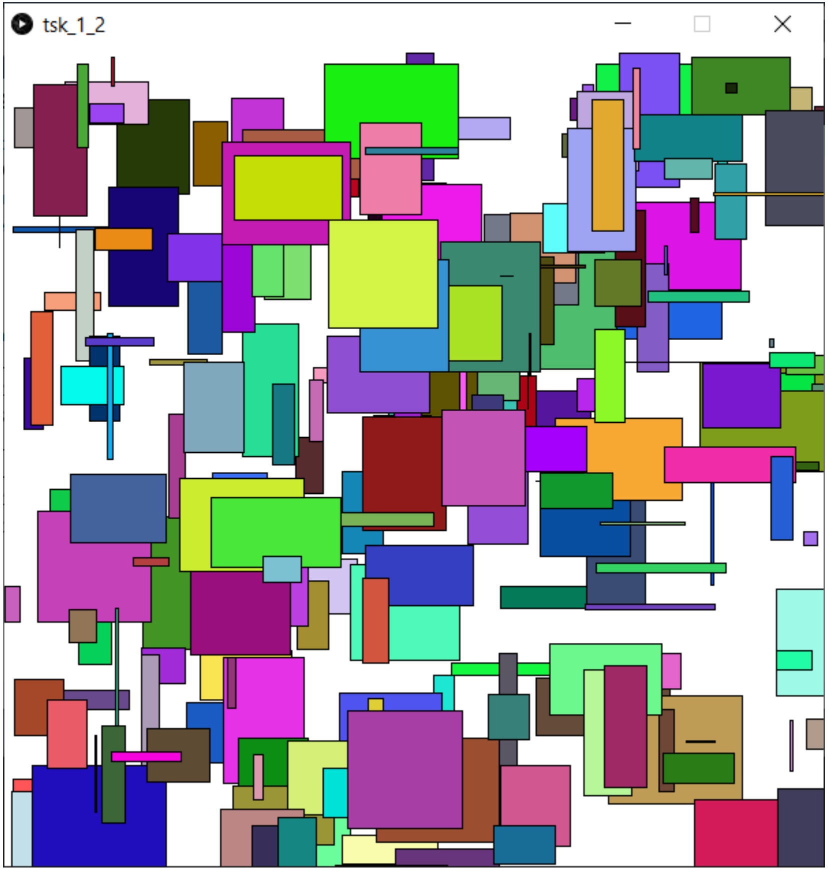
```
void setup(){
size(600, 600);
background(255);

}

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(600);
float y2 = random(600);

stroke(c1, c2, c3);
line(x1, y1, x2, y2);

}
```



1.2.

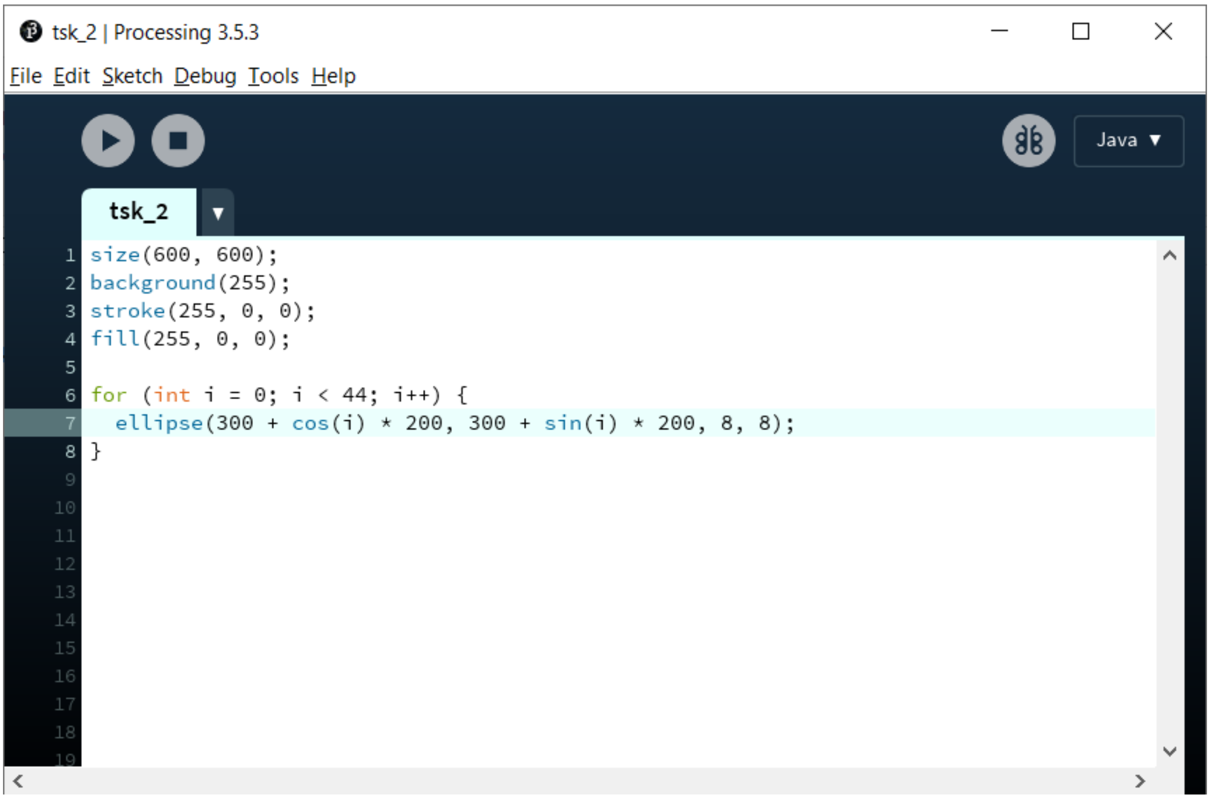


```
void setup(){
size(600, 600);
background(255);
rectMode(CENTER);

}

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);

}
```

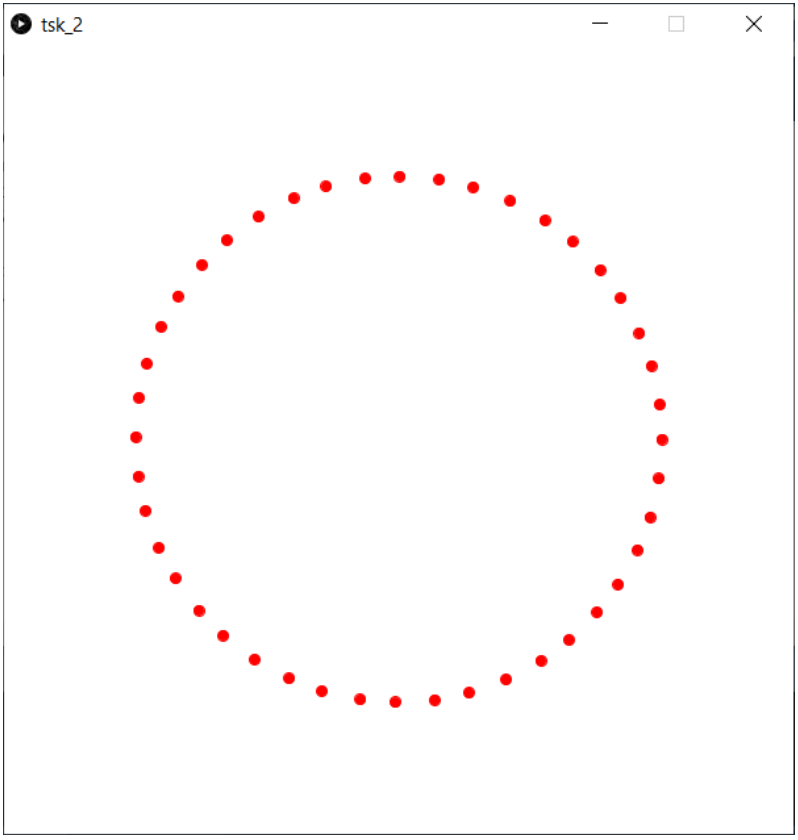


2.

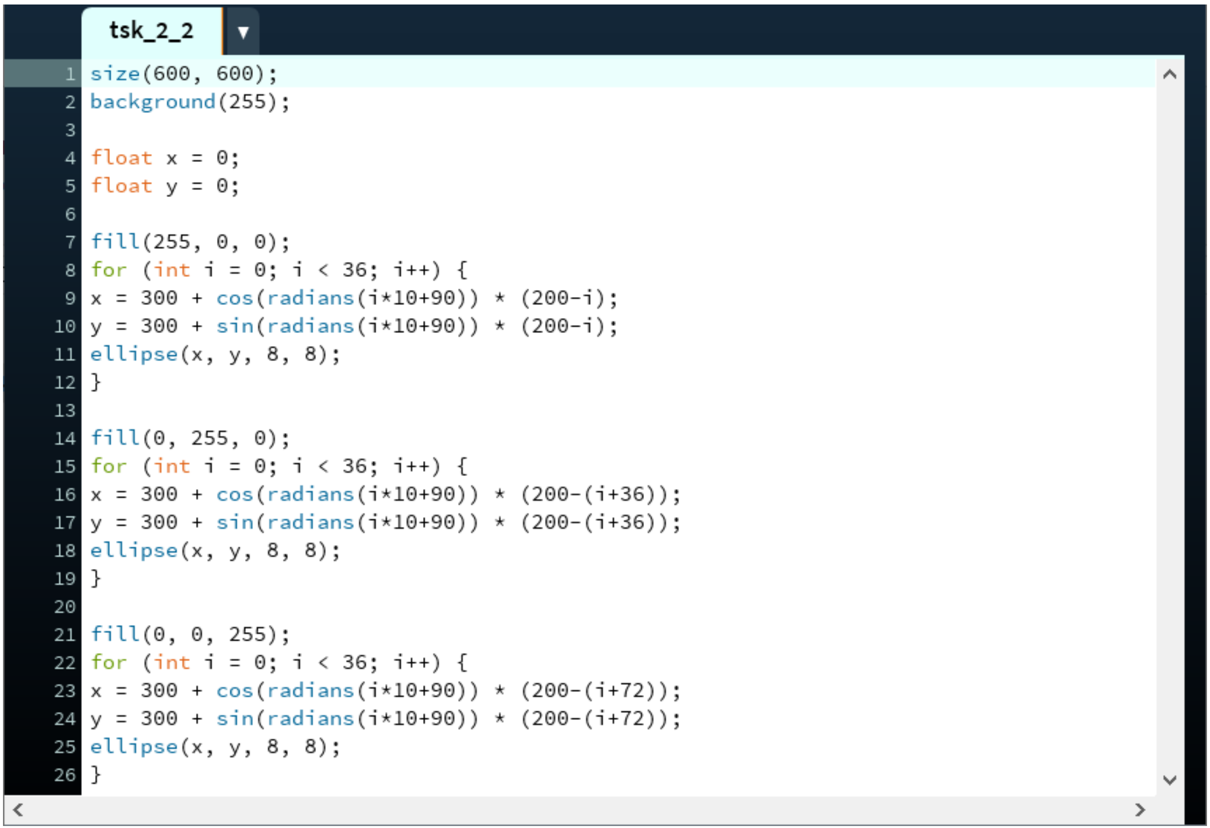
2.1.



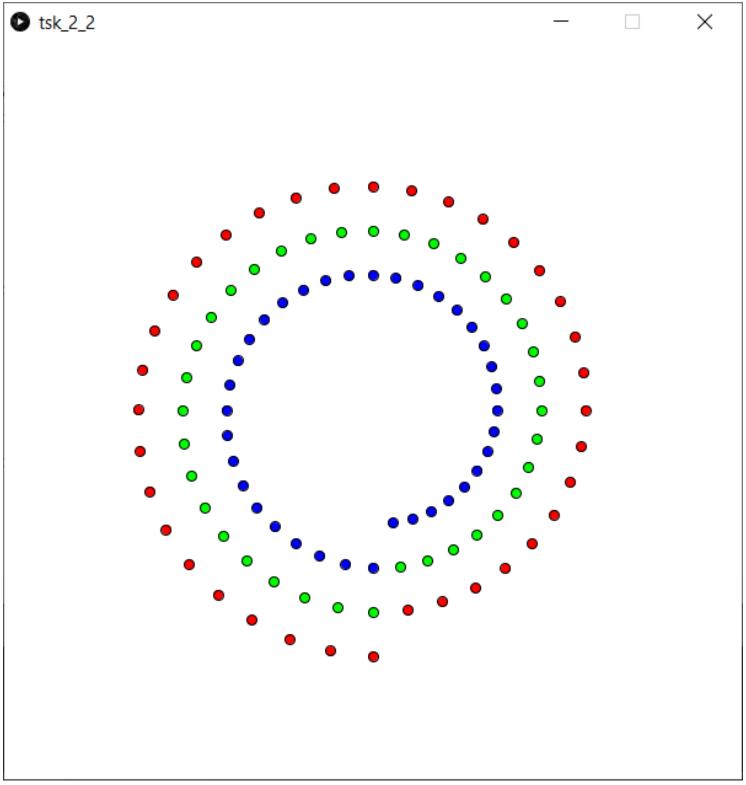
```
size(600, 600);
background(255);
stroke(255, 0, 0);
fill(255, 0, 0);

for (int i = 0; i < 44; i++) {
  ellipse(300 + cos(i) * 200, 300 + sin(i) * 200, 8, 8);
}
```



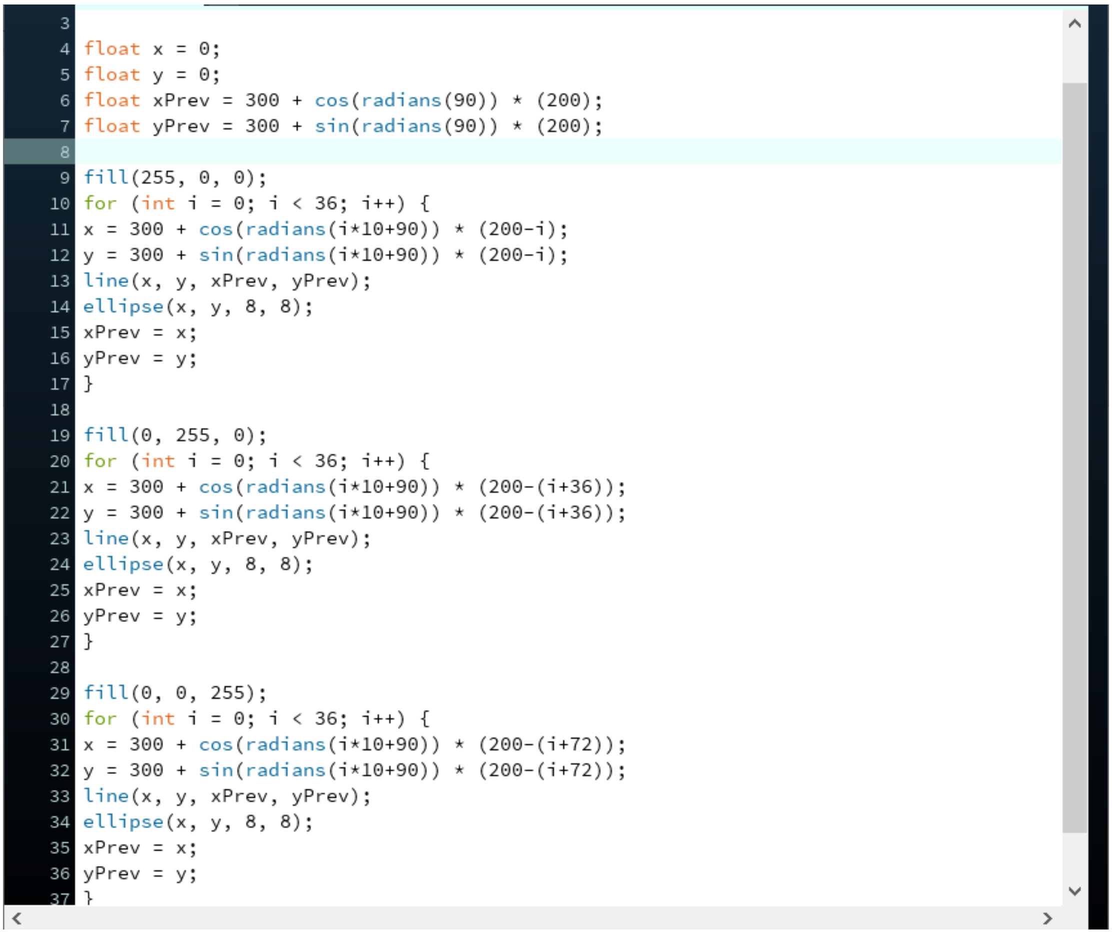
2.2.



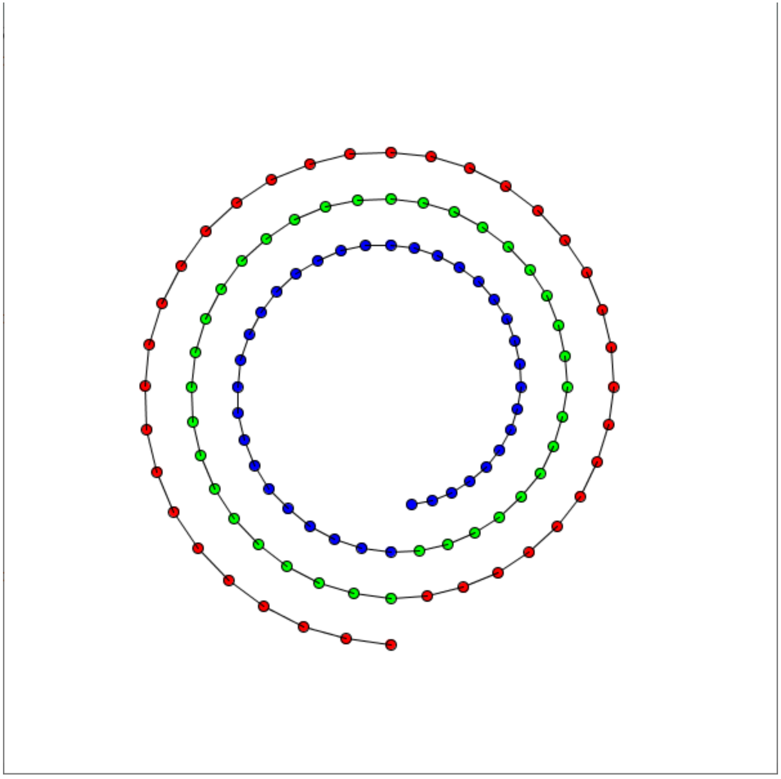
```
size(600, 600);
background(255);

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);
}
```



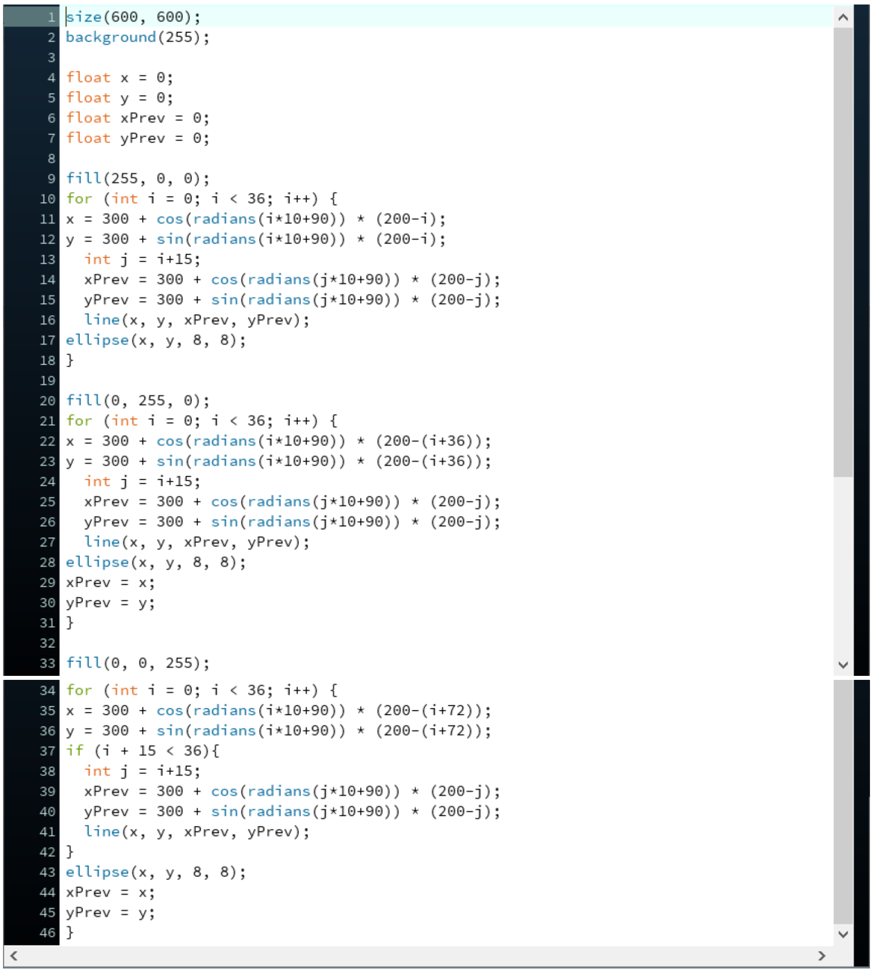
2.3.



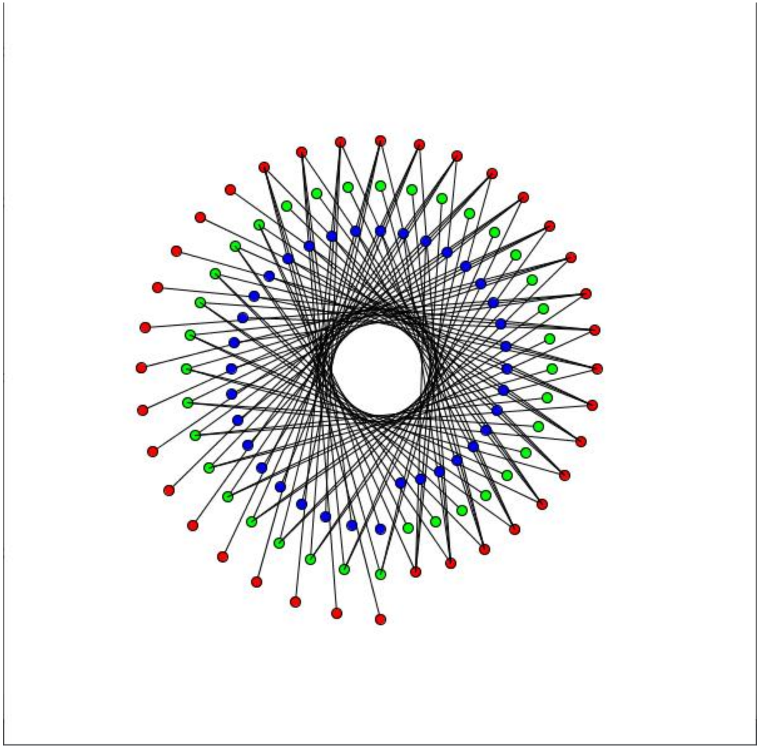
```

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;
}
```

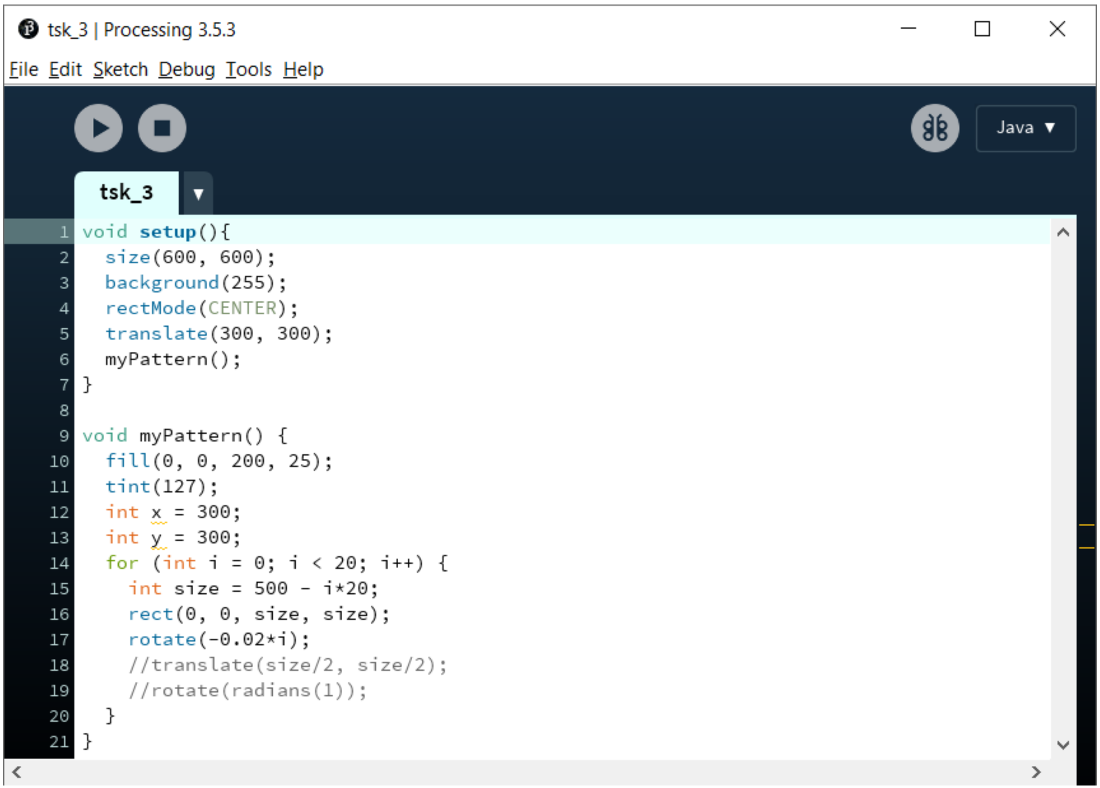


3.

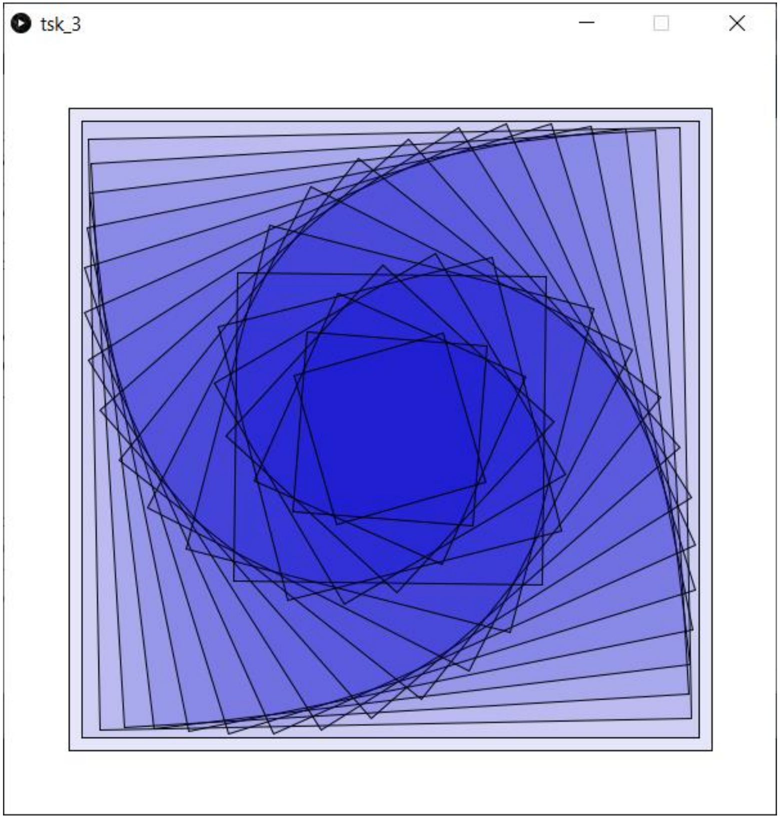
3.1.

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

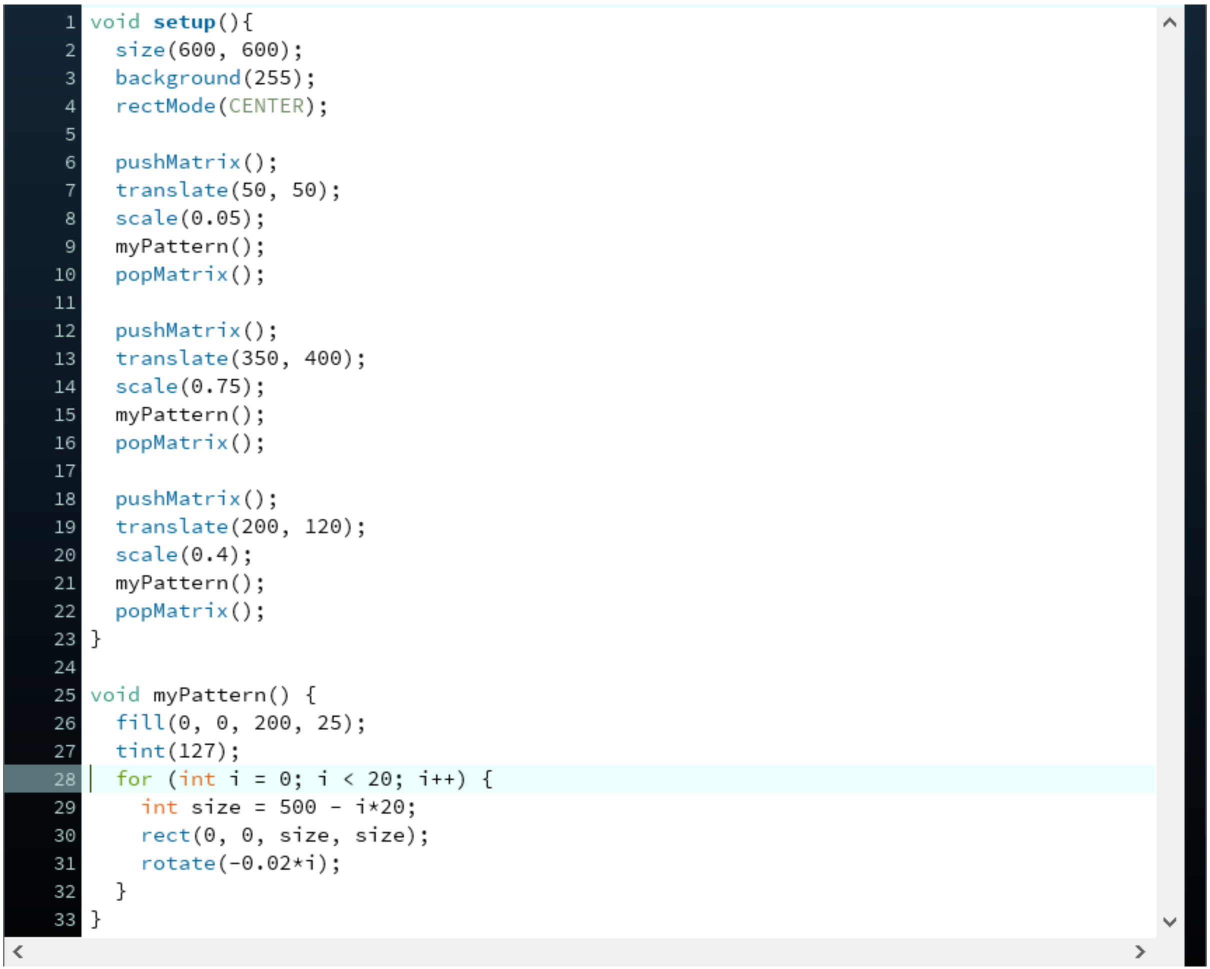
3.1.2. With draw() the squares are all a deep blue colour because they are drawn multiple times on top of one another which stacks the colours across all the squares



```
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);
  }
}
```