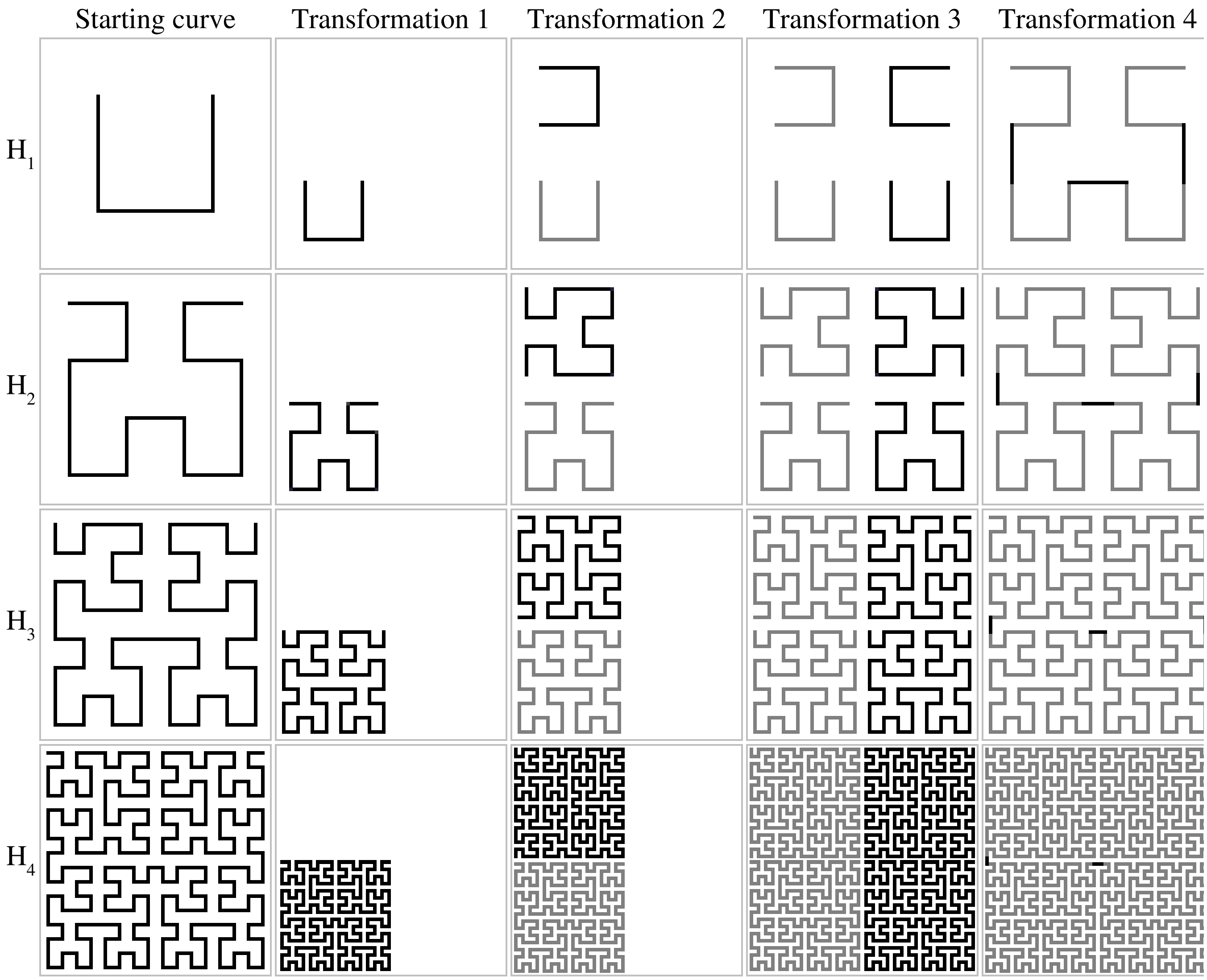
Problem B: Hilbert Curve Intersections

Source file: `hilbert.{c, cpp, java, pas}`
Input file: `hilbert.in`
Output file: `hilbert.out`

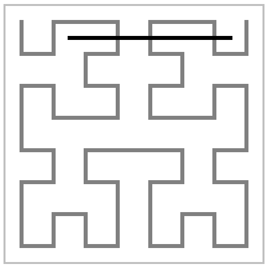


David Hilbert proved the existence of a very counter-intuitive curve that fills space. The construction of the Hilbert curve is based on a sequence of curves, H_1, H_2, H_3, H_4, ... composed of horizontal and vertical segments. Each curve lies in the unit square `[0, 1]` × `[0, 1]`. H_1 contains just three segments, connecting the points (¼, ¾) to (¼, ¼) to (¾, ¼) to (¾, ¾). H_n is defined recursively in terms of H_{n-1}, for $n = 2, 3, \ldots$ by four transformations:

1. Halve all coordinates in H_{n-1}.
2. Add a copy rotated 90 degrees counterclockwise about the point (`0`, ½).
3. Add the reflection across the line $x = ½$.
4. Let $m = ½^{n+1}$. Add segments connecting endpoints (½ - m, ½ - m) to (½ + m, ½ - m), (m, ½ - m) to (m, ½ + m), and (`1` - m, ½ - m) to (`1` - m, ½ + m).

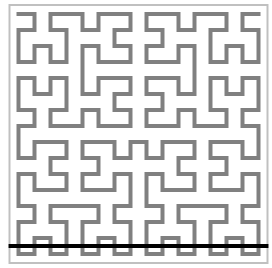
Your job is to count the number of intersections of horizontal line segments with these curves. For example, consider Figures 1 and 2, which illustrate the first two example input data sets below.

Figure 1



Segment from (2/8, 7/8) to (7/8, 7/8) crossing H_3 three times.

Figure 2



Segment from (0/16, 1/16) to (16/16, 1/16) crossing H_4 sixteen times.

The coordinates of vertices of H_n are odd multiples of $½^{n+1}$. The coordinates of horizontal segment endpoints will always be multiples of $½^n$. Hence the specified horizontal segment can only cross vertical segments in H_n.

Input consists of one to 100 data sets, one per line, followed by a final line containing only 0. Each data set consists of four integers separated by blanks in the form

$$n\ x_1\ x_2\ y$$

which represents H_n and the segment from $(x_1/2^n, y/2^n)$ to $(x_2/2^n, y/2^n)$, where $0 < n < 31$, $x_1 < x_2$, and each of x_1, x_2, and y lie in the range 0 to 2^n, inclusive.

The output is one integer per line for each data set: the number of intersections of H_n with the segment. **Caution**: A brute force solution that computes each intersection individually will not finish within the one minute time limit. As you can see below, there may be more than one billion intersections for any data set.

Example input:	Example output:
3 2 7 7 4 0 16 1 30 1 1073741823 1 0	3 16 1073741822

Last modified on October 26, 2002 at 7:25 PM.