```
/* Problem 4--Twin Primes
   There is no need to test all numbers less than n to see if it's
   prime.  All you really have to do is test all PRIMES less than or
   equal to the SQUARE ROOT of n.  My approach wasn't that
   sophisticated; I just tested odd numbers up to the square root.
   You know, you could save known primes in a table, I suppose, for
   future input cases, but this wasn't necessary. */

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

public class prob4 {

 private static Scanner in;
 private static PrintWriter out;
 private static int cs, l, h;

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

  in = new Scanner (new File ("prob4.in"));
  out = new PrintWriter ("prob4.out");
  cs = 1;
  while (true) {
   l = in.nextInt(); h = in.nextInt();
   if (l==0 && h ==0) break;
   out.printf ("Case %d: There are %d pairs of twin primes between %d"+
               " and %d.\r\n\r\n",cs++,CountPrimes(),l,h);
  }
  in.close ();
  out.close ();
 }

 public static boolean Prime (int p) throws Exception {

  if (p <= 1) return false; /* 1 isn't prime */
  if (p==2) return true; /* 2 is prime */
  if (p%2==0) return false; /* Other even numbers aren't prime */
  for (int i=3; i*i <= p; i+=2) /* Test odds */
   if (p%i==0) return false;
  return true;
 }

 public static int CountPrimes () throws Exception {

  int low = l;
  int high = h;
  int ct = 0;
  if (low%2==0) low++; /* Make sure we only test odds */
  if (high%2==0) high--;
  for (int p = low; p <= high-2; p+=2)
   if (Prime(p) && Prime (p+2)) ct++;
  return ct;
 }
}

```