by Jeffreys Copeland and Haemer

PAUL STODDARD

"I have not kept my square;
but that to come
Shall all be done by the rule."
– William Shakespeare,
Antony and Cleopatra

"What mighty contests rise
from trivial things."
– Alexander Pope,
The Rape of the Locker

Going Through Our Lockers

Last month, we were inspired by an elementary school math class. This month, we'll take you to middle school or junior high, depending on your local school district. Leaving elementary school brings many changes. The one that our daughters seem to attach the most importance to is…lockers!

Here's a problem that a middle-school math teacher gave us over coffee:

Imagine all the students lined up in front of their lockers–Student One in front of Locker 1, Student Two in front of Locker 2 and so on. All the lockers are closed.

We start by having Student One open Locker 1, then open Locker 2, then Locker 3, Locker 4 and so on down the line. When he opens the last locker, he returns to his own locker and stands in front of it.

Next, Student Two closes Locker 2, then Locker 4, Locker 6 and every other locker down to the end of the line, and returns to her place.

Now it is Student Three's turn. Student Three closes Locker 3, opens Locker 6 (which, remember, was opened by Student One and then closed by Student Two), closes Locker 9 and so on, fiddling with the door of every third locker.

If we continue this process through to the last student, which lockers are open, which are closed, and why?

Before we go any further, what's your answer?

What?

Our answer was, "Um…The prime-numbered lockers are open?" (Two great minds in one great rut.) Wrong.

If this was your answer too, don't feel bad. We've given several of our friends this problem, and many of them gave us this same wrong answer. Another frequent off-the-cuff wrong guess is the Fibonacci series. Michael, who works behind the counter in our neighborhood coffeehouse, guessed factorials. That's wrong too.

Instead of guessing, perhaps you tried using paper, a pencil and an eraser. We, of course, wrote a short program (see Listing 1 on Page 44).

Lines 1 through 3 are boilerplate, with a shebang line containing the `-w` flag, and the compulsive, nit-picking strict module that we tend to use because we're poor typists.

Lines 4 and 5 use the first argument as the total number of lockers and exit with a bitter complaint if it isn't present.

Normally, Perl arrays begin at index 0, as in C, but we want our locker numbers to begin at 1, so line 6 says that in this program, all our array indices are going to begin at 1, as in FORTRAN.

Line 7 sets up an array of lockers and starts with them all closed (we'll use `$locker[5] = 0;` to mean Locker 5 is closed and `$locker[5] = 1;` to mean Locker 5 is open).

Lines 8 through 12 perform the openings and closings and lines 13 through 15 print out the numbers of the open lockers.

Listing 1

```
#!/usr/local/bin/perl -w
# $ID: lockers,v 1.1 1998/12/22 01:26:41 jsh Exp $

use strict;

my $usage = "usage: $0 num_lockers";
my $n = shift || die $usage;

$[ = 1;
my @lockers = (0)x$n;

for (my $i=1; $i <= $n; $i++) {
 for (my $j=$i; $j <= $n; $j += $i) {
  $lockers[$j] = $lockers[$j] ? 0 : 1;
 }
}

for (my $i=1; $i <= $n; $i++) {
 print "$i\n" if ($lockers[$i]);
}
```

Listing 2

```
1   #!/usr/local/bin/perl -w
2   # $ID: divisors,v 1.1 1998/12/22 01:26:41 jsh Exp $

3   use strict;

4   my $usage = "usage: $0 num_lockers";
5   my $n = shift || die $usage;

6   $[ = 1;
7   my @divisors;

8   for (my $i=1; $i <= $n; $i++) {
9    for (my $j=$i; $j <= $n; $j += $i) {
10    push @$divisors[$j], $i;
11   }
12  }

13  for (my $i=1; $i <= $n; $i++) {
14   my $t = scalar @$divisors[$i]};
15   printf "%3d [%2d] : ", $i, $t;
16   print "@$divisors[$i]}\n";
17  }
```

And the output? (Drumroll, please.)

```
$ lockers 200
1 4 9 16 25 36 49 64
81 100 121 144 169 196
```

Yup. Squares.

(Like you, middle-schooler Gillian Haemer said “squares,” but that requires the sophistication to know what a square is. Luckily, there’s more than one way to look at patterns. Her little sister, Zoe, said “3, 5, 7, 9… I see the pattern.” Do you see what she saw?)

But that’s only the first half of the answer.

Why?

Last month, when we were talking about elementary-school math, we remarked that computers sometimes raise more questions than they answer. Here’s a case where a short program gave us a quick, easy-to-interpret pattern as the answer to an interesting puzzle. This forced us head-on into the question, “Why?”

So, let’s think. Well, if you’re locker *N*, who messes with your door? Easy. The student corresponding to each of your divisors, including Student One and Student *N*. The locker door ends up being open if, and only if, it has an odd number of divisors.

So squares, and nothing else, have an odd number of divisors? Precisely.

Okay, why’s that? Let’s take a look at the divisors of each number–another short program (see Listing 2).

This program is just a variant of our program in Listing 1. This time, instead of opening or closing locker doors, line 14 pushes each divisor of `$j` onto the end of the array `@$divisors[$j]`, and lines 15 and 16 print out the lists of divisors, along with the number of elements in each list.

```
$ divisors 30
  1 [ 1] : 1
  2 [ 2] : 1 2
  3 [ 2] : 1 3
  4 [ 3] : 1 2 4
  5 [ 2] : 1 5
  6 [ 4] : 1 2 3 6
  7 [ 2] : 1 7
  8 [ 4] : 1 2 4 8
  9 [ 3] : 1 3 9
 10 [ 4] : 1 2 5 10
 11 [ 2] : 1 11
 12 [ 6] : 1 2 3 4 6 12
 13 [ 2] : 1 13
 14 [ 4] : 1 2 7 14
 15 [ 4] : 1 3 5 15
 16 [ 5] : 1 2 4 8 16
 17 [ 2] : 1 17
 18 [ 6] : 1 2 3 6 9 18
 19 [ 2] : 1 19
 20 [ 6] : 1 2 4 5 10 20
 21 [ 4] : 1 3 7 21
 22 [ 4] : 1 2 11 22
 23 [ 2] : 1 23
 24 [ 8] : 1 2 3 4 6 8 12 24
 25 [ 3] : 1 5 25
 26 [ 4] : 1 2 13 26
 27 [ 4] : 1 3 9 27
 28 [ 6] : 1 2 4 7 14 28
 29 [ 2] : 1 29
 30 [ 8] : 1 2 3 5 6 10 15 30
```

Aha. Well, the first thing we see is something we already knew: only prime numbers have exactly two divisors (one and themselves). What about some of the other compound numbers? The powers are easy: 2 has two divisors, 4 has three, 8 has four, 16 has five and so on. It's easy to see that for any prime, p^n has $n + 1$ divisors: p^0, p^1, … p^n.

But how about the other compound numbers, like 72? $72= 2^3 \times 3^2$.

First, we arrange all the divisors in a table, like this:

x	1	3	9
1	1	3	9
2	2	6	18
4	4	12	36
8	8	24	72

Across the top, we have the possible powers of three and down the side, the possible powers of two. Taken together, the table entries constitute all (3+1) x (2+1) = 12 possible combinations.

Similarly, though harder to draw, it should be pretty clear that the divisors of $900 = 2^2 \times 3^2 \times 5^2$ can be laid out in a (2+1) x (2+1) x (2+1) = 3 x 3 x 3 cube. (Okay, okay, "a three-dimensional rectangular parallelepiped.")

Much harder to draw would be the four-dimensional grid of divisors of $4902963250500 = 2^2 \times 3^5 \times 5^3 \times 7^9$, but it's not hard to see how many elements would be in it: 3 x 6 x 4 x 10 = 720.

So which numbers will have a multidimensional array of divisors with an odd number of elements? Only those with an odd number of elements in every dimension; an even number in any dimension would make the product of the dimensions even.

And how many elements are there in each dimension? One more than the power of the prime that dimension represents. (Thus, in our first example above, $72 = 2^3 \times 3^2$, we have four elements in the dimension representing the prime factor 2, and three elements in the dimension representing the prime factor 3.)

But for each axis to be of odd length, each prime factor must be raised to an even power. And if each prime factor is raised to an even power, then the number is a square.

For example, $144 = 12^2 = (2^2 \times 3^1)^2 = 2^4 \times 3^2$ will have (4+1) x (2+1) = 15 factors, and Locker 144 will be open.

Not bad, eh?

Not Again!

Okay, here's another one.

Same students, same lockers, opposite rules. This time, Student One doesn't open his locker or anyone else's. Student Two doesn't open Locker 2, but opens Locker 3, skips Locker 4, opens Locker 5 and so on.

Student Three doesn't open Locker 3, does open Locker 4, closes Locker 5, ignores Locker 6 and goes on changing the state of every locker that is not divisible by three. Now which lockers are open?

("Um…Primes?" No again.)

Listing 3

```
#!/usr/local/bin/perl -w
# $ID: lockers2,v 1.1 1998/12/22 15:29:44 jsh Exp $

use strict;

my $usage = "usage: $0 num_lockers";
my $n = shift || die $usage;

$[ = 1;
my @lockers = (0)x$n;

for (my $i=1; $i <= $n; $i++) {
 for (my $j=$i; $j <= $n; $j++) {
  $lockers[$j] = ($lockers[$j] ? 0 : 1) if $j%$i;
 }
}

for (my $i=1; $i <= $n; $i++)  {
 print "$i\n" unless $lockers[$i];
}
```

Listing 3 shows the code. And the answer:

```
$ lockers2 100
1 2 6 8 9 10
12 14 18 20 22
24 25 26 28 30
32 34 38 40 42
44 46 48 49 50
52 54 56 58 60
62 66 68 70 72
74 76 78 80 81
82 84 86 88 90
92 94 96 98
```

Aha! It's the even numbers. Well, almost.

What's the pattern? And why?

We see the answer to the first question, but not a good proof for the second one. Maybe you have to be a middle-schooler to come up with one; if you have a middle-schooler who does, please pass the proof along and we'll print it. Meanwhile, we'll ask Gillian Haemer and Allie Copeland.

Until we hear from you or them, happy trails.

Jeffrey Copeland *(*`copeland@alumni.caltech.edu`*) lives in Boulder, CO, and works at Softway Systems Inc. on UNIX internationalization. He spends his spare time rearing children, raising cats and being a thorn in the side of his local school board.*

Jeffrey S. Haemer *(*`jsh@usenix.org`*) works at QMS Inc. in Boulder, CO, building laser printer firmware. Before he worked for QMS, he operated his own consulting firm and did a lot of other things, like everyone else in the software industry.*

Note: The software from this and past Work columns is available at `http://alumni.caltech.edu/~copeland/work`.