```
// The Chance Execution
// Olle Essvik

/* ***** Marbling******
 The colour floating on the water surface,
the resulting pattern, governed by
movements, temperature and dosage. The
marbler's mood and state of mind on a
specific day. A certain difference is
unavoidable. The marbled paper dries.
*/

int   paper_x= 827;
int   paper_y= 1169;
int   water = 149006;
float paint;

/* I look at a picture representing Ken
Perlin. It was Ken Perlin who created the
algorithm that aims to remove the
computerized – a certain graphical and
mechanical appearance in digital images –
and instead emulate nature's seeming
chaotic complexity. The algorithm was
created in 1983 and there are several
pictures on the Internet of Ken Perlin. Ken
in  different ages, photographs in
different resolutions, formats and graphics.
I use Perlin noise and begin typing a code
inspired by marbled papers.
*/

/* An auction in the north of Sweden, held
on the outskirts of a medium-sized Swedish
town.
```


2851.
PAPPER 80:- ST 97 01 30

I make my bids online. The bookbinder died and the son did not take over. The bindery is auctioned off in batches. Hot foil stamping machines, lead types and tools. A coffee maker next to bone folders, book cloth and paper.

I make a bid for one of the larger boxes, containing paper used for front- and endpaper covers. A green filing cabinet with 12 drawers, each filled with marbled papers. Thousands of papers. Most of the papers in the drawer are endpapers. The part that holds together the pages of the book with the front and the back. Uniquely marbled in different colours and styles. Industrial papers produced in larger editions with patterns that appear identical. /*

/* One box for leftover papers. The trace of the second paper, it's shape and the execution in the piece left behind. This is the content of the box in front of me. The papers the bookbinder did not choose. */

/* I'm looking for a picture of the bookbinder on the internet, I find nothing except a name in a database for genealogy. An algorithm searching for names and family bonds following a certain order. */

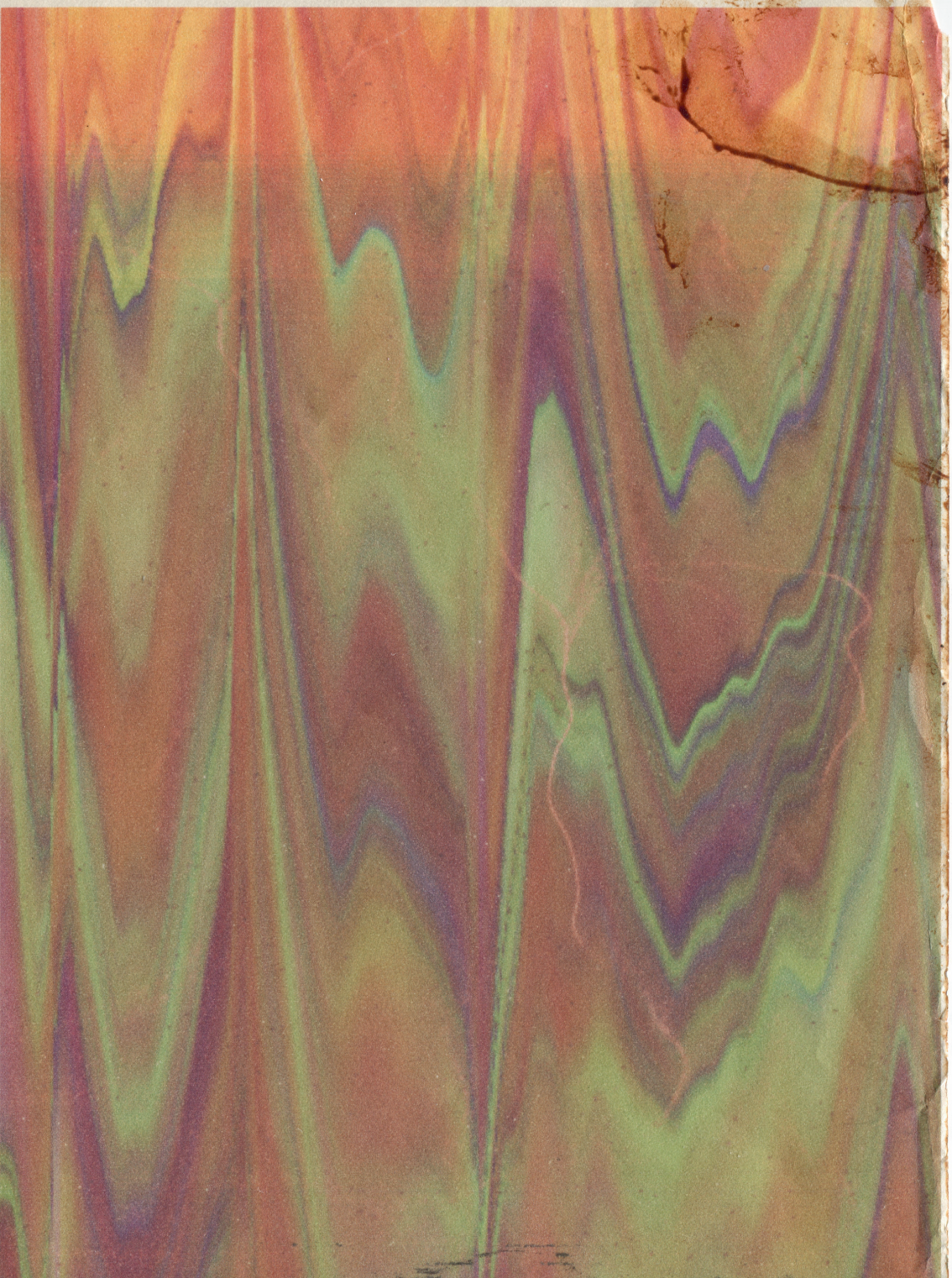
```
int   marblers_mode=0;
float movement;
float surfaceTension;
```


Counter No: 3491
Sorter No: _____
In the event of any note.
Submit this control number.

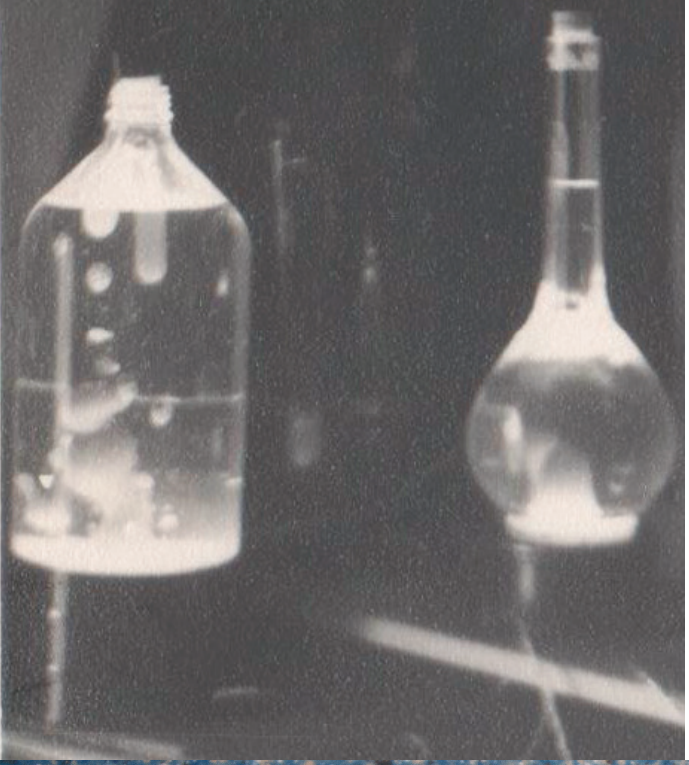
```
int Countress3491;
PVector[] UllaSievert = new PVector[water];
PVector[] DistrictJudge = new PVector[
water];
PVector[] kenPerlin = new PVector[water];

/* The papers in the drawers carry few
traces, apart from the occasional name, the
odd title, a price.

Counteress 3491
Ulla Sievert.
District Judge
Anna
15:-

*/
float bookbinder = random(0, 255);
float lavaLamp = random(0, 255);
float radioNoise = random(0, 255);

float genealogy = random(1)-4;
float anna = random(1)-4;

float mx = map(1, 10, paper_x,  genealogy,
anna);
float my = map(1, 10, paper_y, anna,
genealogy);
void setup() {
size(827, 1169, P2D);

for(Countress3491 = 0; Countress3491 <
water; Countress3491++) {
```

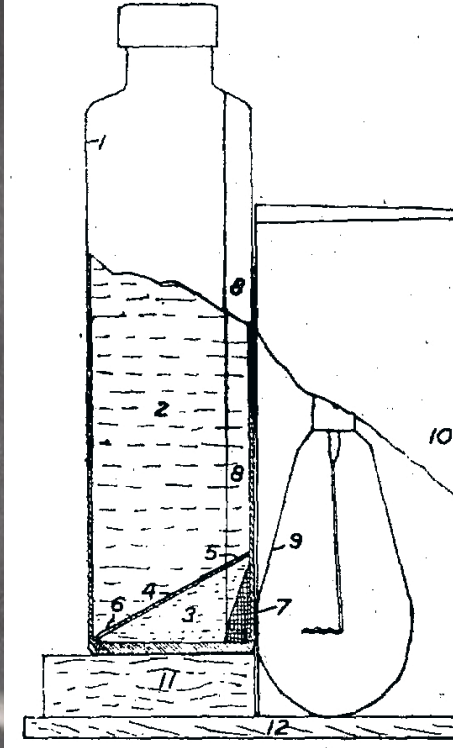
```
UllaSievert[Countress3491]
= new PVector(random(0, paper_x), random(0,
paper_y));
      DistrictJudge[Countress3491] = new
PVector(random(0, paper_x), random(0,
paper_y));
      kenPerlin[Countress3491] = new
PVector(random(0, paper_x), random(0,
paper_y));
  }      }

/* I find a note that reads "counteress no.
"3491", and picture a person behind that
number, someone, long ago, counting papers.
Someone with a family name who counted
paper, based on given instructions and
systems. But countress 3491 could also be
the name of a machine. */

void draw() {
float paper = movement;
for (int x = 0; x < paper_x; x++) {
float marblers = surfaceTension;

for (Countress3491 = 0; Countress3491 <
paper_y; Countress3491++) {
float r = map(noise(paper, marblers), 0, 1,
0, bookbinder);
float g = map(noise(marblers, paper), 0, 1,
0, lavaLamp);
float b = map(noise(paper, marblers, paper),
0, 1, 0, radioNoise);
loadPixels();
pixels[x+Countress3491*paper_x] = color
```

```
(r, g, b);
marblers += -pow(5, my) - pow(5, -5*noise(-
paper, marblers)); }
paper += -pow(5, mx) + -pow(5, -5*noise(
paper, paper));}
loadPixels();

/* A profession that no longer exists and
has been replaced by algorithms and
machines. Counteress 3491 is numbers, an
unknown note.
Having examined all the papers I get a
headache. I am the first person to take these
papers out, after decades in the dark. /*

/* I gather up the papers, one of each kind,
in large stacks. I keep them in the same
order as in the drawers. The bindery had
been around for a long time; several of the
papers are at least 50 years old. The pile
has been growing gradually and the oldest is
right at the bottom. They  follow a certain
order. */

updatePixels();

   for (  paint=0; paint<=60000; paint+=0.1)
  {
     stroke(0, 0, 0,random(15));
     ellipse(paint,random(1600),ra
dom(1),random(1));
  }

float noisy = 0;
```



```
for(marblers_mode = 0; marblers_mode <
water; marblers_mode++) { stroke(lavaLamp,
radioNoise, bookbinder, 45);
      point(UllaSievert[marblers_mode].x,
UllaSievert[marblers_mode].y);
      DistrictJudge[marblers_mode].x
= 20*noise(400+UllaSievert[marblers_
mode].x*.0007, 400+UllaSievert[marblers_
mode].y*.0007, noisy*2)*cos(4*PI*noise(
UllaSievert[marblers_mode].x*.007,
UllaSievert[marblers_mode].y*.007,
noisy*.5));
      DistrictJudge[marblers_mode].y =
10*noise(UllaSievert[marblers_mode].x*.0007,
400+UllaSievert[marblers_mode].y*.0007,
noisy*2)*sin(4*PI*noise(UllaSievert[
marblers_mode].x*.007, UllaSievert[marblers_
mode].y*.007, noisy*.5));
      UllaSievert[marblers_mode].add(
DistrictJudge[marblers_mode]);  }}

/* I press a button on the website and
receive a random number between 1 and 100.
I can see the number, but the number is
not evidence that the algorithm does what
it says it claims. I  press the button five
times: 3, 9, 4, 71, 1. */

/* Silicon Graphics once used a lava lamp to
generate chance for the computer, and
published the random numbers on their
website. The Lava lamp principle is to blend
two immiscible liquids, oil and water. Not
dissimilar to the looks and techniques of
marbling. */
```




1
8
2
8
10
9
5
4
6
3
7
11
12

```
/* I glue the papers. With gauze I attach
the coded endpapers in the end and beginning
of the book with the endpapers found in the
drawer. Leftover papers from the chance
execution. Beginning and end of a book. The
pages that you never read. */

void keyPressed() {
  switch(key) {
  case ' ':
    setup();
    break;
  case 's':
    int m = (int)millis();
    save("pattern"+str(m)+".jpg");
    break;
  default:
    println("unknown key");
    break;
  }}

/*Olle Essvik, under Creative Commons
Attribution ShareAlike https://creativecommons.
org/licenses/by-sa/3.0 adapted and inspired by
2D Perlin noise example from Dan Shiffman's Na-
ture of Code  and  http://www.openprocessing.
org/sketch/157579. The text is from the book
The Chance Execution, Olle Essvik, rojal press.
*/
```

the chance execution

Counter No: 3491

Sorter No: OE

In the event of any note. Submit this control number.