Rakel María Brynjólfsdóttir rmb3@hi.is
Dæmahópur 9 skil: 30.10.15
Dæmatímakennari: Kristófer Montazeri

Tölvunarfræði 1
Heimadæmi 9

Dæmi 1

Leyndo(4,5) skilar 1024, og almennt skilar það a^b.

Dæmi 2

```
public class RecursivePower  {

     public static int exp(int base, int n) {
          if(n == 0) return 1;

          else return base * exp(base, n-1);
     }

     public static void main (String[] args) {
          int base = Integer.parseInt(args[0]);
          int n = Integer.parseInt(args[1]);
          System.out.println(exp(base, n));
     }
}
```

Dæmi 3

$f_n = f_{(n-1)} + f_{(n-2)} - f_{(n-5)}$

Með endurkvæmni:

```
public class Fibonacci4 {
    public static long fib(int n) {
        if (n == 0) return 0;
        if (n == 1) return 1;
        if (n == 2) return 1;
        if (n == 3) return 2;
        if (n == 4) return 3;
        if (n == 5) return 4;
        else return fib(n-1) + fib(n-2) - fib(n-5);
    }
```

Með fylki:

```
public class Fibonacci4 {

     public static long fibo(int n) {
          long[] f = new long[n+5];

          f[1] = 1;
          f[2] = 1;
          f[3] = 2;
          f[4] = 3;
          f[5] = 4;

          for (int i=6; i<=n; i++) {
          f[i] = f[i-1] + f[i-2] -f[i-5];
          }
          return f[n];
     }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        StdOut.println(fibo(N));
    }
}
```

Dæmi 4

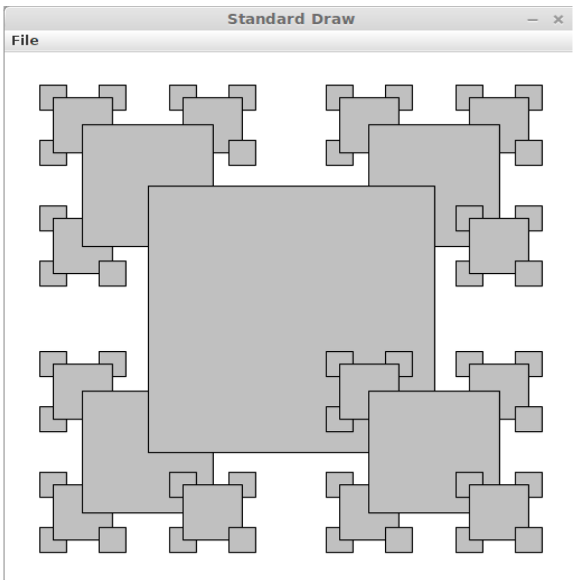
Nr 1:

```
  public static void draw(int n, double x, double y, double size) {
        if (n == 0) return;

        // 2.2 ratio looks good
        double ratio = 2.2;

        // recursively draw 4 smaller trees of order n-1

        draw(n-1, x - size/2, y - size/2, size/ratio);  // lower left
        draw(n-1, x - size/2, y + size/2, size/ratio);  // upper left
        draw(n-1, x + size/2, y + size/2, size/ratio);  // upper right
        drawSquare(x, y, size);
        draw(n-1, x + size/2, y - size/2, size/ratio);  // lower right
    }
```



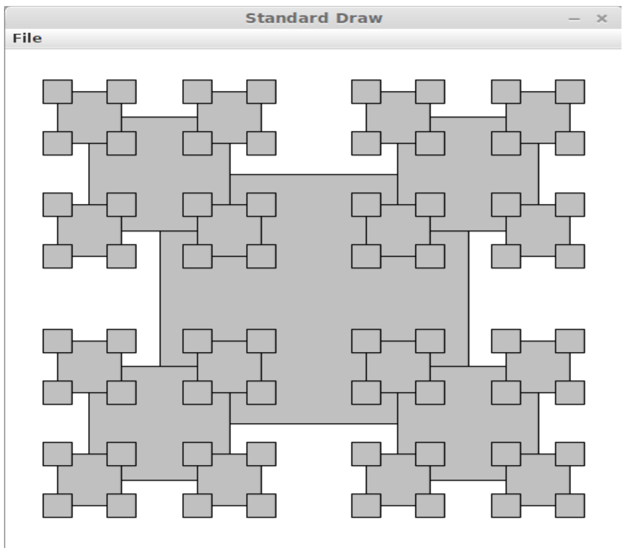
Nr 2:

```
public static void draw(int n, double x, double y, double size) {
     if (n == 0) return;

     // 2.2 ratio looks good
     double ratio = 2.2;

     // recursively draw 4 smaller trees of order n-1
     drawSquare(x, y, size);
     draw(n-1, x - size/2, y - size/2, size/ratio);  // lower left
     draw(n-1, x - size/2, y + size/2, size/ratio);  // upper left
     draw(n-1, x + size/2, y + size/2, size/ratio);  // upper right
     draw(n-1, x + size/2, y - size/2, size/ratio);  // lower right


 }
```



Nr 3:

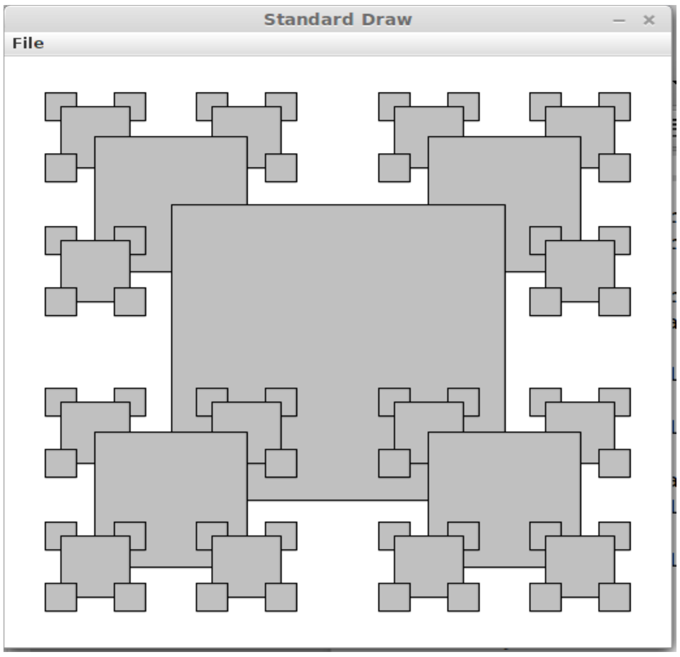
```
  public static void draw(int n, double x, double y, double size) {
       if (n == 0) return;


       // 2.2 ratio looks good
       double ratio = 2.2;

       // recursively draw 4 smaller trees of order n-1
       drawSquare(x, y, size);

       draw(n-1, x - size/2, y + size/2, size/ratio);  // upper left
       draw(n-1, x + size/2, y + size/2, size/ratio);  // upper right
       drawSquare(x, y, size);
       draw(n-1, x + size/2, y - size/2, size/ratio);  // lower right
       draw(n-1, x - size/2, y - size/2, size/ratio);  // lower left

   }
```



Dæmi 5

```
public class MySierpinski  {

     public static void drawTri(double[] x, double[] y) {
          StdDraw.filledPolygon(x,y); }

     public static void draw(int n,double x0, double x1, double x2,
                               double y0, double y1, double y2 ) {
          if (n==0) return;

          double [] x = new double[3];
          double [] y = new double[3];

          x[0] = (x0+x1)/2.0;
          x[1] = (x1+x2)/2.0;
          x[2] = (x2+x0)/2.0;
          y[0] = (y0+y1)/2.0;
          y[1] = (y1+y2)/2.0;
          y[2] = (y2+y0)/2.0;

          StdDraw.setPenColor(StdDraw.WHITE);
          drawTri(x,y);

          draw(n-1, x[0], x[1], x1, y[0], y[1], y1);
          draw(n-1, x[1], x[2], x2, y[1], y[2], y2);
          draw(n-1, x[2], x[0], x0, y[2], y[0], y0);
     }

     public static void main (String[] args) {
          int N =Integer.parseInt(args[0]);

          double[] x = {0.0, 0.5, 1.0};
          double[] y = {0.0, 0.866, 0.0};

          double x0 = 0.0, y0 = 0.0;
          double x1 = 0.5, y1 = 0.866;
          double x2 = 1.0, y2 = 0.0;

          StdDraw.setPenColor(StdDraw.BLACK);
          drawTri(x,y);
          draw(N, x0, x1, x2, y0, y1, y2);
     }
}
```