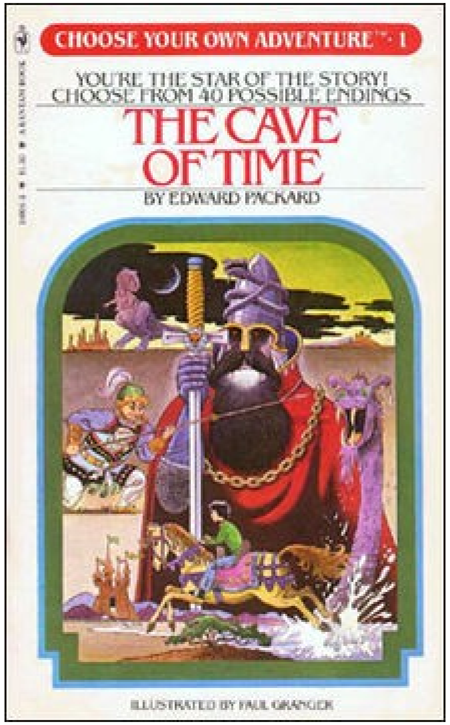
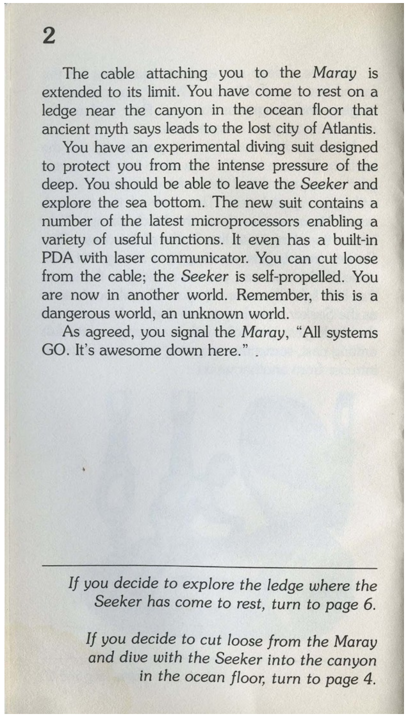
# Basic Programming Part III

| | | |
|---|---|---|
| Text-based video games | → | arrays of char arrays |
| Computational poetry & the cadavre exquis | → | random(), randomSeed() |
| Plotter art & custom drawing machines | → | math functions |

# Choose your own adventure


CHOOSE YOUR OWN ADVENTURE™ · 1
YOU'RE THE STAR OF THE STORY!
CHOOSE FROM 40 POSSIBLE ENDINGS
THE CAVE OF TIME
BY EDWARD PACKARD
ILLUSTRATED BY PAUL GRANGER



2

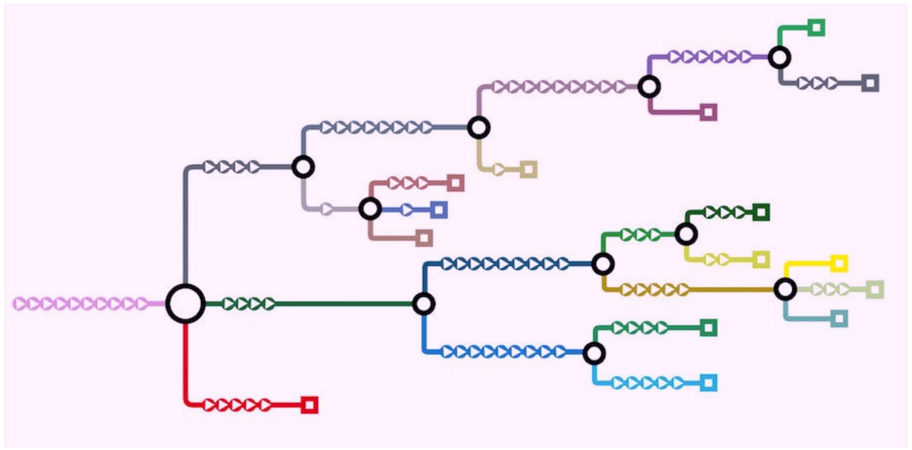
The cable attaching you to the *Maray* is extended to its limit. You have come to rest on a ledge near the canyon in the ocean floor that ancient myth says leads to the lost city of Atlantis.

You have an experimental diving suit designed to protect you from the intense pressure of the deep. You should be able to leave the *Seeker* and explore the sea bottom. The new suit contains a number of the latest microprocessors enabling a variety of useful functions. It even has a built-in PDA with laser communicator. You can cut loose from the cable; the *Seeker* is self-propelled. You are now in another world. Remember, this is a dangerous world, an unknown world.

As agreed, you signal the *Maray*, "All systems GO. It's awesome down here."

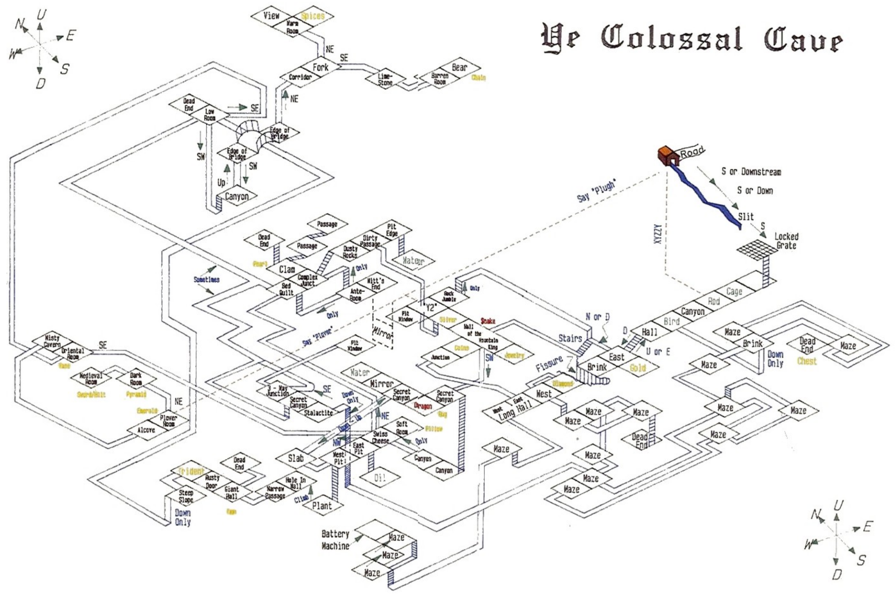
*If you decide to explore the ledge where the Seeker has come to rest, turn to page 6.*

*If you decide to cut loose from the Maray and dive with the Seeker into the canyon in the ocean floor, turn to page 4.*



# Colossal Cave Adventure

```
.run adven


WELCOME TO ADVENTURE!!  WOULD YOU LIKE INSTRUCTIONS?

yes

SOMEWHERE NEARBY IS COLOSSAL CAVE, WHERE OTHERS HAVE FOUND FORTUNES IN
TREASURE AND GOLD, THOUGH IT IS RUMORED THAT SOME WHO ENTER ARE NEVER
SEEN AGAIN.  MAGIC IS SAID TO WORK IN THE CAVE.  I WILL BE YOUR EYES
AND HANDS.  DIRECT ME WITH COMMANDS OF 1 OR 2 WORDS.  I SHOULD WARN
YOU THAT I LOOK AT ONLY THE FIRST FIVE LETTERS OF EACH WORD, SO YOU'LL
HAVE TO ENTER "NORTHEAST" AS "NE" TO DISTINGUISH IT FROM "NORTH".
(SHOULD YOU GET STUCK, TYPE "HELP" FOR SOME GENERAL HINTS.  FOR INFOR-
MATION ON HOW TO END YOUR ADVENTURE, ETC., TYPE "INFO".)
                              - - -
THIS PROGRAM WAS ORIGINALLY DEVELOPED BY WILLIE CROWTHER.  MOST OF THE
FEATURES OF THE CURRENT PROGRAM WERE ADDED BY DON WOODS (DON @ SU-AI).
CONTACT DON IF YOU HAVE ANY QUESTIONS, COMMENTS, ETC.

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK BUILDING.
AROUND YOU IS A FOREST.  A SMALL STREAM FLOWS OUT OF THE BUILDING AND
DOWN A GULLY.

east

YOU ARE INSIDE A BUILDING, A WELL HOUSE FOR A LARGE SPRING.

THERE ARE SOME KEYS ON THE GROUND HERE.

THERE IS A SHINY BRASS LAMP NEARBY.

THERE IS FOOD HERE.
```



# Arrays of chars

Initialzing an array of chars

The compiler counts the elements and creates an array of the appropriate size (length, starting at zero, plus 1 for the null character '\0' at the end)

```
char message[] = {"hello"};
```

Initialzing an array of an array of chars

Length of longest char array in array

```
char message[][6] = {
            {"hello"},
            {"there"},
        };
```

Automatically calculated (=2)

# Arrays of chars

Index of array of
chars in array

Printing an array of
chars in the array

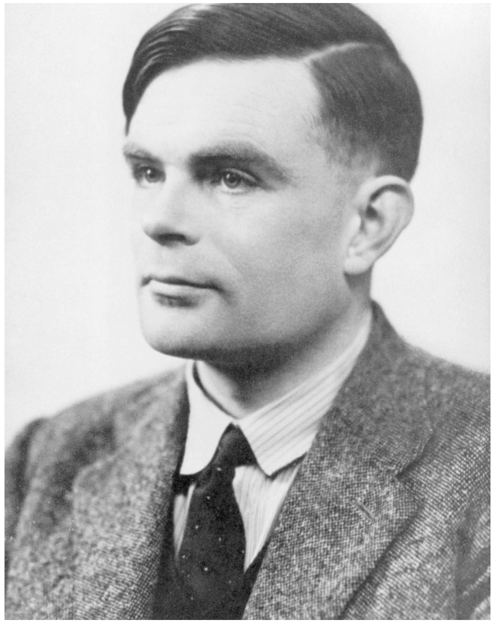
```
Serial.println(message[i]);
```

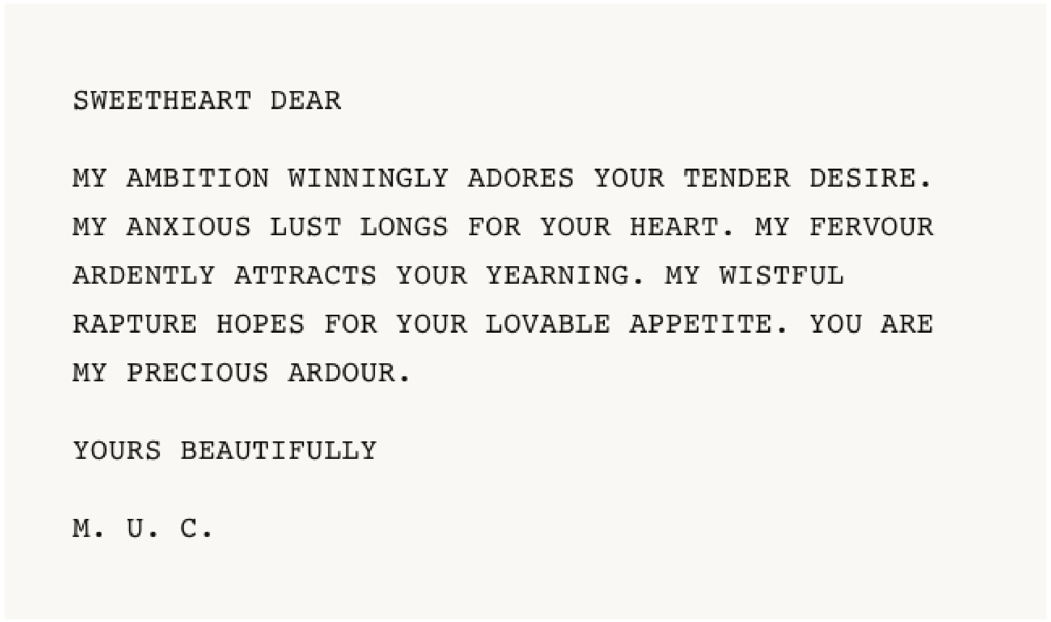
Get number of arrays of
chars in the array

```
(sizeof(message) / sizeof(message[0]));
```

total bytes in array / bytes of first element
= total number of elements

# Love Letter Generator





SWEETHEART DEAR

MY AMBITION WINNINGLY ADORES YOUR TENDER DESIRE. MY ANXIOUS LUST LONGS FOR YOUR HEART. MY FERVOUR ARDENTLY ATTRACTS YOUR YEARNING. MY WISTFUL RAPTURE HOPES FOR YOUR LOVABLE APPETITE. YOU ARE MY PRECIOUS ARDOUR.

YOURS BEAUTIFULLY

M. U. C.

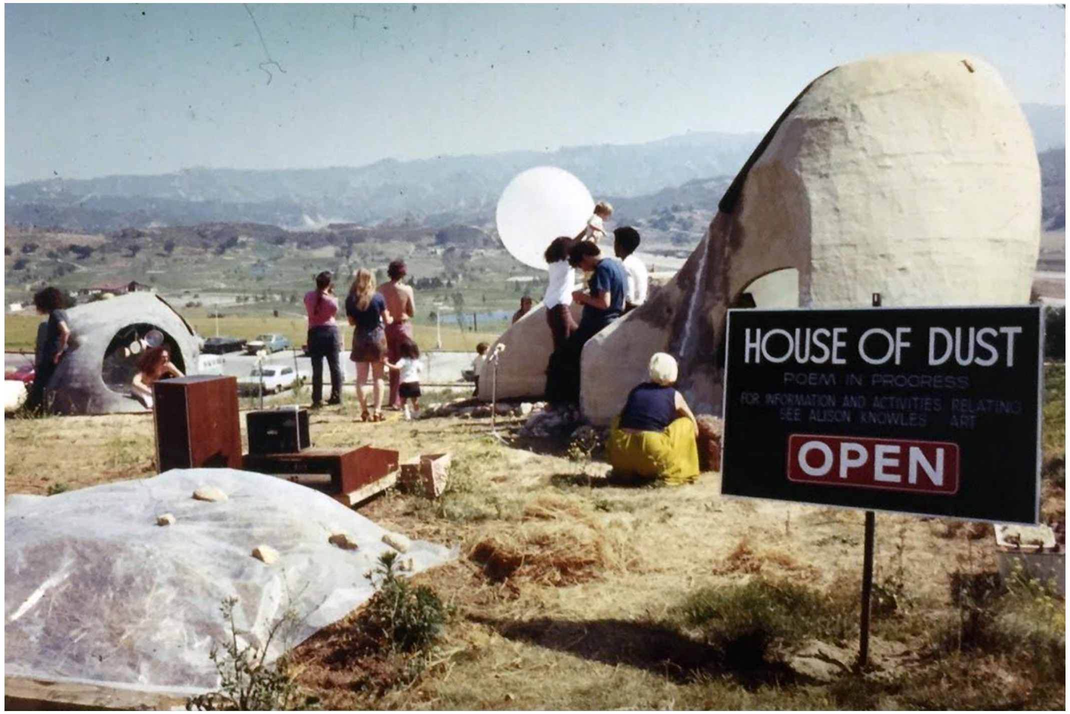
Alan Turing

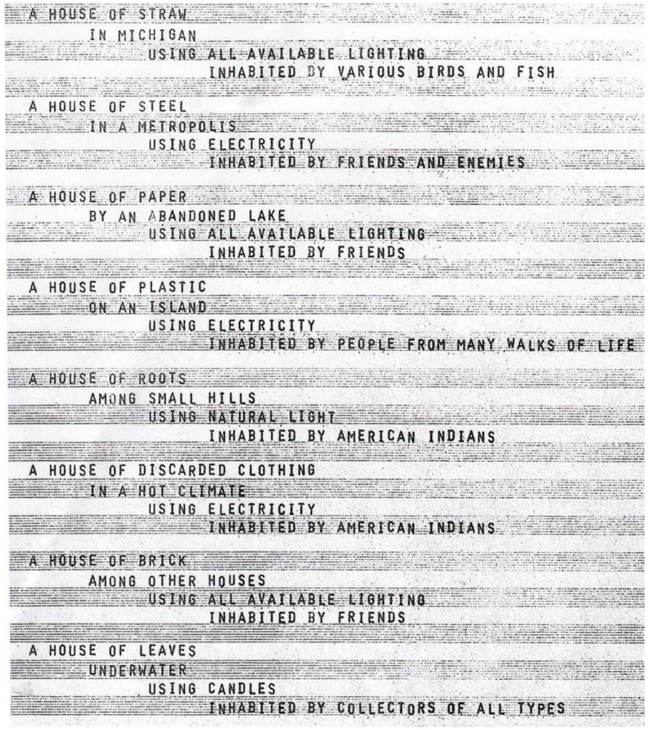
# Love Letter Generator

```
from random import choice
import textwrap

first = ['DARLING', 'DEAR', 'HONEY', 'JEWEL']
second = ['DUCK', 'LOVE', 'MOPPET', 'SWEETHEART']
adjectives = ['ADORABLE', 'AFFECTIONATE', 'AMOROUS', 'ANXIOUS', 'ARDENT', 'AVID', 'BREATHLESS', 'BURNING', 'COVETOUS', 'CRAVING', 'CURIOUS', 'DARLING', 'DEAR', 'DEVOTED', 'EAGER', 'EROTIC', 'FERVENT', 'FOND', 'IMPATIENT', 'KEEN', 'LITTLE',
'LOVEABLE', 'LOVESICK', 'LOVING', 'PASSIONATE', 'PRECIOUS', 'SWEET', 'SYMPATHETIC', 'TENDER', 'UNSATISFIED', 'WISTFUL']
nouns = ['ADORATION', 'AFFECTION', 'AMBITION', 'APPETITE', 'ARDOUR', 'CHARM', 'DESIRE', 'DEVOTION', 'EAGERNESS', 'ENCHANTMENT', 'ENTHUSIASM', 'FANCY', 'FELLOW FEELING', 'FERVOUR', 'FONDNESS', 'HEART', 'HUNGER', 'INFATUATION', 'LIKING', 'LONGING',
'LOVE', 'LUST', 'PASSION', 'RAPTURE', 'SYMPATHY', 'TENDERNESS', 'THIRST', 'WISH', 'YEARNING']
adverbs = ['AFFECTIONATELY', 'ANXIOUSLY', 'ARDENTLY', 'AVIDLY', 'BEAUTIFULLY', 'BREATHLESSLY', 'BURNINGLY', 'COVETOUSLY', 'CURIOUSLY', 'DEVOTEDLY', 'EAGERLY', 'FERVENTLY', 'FONDLY', 'IMPATIENTLY', 'KEENLY', 'LOVINGLY', 'PASSIONATELY', 'SEDUCTIVELY',
'TENDERLY', 'WINNINGLY', 'WISTFULLY']
verbs = ['ADORES', 'ATTRACTS', 'CARES FOR', 'CHERISHES', 'CLINGS TO', 'DESIRES','HOLDS DEAR', 'HOPES FOR', 'HUNGERS FOR', 'IS WEDDED TO', 'LIKES', 'LONGS FOR', 'LOVES', 'LUSTS AFTER', 'PANTS FOR', 'PINES FOR', 'PRIZES', 'SIGHS FOR', 'TEMPTS',
'THIRSTS FOR', 'TREASURES', 'WANTS', 'WISHES', 'WOOS', 'YEARNS FOR']

def maybe(words):
    if choice([False, True]):
        return ' ' + choice(words)
    return ''

def longer():
    return (' MY' + maybe(adjectives) + ' ' + choice(nouns) +
            maybe(adverbs) + ' ' + choice(verbs) + ' YOUR' +
            maybe(adjectives) + ' ' + choice(nouns) + '.')

def shorter():
    return (' ' + choice(adjectives) + ' ' + choice(nouns) + '.')

text = ''
you_are = False
for i in range(0,5):
    type = choice(['longer', 'shorter'])
    if type == 'longer':
        text = text + longer()
        you_are = False
    else:
        if you_are:
            text = text[:-1] + ': MY' + shorter()
            you_are = False
        else:
            text = text + ' YOU ARE MY' + shorter()
            you_are = True

print('')
print(choice(first) + ' ' + choice(second))
print('')
print(textwrap.fill(text, 80))
print('')
print('                                        YOURS ' + choice(adverbs))
print('')
print('                                        M.U.C.')
print('')
```

https://nickm.com/memslam/love_letters.html

# House of Dust


HOUSE OF DUST
POEM IN PROGRESS
FOR INFORMATION AND ACTIVITIES RELATING SEE ALISON KNOWLES ART
OPEN



```
A HOUSE OF STRAW
      IN MICHIGAN
            USING ALL AVAILABLE LIGHTING
                  INHABITED BY VARIOUS BIRDS AND FISH
A HOUSE OF STEEL
      IN A METROPOLIS
            USING ELECTRICITY
                  INHABITED BY FRIENDS AND ENEMIES
A HOUSE OF PAPER
      BY AN ABANDONED LAKE
            USING ALL AVAILABLE LIGHTING
                  INHABITED BY FRIENDS
A HOUSE OF PLASTIC
      ON AN ISLAND
            USING ELECTRICITY
                  INHABITED BY PEOPLE FROM MANY WALKS OF LIFE
A HOUSE OF ROOTS
      AMONG SMALL HILLS
            USING NATURAL LIGHT
                  INHABITED BY AMERICAN INDIANS
A HOUSE OF DISCARDED CLOTHING
      IN A HOT CLIMATE
            USING ELECTRICITY
                  INHABITED BY AMERICAN INDIANS
A HOUSE OF BRICK
      AMONG OTHER HOUSES
            USING ALL AVAILABLE LIGHTING
                  INHABITED BY FRIENDS
A HOUSE OF LEAVES
      UNDERWATER
            USING CANDLES
                  INHABITED BY COLLECTORS OF ALL TYPES
```

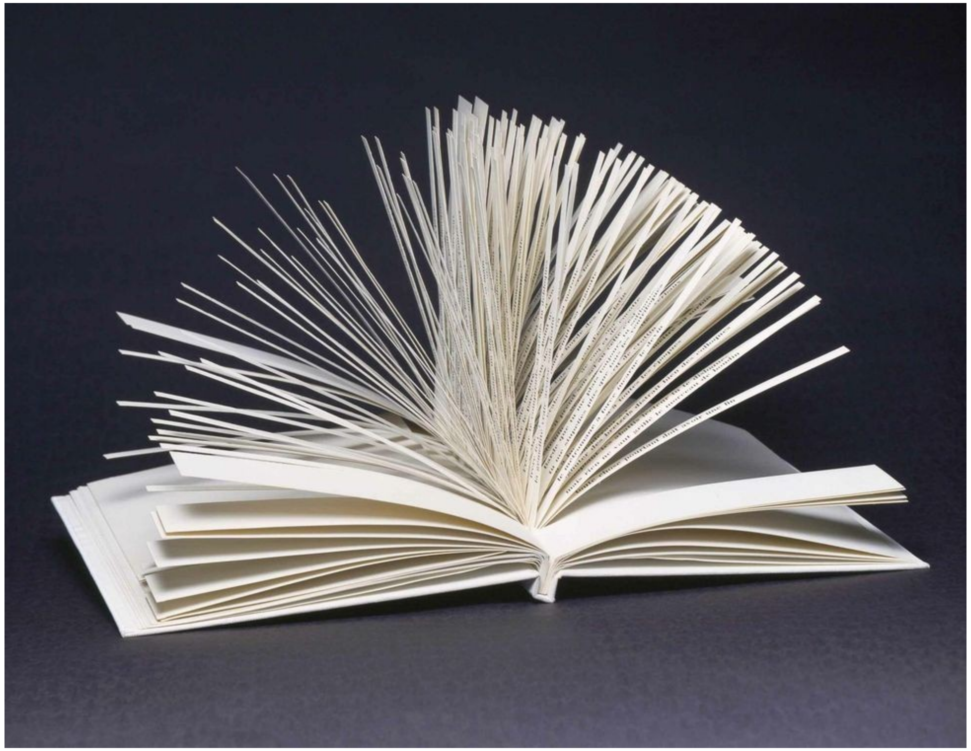
Alison Knowles

# House of Dust

```
from random import choice

material = ['SAND', 'DUST', 'LEAVES', 'PAPER', 'TIN', 'ROOTS', 'BRICK', 'STONE', 'DISCARDED CLOTHING', 'GLASS', 'STEEL',
'PLASTIC', 'MUD', 'BROKEN DISHES', 'WOOD', 'STRAW', 'WEEDS']

location = ['IN A GREEN, MOSSY TERRAIN', 'IN AN OVERPOPULATED AREA', 'BY THE SEA', 'BY AN ABANDONED LAKE', 'IN A DESERTED
FACTORY', 'IN DENSE WOODS', 'IN JAPAN', 'AMONG SMALL HILLS', 'IN SOUTHERN FRANCE', 'AMONG HIGH MOUNTAINS', 'ON AN ISLAND', 'IN A
COLD, WINDY CLIMATE', 'IN A PLACE WITH BOTH HEAVY RAIN AND BRIGHT SUN', 'IN A DESERTED AIRPORT', 'IN A HOT CLIMATE', 'INSIDE A
MOUNTAIN', 'ON THE SEA', 'IN MICHIGAN', 'IN HEAVY JUNGLE UNDERGROWTH', 'BY A RIVER', 'AMONG OTHER HOUSES', 'IN A DESERTED
CHURCH', 'IN A METROPOLIS', 'UNDERWATER']

light_source = ['CANDLES', 'ALL AVAILABLE LIGHTING', 'ELECTRICITY', 'NATURAL LIGHT']

inhabitants = ['PEOPLE WHO SLEEP VERY LITTLE', 'VEGETARIANS', 'HORSES AND BIRDS', 'PEOPLE SPEAKING MANY LANGUAGES WEARING LITTLE
OR NO CLOTHING', 'ALL RACES OF MEN REPRESENTED WEARING PREDOMINANTLY RED CLOTHING', 'CHILDREN AND OLD PEOPLE', 'VARIOUS BIRDS AND
FISH', 'LOVERS', 'PEOPLE WHO ENJOY EATING TOGETHER', 'PEOPLE WHO EAT A GREAT DEAL', 'COLLECTORS OF ALL TYPES', 'FRIENDS AND
ENEMIES', 'PEOPLE WHO SLEEP ALMOST ALL THE TIME', 'VERY TALL PEOPLE', 'AMERICAN INDIANS', 'LITTLE BOYS', 'PEOPLE FROM MANY WALKS
OF LIFE', 'NEGROS WEARING ALL COLORS', 'FRIENDS', 'FRENCH AND GERMAN SPEAKING PEOPLE', 'FISHERMEN AND FAMILIES', 'PEOPLE WHO LOVE
TO READ']

print('')
print('A HOUSE OF ' + choice(material))
print('      ' + choice(location))
print('            USING ' + choice(light_source))
print('                  INHABITED BY ' + choice(inhabitants))
print('')
```

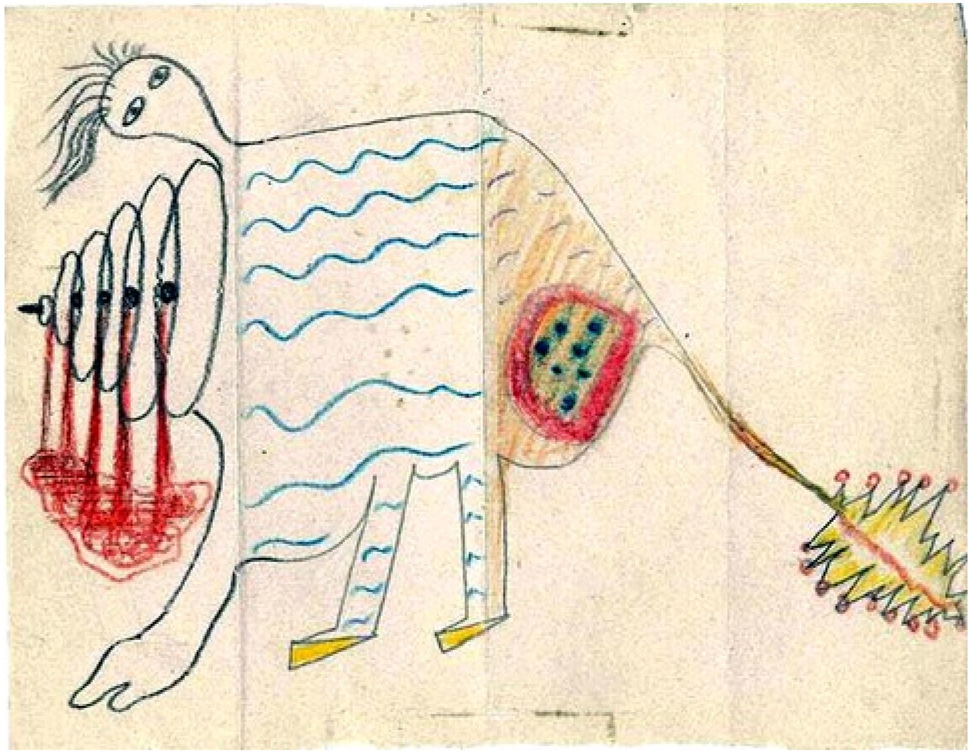
https://nickm.com/memslam/a_house_of_dust.html

# Cent mille milliards de poèmes



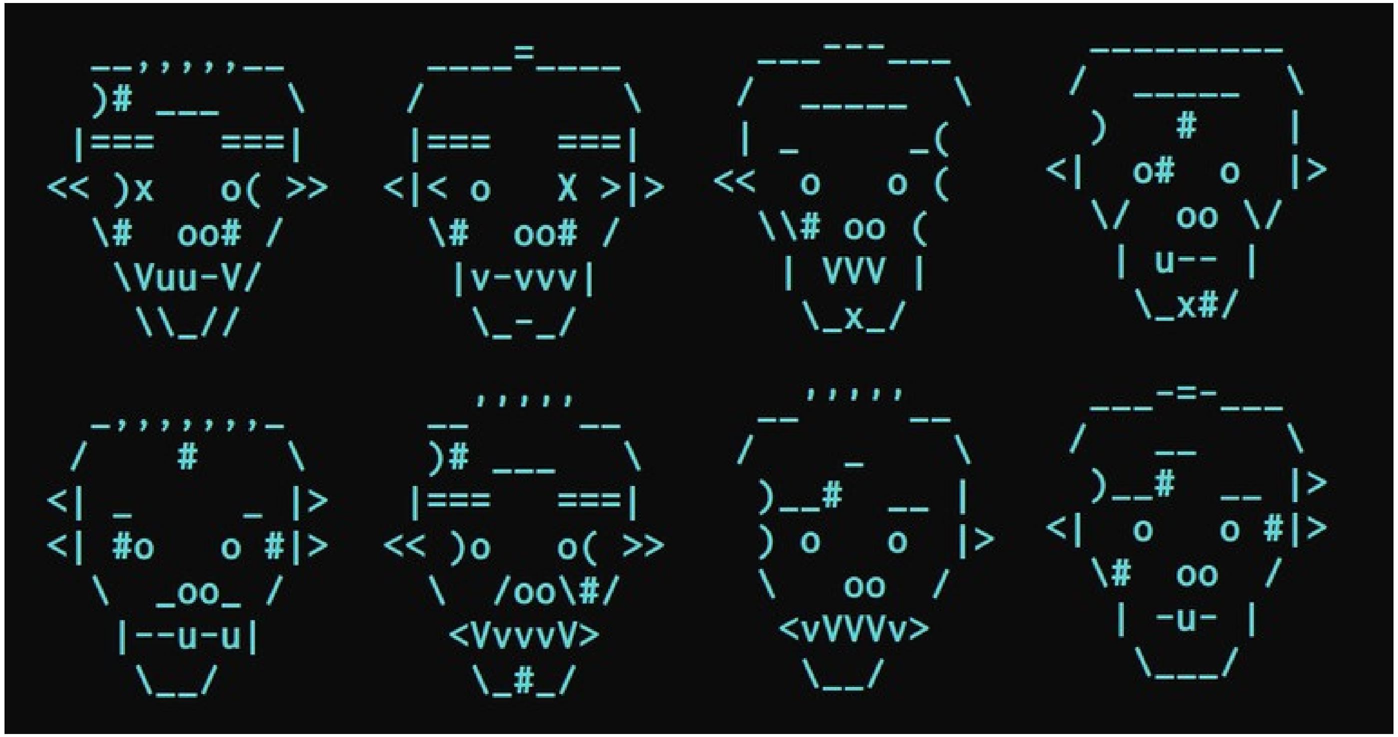
Raymond Queneau

# Cadavre exquis



Jacques Prévert

# Warsim



```
  __,,,,,__
  )# ___   \
 |===   ===|
<< )x   o( >>
  \#  oo# /
   \Vuu-V/
    \\_//
```

```
      =
  ____ ____
 /         \
 |===   ===|
<|< o   X >|>
  \#  oo# /
   |v-vvv|
    \_-_/
```

```
     ---
  ___   ___
 /   _____ \
 | _     _(
<<  o   o (
  \\# oo (
   | VVV |
    \_x_/
```

```
  _________
 /   _____ \
  )   #    |
<|  o#  o  |>
  \/  oo \/
   | u-- |
    \_x#/
```

```
  _,,,,,,,_
 /    #    \
<| _     _ |>
<| #o   o #|>
  \  _oo_ /
   |--u-u|
    \__/
```

```
    ,,,,,
  __     __
  )# ___   \
 |===   ===|
<< )o   o( >>
  \  /oo\#/
   <VvvvV>
    \_#_/
```

```
    ,,,,,
  __     __
 /    _    \
  )__#   __ |
  ) o   o  |>
  \   oo  /
   <vVVVv>
    \__/
```

```
     -=-
  ___   ___
 /   __    \
  )__#  __ |>
<|  o   o #|>
  \#  oo  /
   | -u- |
    \___/
```

https://www.youtube.com/watch?v=IEJElbZuHAU&ab_channel=VegaVideoNetwork

# random()

```
for (int i = 0; i < 5; i++)
// prints 5 pseudo-random numbers picked between 1-100
{
  int x = random(100);
  Serial.println(x);
}
```

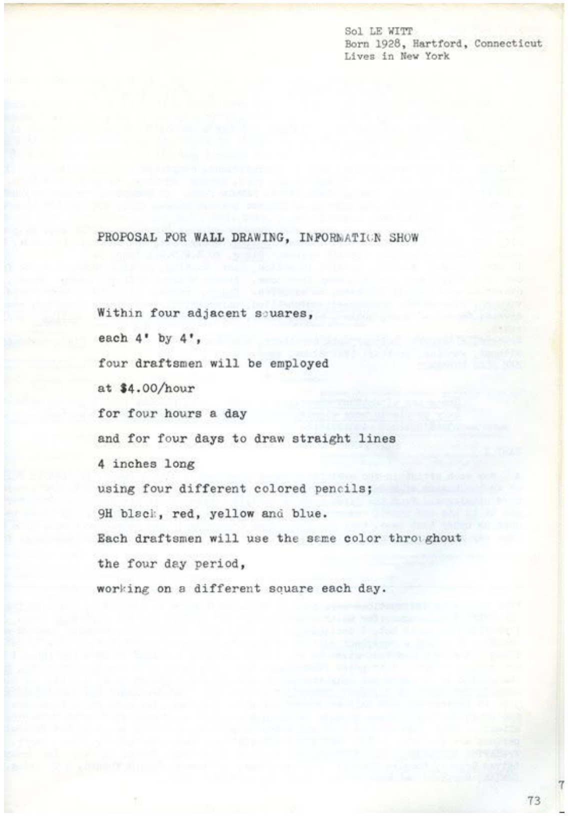
# randomSeed()

```
long randNumber;

void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
}

void loop() {
  randNumber = random(100);
  Serial.println(randNumber);
  delay(50);
}
```

# random string

```
char First [][30] = {
 { "You walk forward." },
 { "You hear a sound." },
 { "You turn to look behind you." },
 { "You open your backpack." },
 };

void setup() {

  Serial.begin (9600);
  randomSeed(analogRead(0));
}

void loop() {

  int randex = random(sizeof(First) / sizeof(First[0])); //random index in bounds
  Serial.println(First[randex]);
}
```

# Wall Drawings Instructions

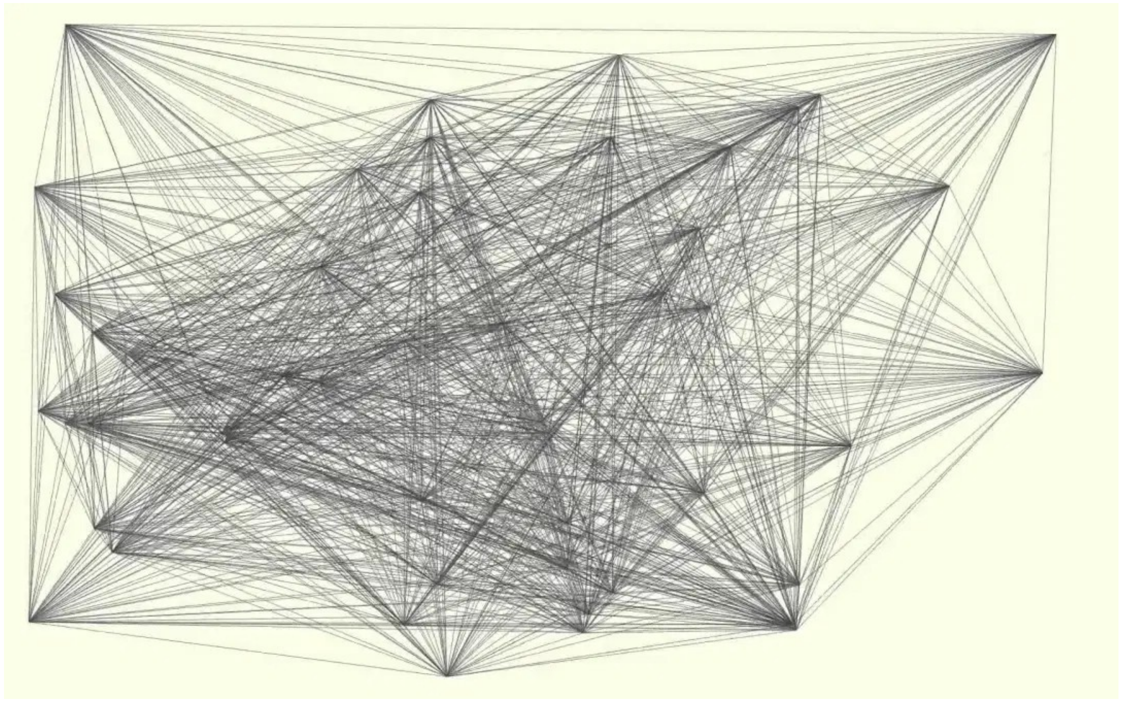


Sol LE WITT
Born 1928, Hartford, Connecticut
Lives in New York

PROPOSAL FOR WALL DRAWING, INFORMATION SHOW

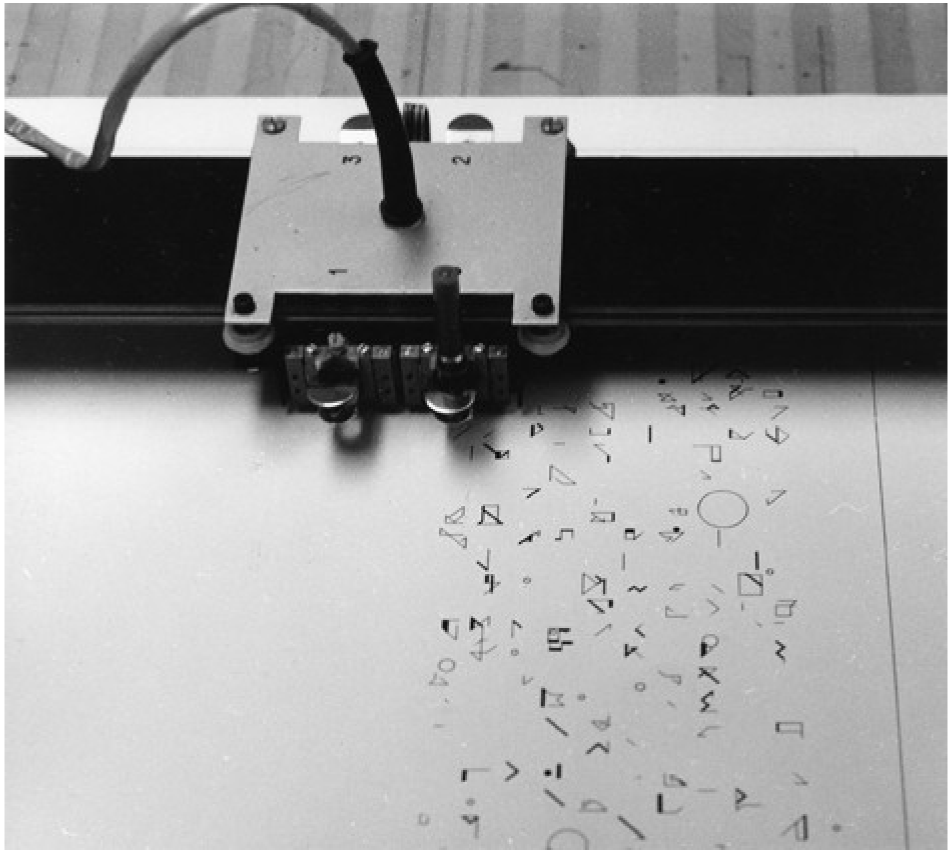
Within four adjacent squares,
each 4' by 4',
four draftsmen will be employed
at $4.00/hour
for four hours a day
and for four days to draw straight lines
4 inches long
using four different colored pencils;
9H black, red, yellow and blue.
Each draftsmen will use the same color throughout
the four day period,
working on a different square each day.

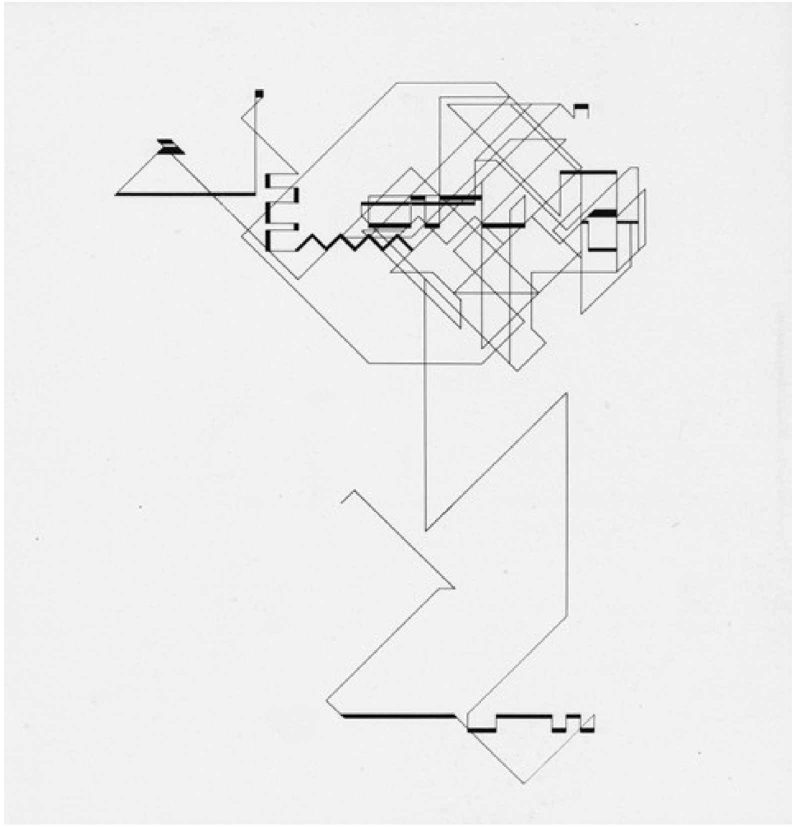
73



Sol Lewitt

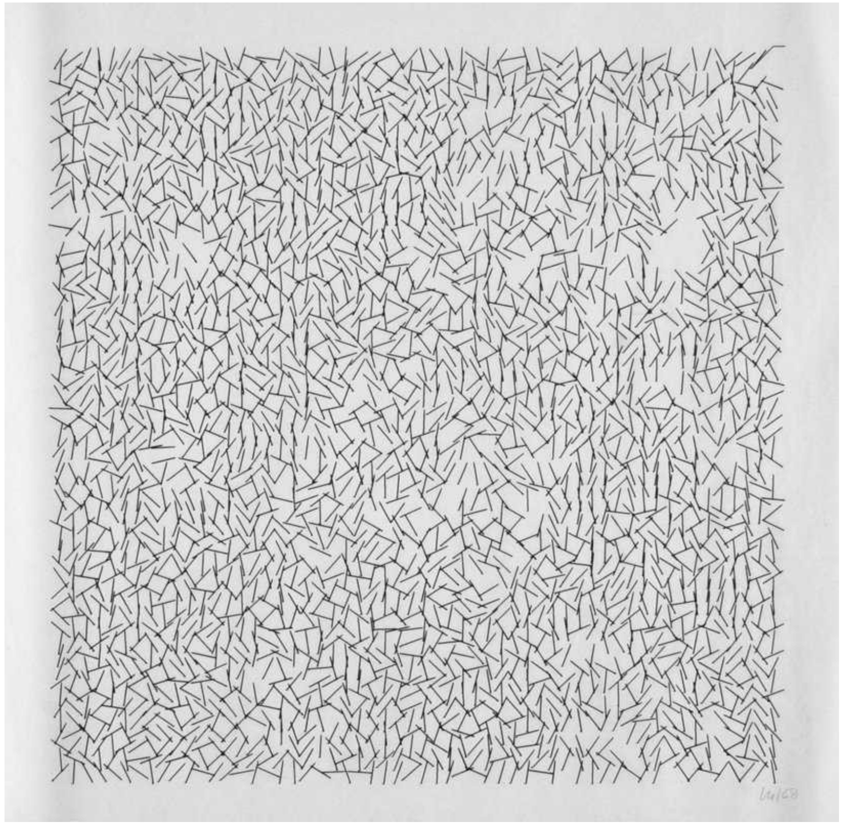
# Plotter Drawings

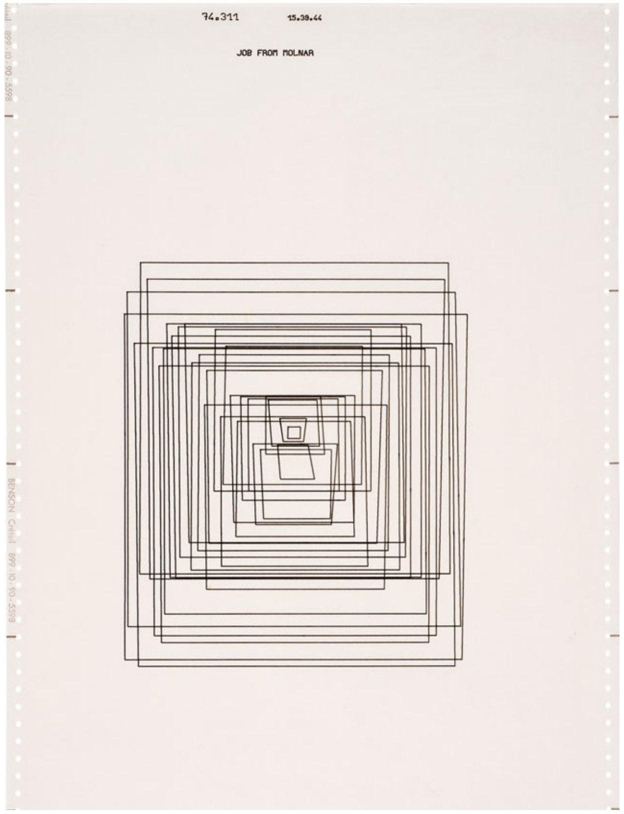




Manfred Mohr

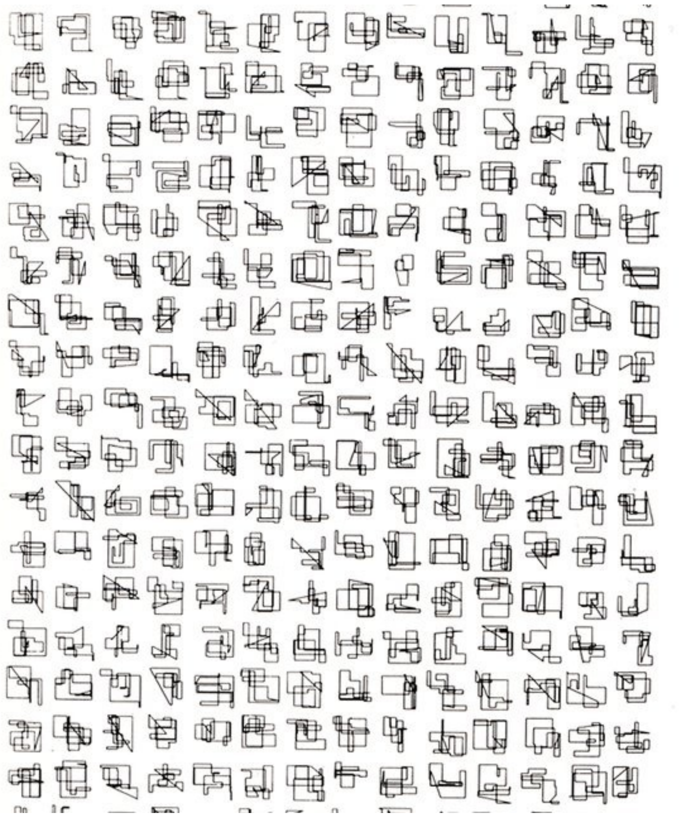
# Plotter Drawings

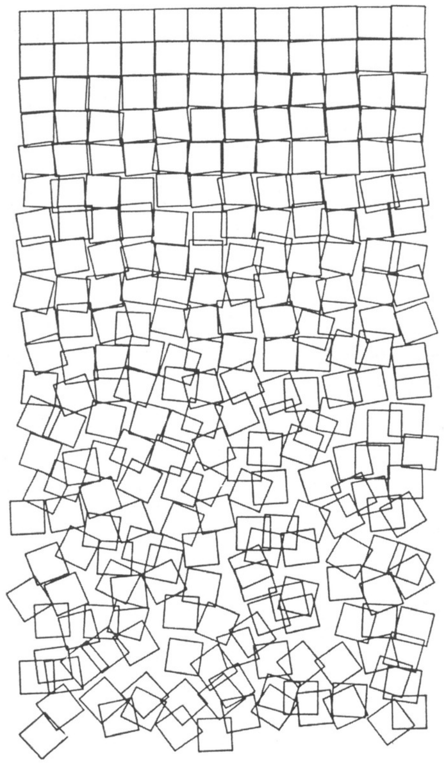




Vera Molnár

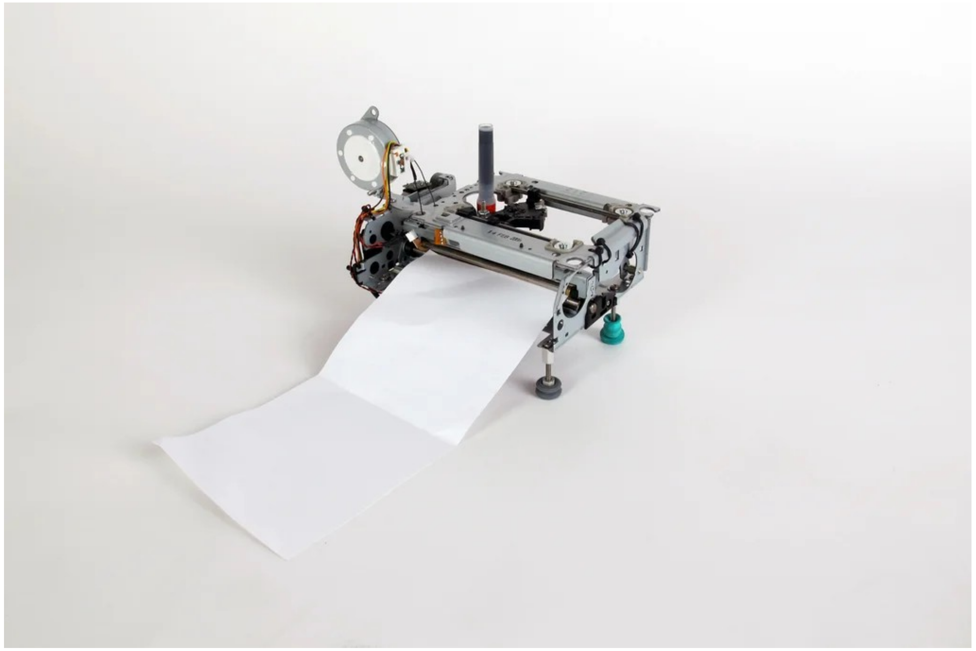
# Plotter Drawings

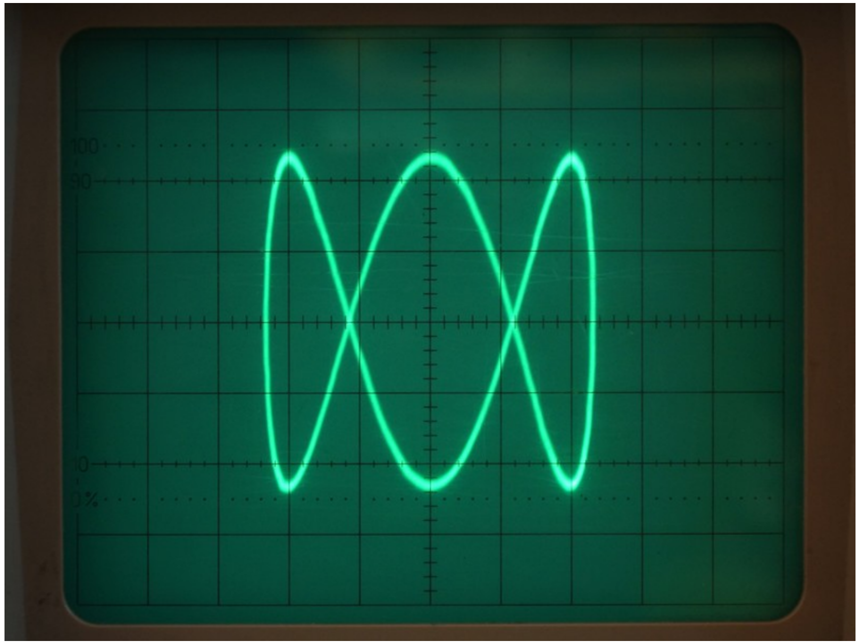




Georg Nees

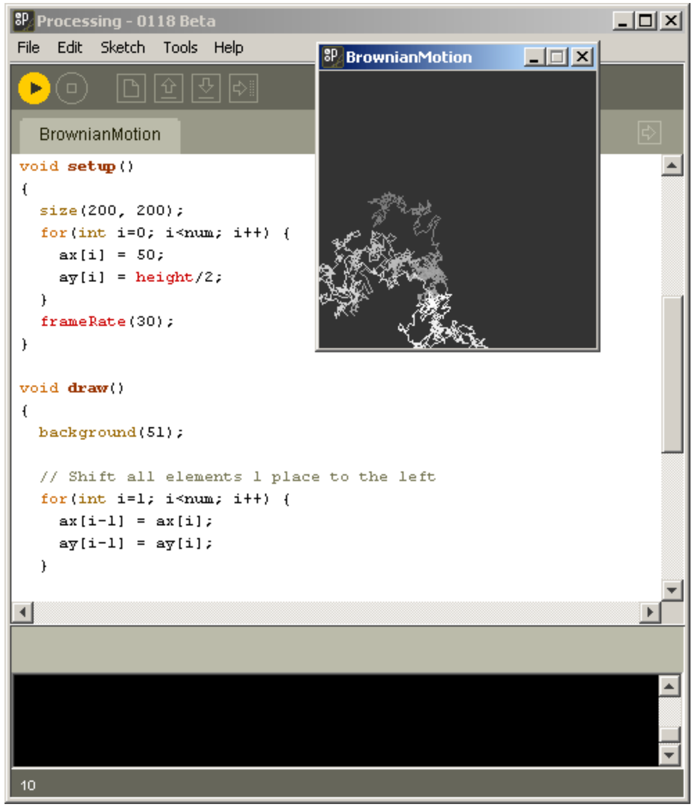
# Arduino controlled XY plotters





## Processing

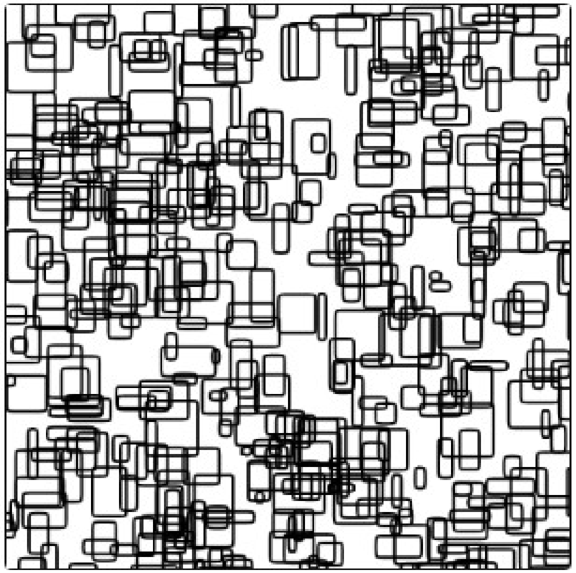
C++ like Arduino
For 2D visualizations



```
void setup()
{
  size(200, 200);
  for(int i=0; i<num; i++) {
    ax[i] = 50;
    ay[i] = height/2;
  }
  frameRate(30);
}

void draw()
{
  background(51);

  // Shift all elements 1 place to the left
  for(int i=1; i<num; i++) {
    ax[i-1] = ax[i];
    ay[i-1] = ay[i];
  }
```

# Plotter Drawings Code

http://recodeproject.com/



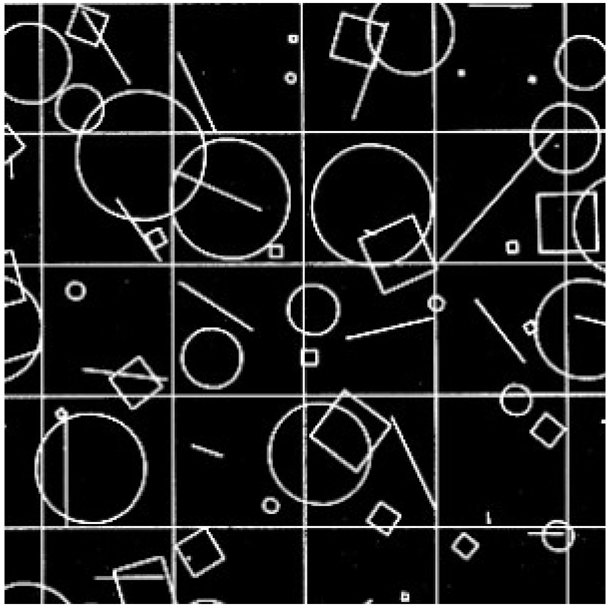
Chris Allick

```
...

void drawRects() {

for( int i = 0; i < 400; i++ ) {

RoundedRect(
random(-10,width),
random(-10,height),
random( 5, 40),
random(5,40), 5, 5 );
}
}

...
```

http://recodeproject.com/translation/chris-allick-direct-random-squares-charles-csuri

# Plotter Drawings Code

http://recodeproject.com/

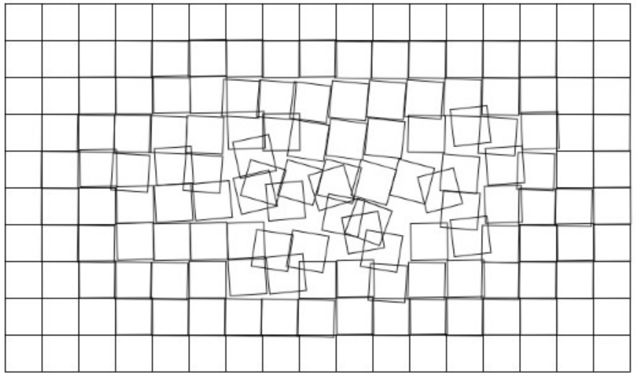

Aaron Marcus

```
...

    //generate random seed values for location and size
    float randLoc = random(-gridSize/2,gridSize/2);
    float randLoc2 = random(-gridSize/2,gridSize/2);
    float randLoc3 = random(-gridSize/2,gridSize/2);
    float randLoc4 = random(-gridSize/2,gridSize/2);
    float randLoc5 = random(-gridSize/2,gridSize/2);
    float randLoc6 = random(-gridSize/2,gridSize/2);
    float randLoc7 = random(-gridSize/2,gridSize/2);
    float randLoc8 = random(-gridSize/2,gridSize/2);
    float sqSize = random(0, (gridSize-10)/2);
...

    //draw squares
    translate(x+randLoc3, y+randLoc4);
    rotate(random(TWO_PI));
    rect(0, 0, sqSize, sqSize);

...
```

http://recodeproject.com/artwork/v3n2untitled-2-marcus

# Plotter Drawings Code

http://recodeproject.com/

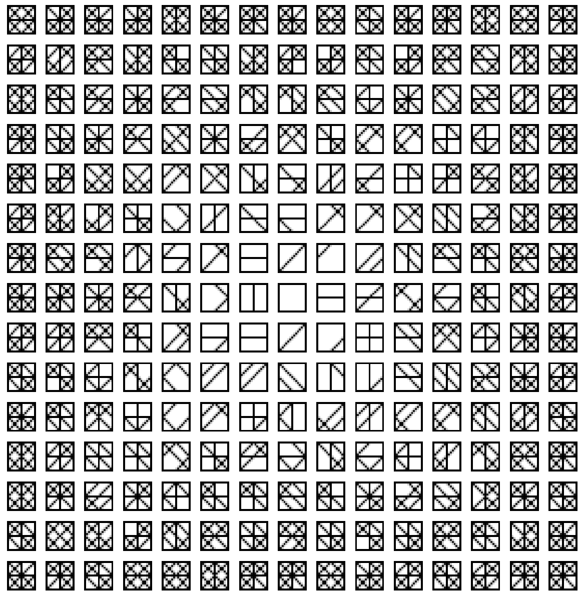

Sermad Buni

```
...

if(canvasheight % 2 == 0) {
      iw = canvaswidth/2 - Math.abs(i - canvaswidth/2);
    } else {
      iw = canvaswidth/2 - 0.5 - Math.abs(i - canvaswidth/2 - 0.5);
    }

    if(canvasheight % 2 == 0) {
      jh = canvasheight/2 - 0.5 - Math.abs(j - canvasheight/2 + 0.5);
    } else {
      jh = canvasheight/2 - Math.abs(j - canvasheight/2);
    }

    // check if we are not drawing the outer edges
    if( jh != 0 || iw != 0) {

      // rotate the square
      rotate( radians(iw * iw * jh * jh * random(-randomness,randomness)) );

...
```

http://recodeproject.com/translation/sermad-buni-direct-boxes-i-william-kolomyjec

# Plotter Drawings Code

http://recodeproject.com/



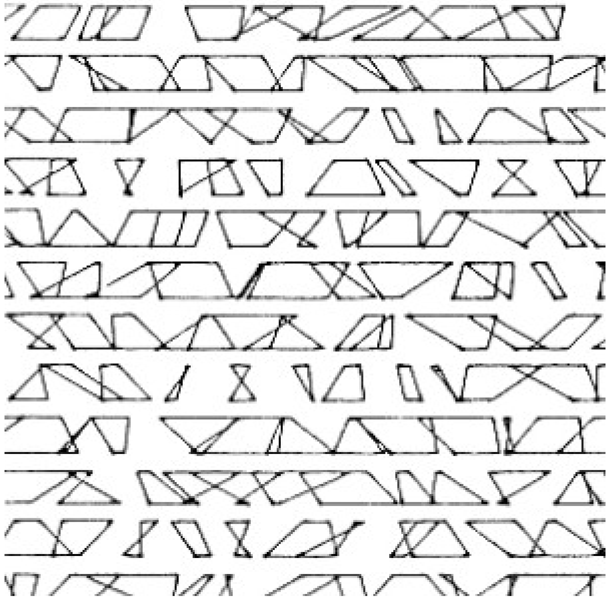
Greg Borenstein

```
// do the strokes in a random different order each time
Collections.shuffle(strokes);

for (int i = 0; i < distFromMiddle; i++) {
drawSegment( strokes.get(i), boxSize, boxSize);
}
…

void drawSegment(int i, int w, int h) {
  switch(i) {
  case 0:
    line(0, 0, w, h);
    break;
  case 1:
    line(w, 0, 0, h);
    break;
  case 2:
    line(0, h/2, w, h/2);
    break;
  case 3:
    line(0, h/2, w/2, 0);
    break;
  case 4:
    line(w/2, 0, w, h/2);
    break;
…
```

http://recodeproject.com/translation/greg-borenstein-direct-structure-square-series-inwards-roger-coqart

# Plotter Drawings Code

http://recodeproject.com/



Vera Molnar

```
...
  //for each grid cell...
  for(int i = 0, gi = gutterSize; i < numTiles; i++, gi +=
gutterSize+tileSize){
    for(int j = 0, gj = gutterSize; j < numTiles; j++, gj +=
gutterSize+tileSize){
      drawTrapezium(gi+tileSize/2, gj+tileSize/2);
    }
  }
}

void drawTrapezium(float xCenter, float yCenter){
  float topScale = random(-2, 2);
  float bottomScale = random(-2, 2);
  float halfTile = tileSize/2.;
  quad(xCenter - tileSize/2 + random(-tileSize, tileSize),
yCenter - halfTile,
        xCenter + tileSize/2 + random(-tileSize, tileSize),
yCenter - halfTile,
        xCenter + tileSize/2 + random(-tileSize, tileSize),
yCenter + halfTile,
        xCenter - tileSize/2 + random(-tileSize, tileSize),
yCenter + halfTile);
}
...
```

http://recodeproject.com/translation/quin-kennedy-direct-unimaginable-images-vera-molnar

# Math functions

```
float x,y;
double z;

z = pow(x, y);//(base, exp)

z = sq(x);
z = sqrt(x);

z = cos(x); // in radians
z = sin(x); // in radians
z = tan(x); // in radians
```

**Float** is a datatype for:

A number that has a decimal point.

Can store: 3.4028235E+38 to -3.4028235E+38.

6-7 decimal digits of precision (total number of digits, not the number to the right of the decimal point.)

Unlike other platforms, where you can get more precision by using a double (e.g. up to 15 digits), on the Arduino, **double** is the same size as float.

One **radian** is 180/π degrees or just under 57.3°

# Curve Drawing Machines

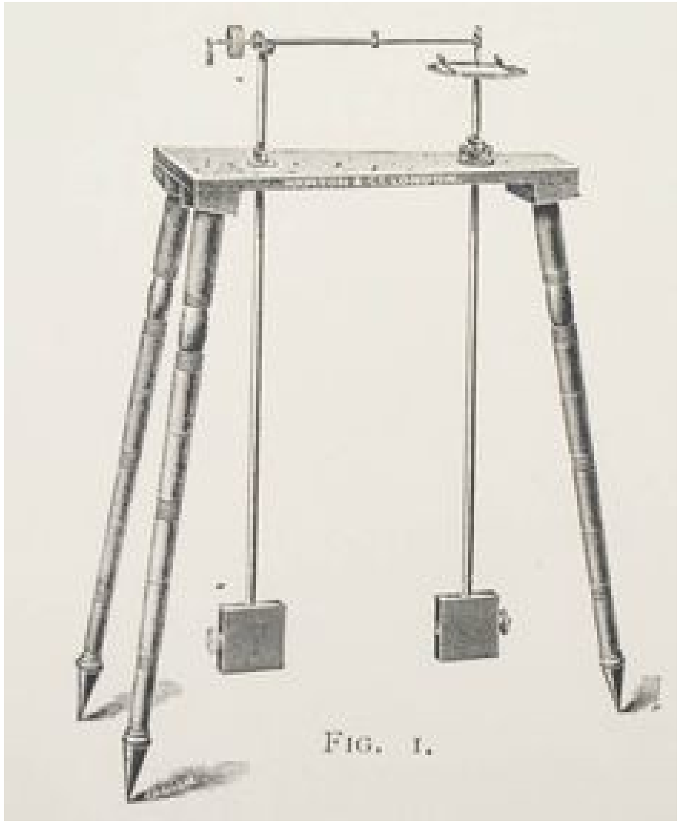
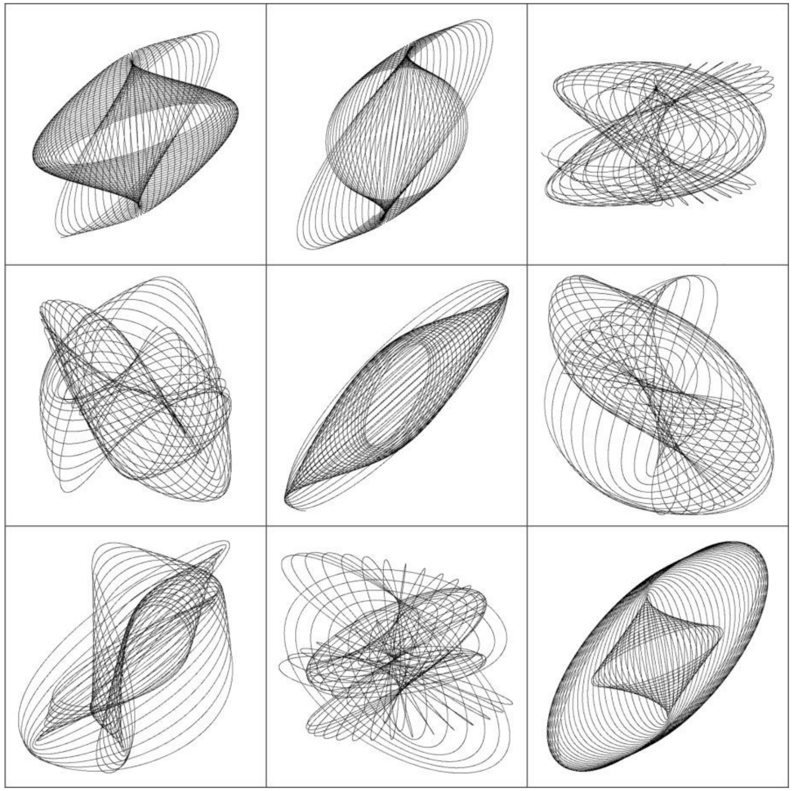
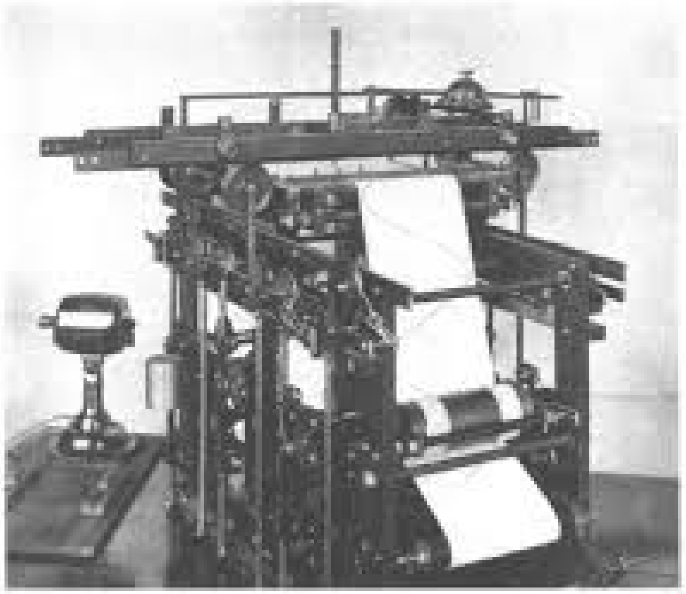
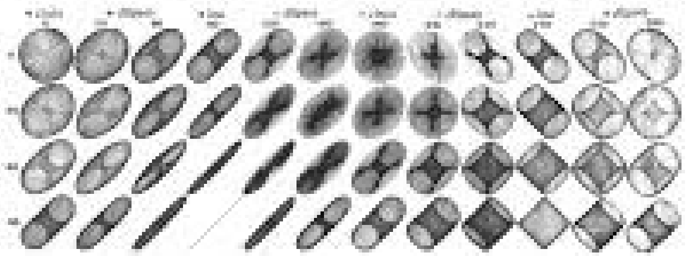


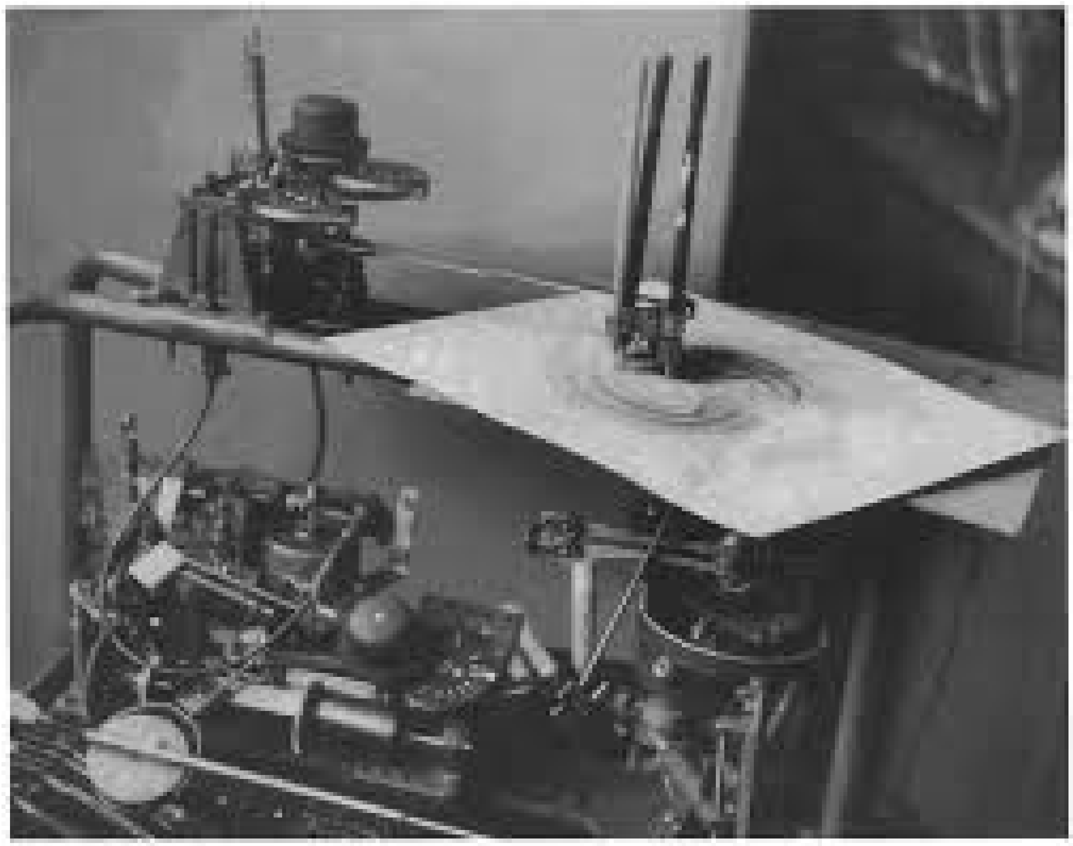
FIG. 1.



# Curve Drawing Machines







# Curve Drawing Machines

$$x(t) = x_1 + \frac{a}{d}\left(x_2 - x_1\right) - \frac{h}{d}\left(y_2 - y_1\right)$$

$$y(t) = y_1 + \frac{a}{d}\left(y_2 - y_1\right) + \frac{h}{d}\left(x_2 - x_1\right)$$

$$\text{where } x_1 = r_1\cos(\alpha t + \phi) + C_{1_x},$$

$$y_1 = r_1\sin(\alpha t + \phi) + C_{1_y},$$

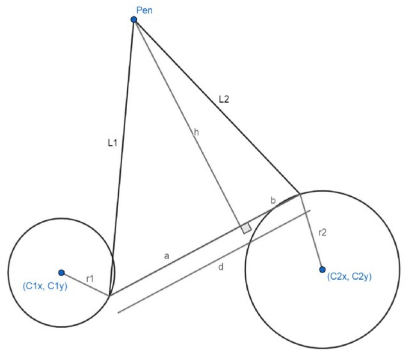
$$x_2 = r_2\cos(\beta t) + C_{2_x},$$

$$y_2 = r_2\sin(\beta t) + C_{2_y},$$

$$d = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2},$$

$$a = \frac{d^2 + l_1{}^2 - l_2{}^2}{2d},$$

$$\text{and } h = \sqrt{l_1{}^2 - a^2}$$



C = constants for positioning the circles

α, β = speeds of the circles

Φ = offset in starting position

r1, r2 = radii of the circles

L1, L2 = lengths of the arms

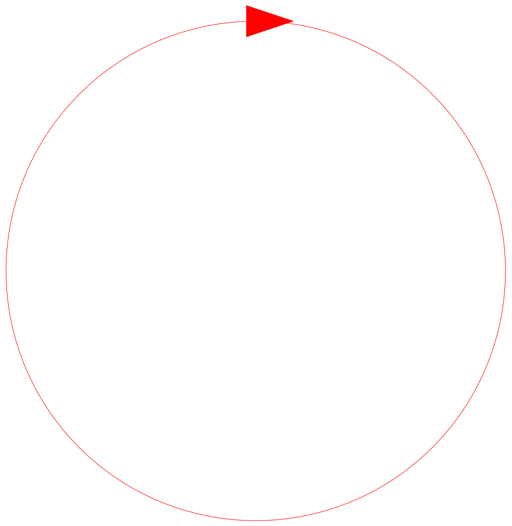
(x1, y1), (x2, y2) = coordinates of the "pivots" where the arms attach to the circles

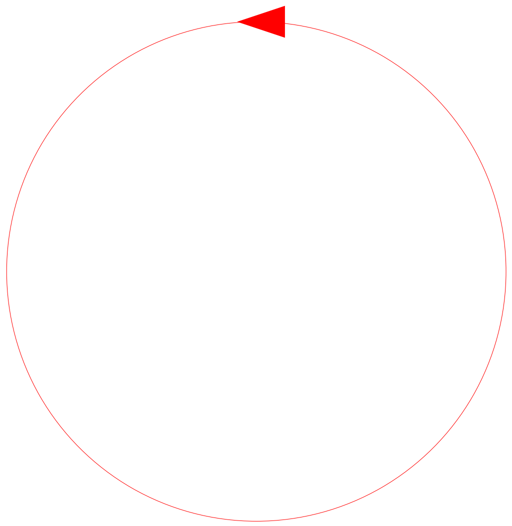
d = length between the pivots

h = altitude of the triangle formed with the arms and d

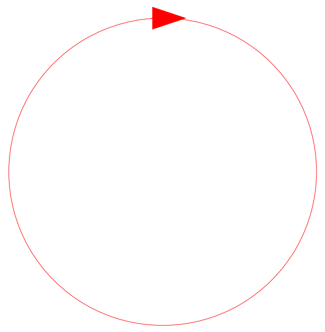
a = length between the first pivot and the point where h meets d

# Drawing Machines

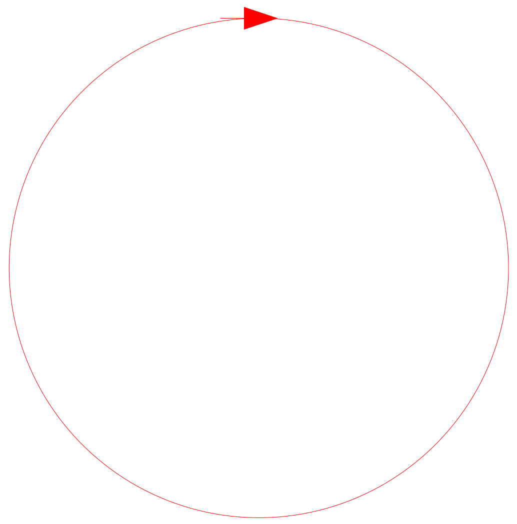

CW


CCW

# Drawing Machines

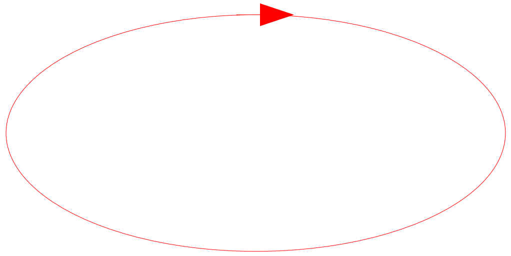


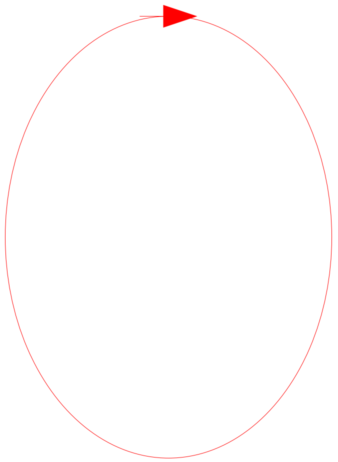
R = 1

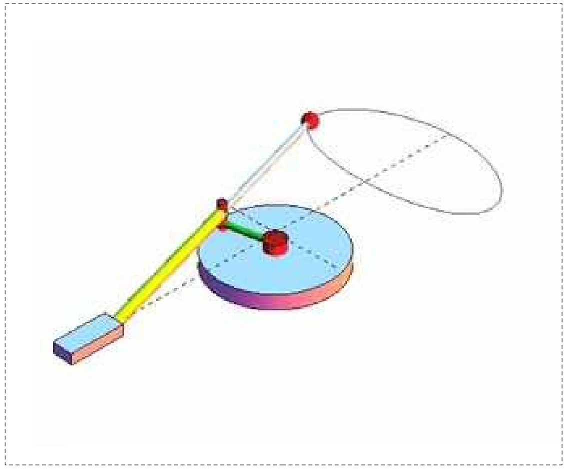


R = 2

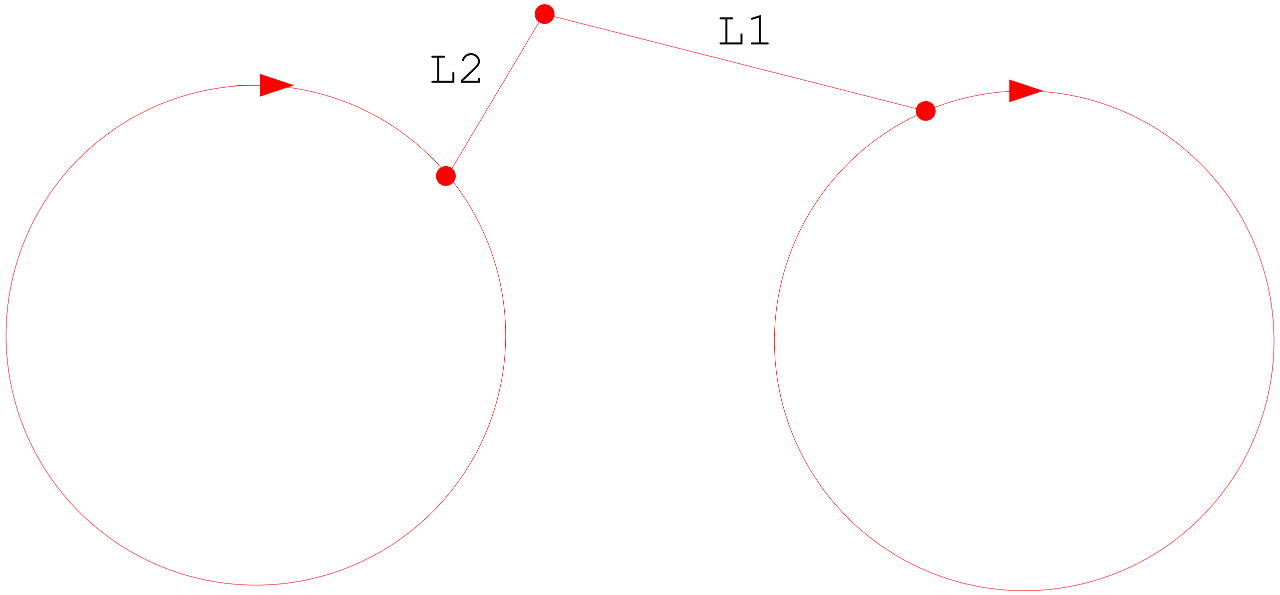
# Drawing Machines

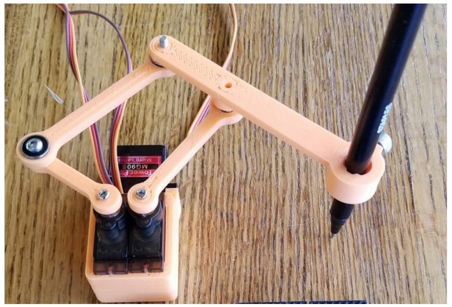




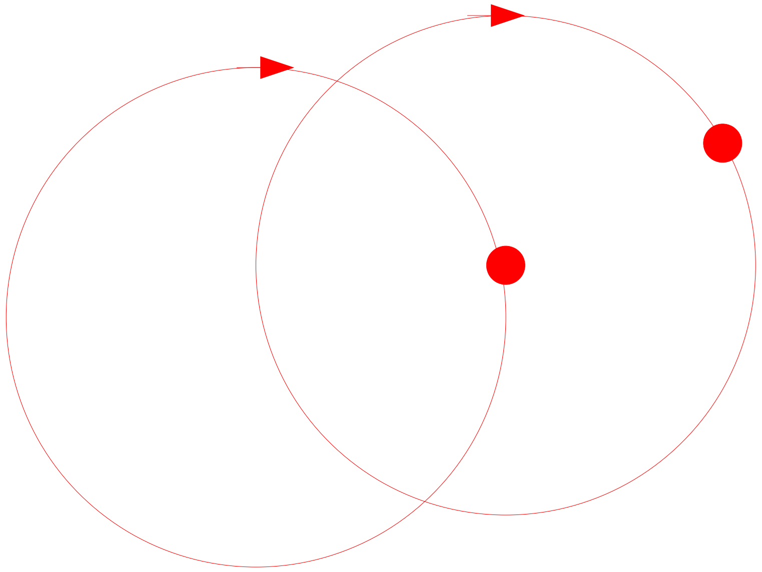


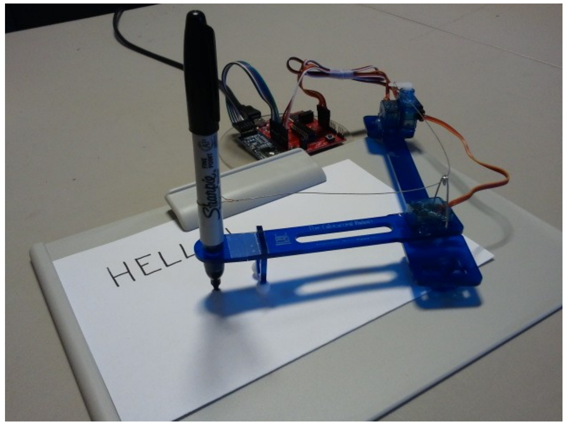
# Drawing Machines



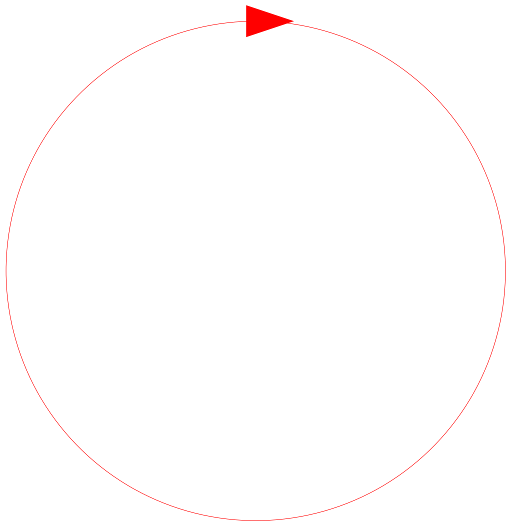


# Drawing Machines

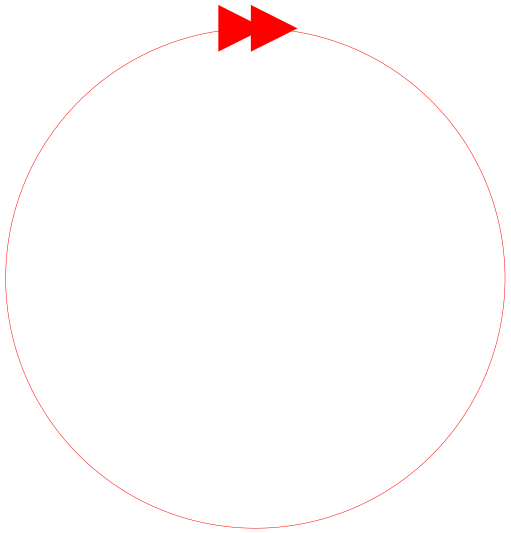




# Drawing Machines

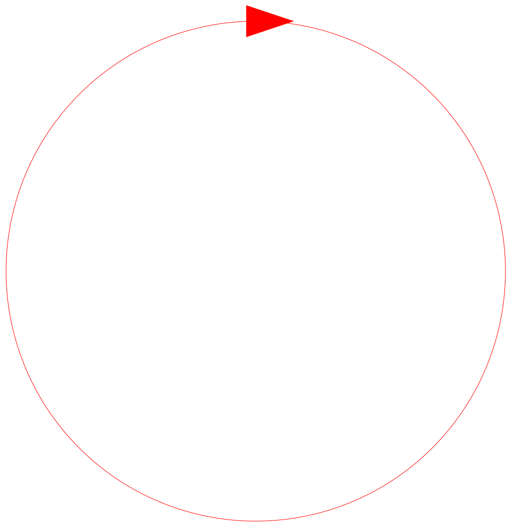


Speed = 1rpm

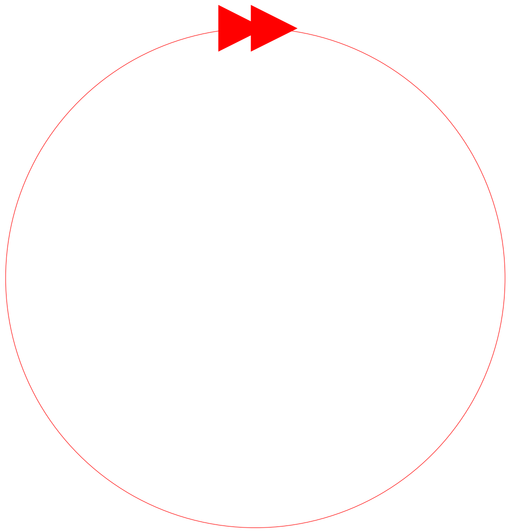


Speed = 2rpm

# Drawing Machines



Speed = 1/time



Speed = constant