4. Computer Code

```
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
/*  Aaron Romain's head spinner  */
/*  Enter how large you'd like the image,
    how many iterations to make, and what to name the file  */

FILE *fp;
int imWidth = 700;
int imHeight = 900;
int n = 1;
char fileName[] = "agromain.svg";
int step = 30;
int ring = 10;
float xc = 400;
float yc = 600;
float radius = 200;
int i;

struct Stroke
{
    int r;
    int g;
    int b;
    int width;
    float opacity;
};

struct Fill
{
    int r;
    int g;
    int b;
    float opacity;
};

```

```
39  struct Lines
40  {
41      float x1[1000];
42      float x2[1000];
43      float y1[1000];
44      float y2[1000];
45  };
46
47  struct Circles
48  {
49      float x[100];
50      float y[100];
51      float r[100];
52  };
53
54  void setStroke(struct Stroke *myStroke)
55  {
56      myStroke->r = 0;
57      myStroke->g = 0;
58      myStroke->b = 0;
59      myStroke->width = 1;
60      myStroke->opacity = 1;
61  }
62
63  void setFill(struct Fill *myFill)
64  {
65      myFill->r = 0;
66      myFill->g = 0;
67      myFill->b = 0;
68      myFill->opacity = 0;
69  }
70
```

```
71  void romain_aaron_getData(struct Lines *myLines, struct Circles *myCircle,
72                            float cx_in, float cy_in, float r_in,
73                            int steps, int rings)
74  {
75      float v[2][1] = {0,r_in};
76      float v2[2][1] = {v[0][0],v[1][0]};
77      float theta = 2*M_PI/steps;
78      float R[2][2] = {{cos(theta),-sin(theta)},{sin(theta),cos(theta)}};
79      int i;
80      myLines->x1[0] = cx_in;
81      myLines->y1[0] = cy_in;
82      for (i=0;i<steps;i++)
83      {
84          v2[0][0] = v[0][0];
85          v2[1][0] = v[1][0];
86          myLines->x2[i] = v[0][0]+cx_in;
87          myLines->y2[i] = v[1][0]+cy_in;
88          v[0][0] = R[0][0]*v2[0][0]+R[0][1]*v2[1][0];
89          v[1][0] = R[1][0]*v2[0][0]+R[1][1]*v2[1][0];
90      }
91      myCircle->x[0] = cx_in;
92      myCircle->y[0] = cy_in;
93      for (i=0;i<rings;i++)
94      {
95          myCircle->r[i] = r_in*powf(.8,i);
96      }
97  }
98
99  void drawLine(float x1, float y1, float x2, float y2,
100                 int stroke_r, int stroke_g, int stroke_b,
101                 float stroke_opacity, int stroke_width)
102 {
103     fprintf(fp,"\n    <line x1 = '%f' y1 = '%f' x2 = '%f' y2 = '%f'",
104             x1, y1, x2, y2);
105     fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'",
106             stroke_r, stroke_g, stroke_b, stroke_opacity);
107     fprintf(fp," stroke-width = '%d' />", stroke_width);
108 }
109
```

```
110 void drawCircle(float cx, float cy, float r,
111                  int fill_r, int fill_g, int fill_b,
112                  float fill_opacity, int stroke_r, int stroke_g,
113                  int stroke_b, float stroke_opacity, int stroke_width)
114 {
115     fprintf(fp,"\n    <circle cx = '%f' cy = '%f' r = '%f'", cx, cy, r);
116     fprintf(fp," fill = 'rgb(%d, %d, %d)' fill-opacity = '%f'",
117             fill_r, fill_g, fill_b, fill_opacity);
118     fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'",
119             stroke_r, stroke_g, stroke_b, stroke_opacity);
120     fprintf(fp," stroke-width = '%d' />", stroke_width);
121 }
122
123 void romain_aaron_drawImage(struct Lines myLines, struct Circles myCircle,
124                             struct Stroke myStroke, struct Fill myFill,
125                             int steps, int rings)
126 {
127     int i;
128     for (i=0;i<steps;i++)
129     {
130         drawLine(myLines.x1[0], myLines.y1[0], myLines.x2[i],
131                 myLines.y2[i], myStroke.r, myStroke.g,
132                 myStroke.b, myStroke.opacity, myStroke.width);
133     }
134     for (i=0;i<rings;i++)
135     {
136         drawCircle(myCircle.x[0], myCircle.y[0], myCircle.r[i],
137                 myFill.r, myFill.g, myFill.b, myFill.opacity,
138                 myStroke.r, myStroke.g, myStroke.b, myStroke.opacity,
139                 myStroke.width);
140     }
141 }
143 void writeSVGHeader(char fileName[], int width, int height)
144 {
145     fp = fopen(fileName,"w");
146     fprintf(fp,"<?xml version='1.0' standalone='no'?>");
147     fprintf(fp,"\n<svg xmlns='http://www.w3.org/2000/svg' ");
148     fprintf(fp,"xmlns:xlink='http://www.w3.org/1999/xlink' ");
149     fprintf(fp,"version='1.1' width = '%d' height = '%d'>", width, height);
150 }
```

```
151
152 void writeSVGFooter()
153 {
154     fprintf(fp,"\n</svg>");
155     fclose(fp);
156 }
157


158 int main()
159 {
160     int i, j, k;
161     srand(time(NULL));
162     writeSVGHeader(fileName, imWidth, imHeight);
163     struct Lines myLines;
164     struct Circles myCircle;
165     struct Fill myFill;
166     struct Stroke myStroke;
167     for (j=0;j<n;j++)
168     {
169         setStroke(&myStroke);
170         setFill(&myFill);
171         float cx_in = xc;
172         float cy_in = yc;
173         float r_in = radius;
174         int steps = step;
175         int rings = ring;
176         romain_aaron_getData(&myLines, &myCircle,
177                              cx_in, cy_in, r_in, steps, rings);
178         romain_aaron_drawImage(myLines, myCircle,
179                                myStroke, myFill, steps, rings);
180     }
181     writeSVGFooter();
182     return 0;
183 }
```