```
/* Problem 5--1 or 4
   It turns out very few numbers shrink to 1.  They almost always shrink
   to 4. */

import java.io.*;
import java.util.*;

public class prob5 {

 private static Scanner in;
 private static PrintWriter out;
 private static int cs;

 public static void main (String[] args) throws Exception {

  cs = 1;
  in = new Scanner (new File ("prob5.in"));
  out = new PrintWriter ("prob5.out");
  while (true) {
   int num = in.nextInt();
   if (num==0) break;
   Process (num);
  }
  in.close();
  out.close ();
 }

 /* This method shrinks the number and prints the result. */
 public static void Process (int n) throws Exception {

  int[] s = Shrink (n);
  out.printf ("Case %d: %d shrinks to %d in %d step(s).\r\n\r\n",
              cs++,n,s[0],s[1]);
 }

 /* This just extracts the digits to compute a new digit, recursively
    until 1 or 4 is reached. */
 public static int[] Shrink (int n) throws Exception {

  int[] s = new int[2]; //Base case
  if (n==1 || n==4) {s[0]=n; s[1] = 0; return s;}
  int nn = 0;
  while (n>0) { //Make the new number
   nn += (n%10)*(n%10);
   n/=10;
  }
  s = Shrink (nn); //Recursively call with new number.
  s[1]++;
  return s;
 }
}
```