by Jeffreys Copeland and Haemer

Plus ça change, plus c'est la même chose.
– The French

Vive la différence.
– Also the French

Differences Among Women

They say nothing ever changes. Well, we've had a first. A couple of months ago, we wrote a column entitled "A Short History of Reading" (November 1998, Page 58). It had some novel features.

• It was the only article we've written that didn't include any code. Copeland remarked that he got a few paragraphs into the first draft and discovered he was channeling James Burke.

• It generated quicker reader responses than any of our past articles–one even before either of us had received our copy of the November issue.

• It prompted a response from Corina Mullen at Atmel Corp., San Jose, CA.

The 75 columns we've written for *RS/Magazine* and *SunExpert* have generated a lot of email. Some folks express appreciation, some pick at our errors, some do both. We've had responses from all over the world and from all sorts of folk. But from a woman? Never.

(In fact, the first *two* responses to the column were from women.)

Think about this. OK, we've never gotten a reader response from Antarctica either but, even without consulting a world almanac, we're willing to chalk this up to the fact that there are very few people in Antarctica to respond in the first place. (Hey, Antarctica. Anyone out there read *SunExpert*?)

This is food for thought for us–the fathers of four daughters. More about this in a moment. First, we want to talk a little about a math problem that Zoe Haemer brought home from third grade.

Differences, Differences, Differences

Here 'tis. Start with any four integers. We'll use (0,1,4,11) to create a simple example. Next, take the absolute value of the differences of adjacent numbers (consider it a circular list): (1,3,7,11), in our example.

Repeat this process until all the numbers are the same:
(0,1,4,11) -> (1,3,7,11) -> (2,4,4,10) -> (2,0,6,8) -> (2,6,2,6) -> (4,4,4,4)

That took five steps. Zoe's challenge was to find one that took six. The book her teacher took the problem from said that no one had ever found an example that required more than six.

We think this is a great problem for a third-grader for several reasons, the most important of which is that it requires a lot of subtracting.

It is also a great problem for parents who program because it takes very little programming to find four-tuples that generate more than six steps. Our personal favorite is (0,1,4,9); not only are they the first four square numbers, they're all single digits:
(0,1,4,9) -> (1,3,5,9) -> (2,2,4,8) -> (0,2,4,6) -> (2,2,2,6) -> (0,0,4,4) -> (0,4,0,4) -> (4,4,4,4)

Seven steps.

It really impresses your third-grader when you show her you can do something that her math book says no one's ever been able to do before.

But discovering that something is unusual when it should be commonplace piques the curiosity. You understand, it's like getting your first reader response from a woman. Haemer talked to his daughter's math teacher to investigate further.

The teacher's first response? "Several fathers got interested in that problem and sent in solutions with seven steps." We note that's not "several parents."

We'll return to this math problem in a moment. Patience.

Where Are the Missing Women?

So where are the women? We're not sure, but let's knock down some commonly advanced straw men. (Oh, stop it. You know good-and-well what we mean.)

• Women don't program. Two of *SunExpert*'s columnists, S. Lee Henry and Æleen Frisch, are women, and both Copeland's wife and Haemer's ex-wife are professional programmers. One wonders how Rear Admiral Grace Murray Hopper would have used the colorful vocabulary of the Navy to describe people who make such a claim.

At the same time, we will note that there is something odd about women and programming. A few years back, when Haemer taught a series of undergraduate computer science courses at the University of Colorado, he noticed something puzzling. Each course had a similar distribution of grades across the sexes, but each successive course in the curriculum had a noticeably smaller percentage of women. The sex ratio didn't change because of a difference in ability, but by the time it went from the first course to the last, classes went from being nearly half women to being nearly all men.

Women didn't drop out of computing because they couldn't do the work, but they did drop out of it.

• Women don't do math. When Riley Haemer was nine years old, she explained to her father that she was going to be good in math because "girls are good in math." This wasn't some parroted piece of political correctness, just keen observation: Riley's mother got her Ph.D. finding new methods to solve systems of partial differential equations on big, fast computers; Riley's Aunt Barb has a Ph.D. in mathematics from MIT; Riley's Grandma Martile got her B.S. in mathematics and chemistry in 1936, and was in the first group of women to go to U.S. Army Engineering school at Fort Belvoir, VA, during World War II.

In contrast, both the Jeffs' fathers were artists. Haemer's father became a painter when the Syracuse Forestry school refused to admit him because of his D in high-school geometry. Copeland's father became a silversmith when he realized how uninteresting he found his father's work in chemistry.

This generalization from her family was reinforced by Riley's grade-school experience. Early on, girls develop noticeably more quickly. Her gifted-and-talented math class had only one boy. (Oh, and who taught the class? Nearly all grade-school teachers are women.)

If math and computing were female-dominated, we'd hear sociological Just-So stories attributing this to early educational experiences.

• Women don't write us or send solutions in to school because they don't want to call attention to themselves. Maybe women don't send email because they fear cyberstalkers. Perhaps visions of Jeffs chasing them down long electronic corridors deter them from writing us. After all, there are two of us, making it that much easier to corner them. (And just to allay any nervousness, we asked for Corina's permission to use her name above. She made us promise to spell it correctly.)

And they don't send novel solutions to school with their kids because they're afraid that doing so will get undesired attention from Mrs. Cerny, the math teacher.

• Women just aren't as verbal as men. Oops. We got carried away there for a minute.

Get the picture? We have our own pet theories, probably no better, but at least they're different. Here are a couple:

• Women have something better to do.

Q: What's the difference between a dead skunk in the middle of the road and a dead programmer in the middle of the road?

A: The skunk had a life.

Zoe's mother was the first woman to earn a Ph.D. in chemical engineering from the University of Colorado. At one point, undergraduate representatives from the Society of Women Engineers invited her to speak to them. The question they most wanted to ask her? "Isn't it hard to get dates?"

This question may put the cart before the horse. For many people, the years from high school through the early 20s are full of important social activity. We spent a large fraction of this time in the basements of computing centers and engineering buildings. Would we have if we could have gotten dates instead? Be honest.

Q: What's the difference between a dead skunk in the middle of the road and a dead programmer in the middle of the road?

A: The skunk had a life.

Female engineers, in contrast, don't seem to have major difficulties getting a date when they want one. If all else fails, they can date male engineers. Maybe all those extra hours guys spend debugging instead of dating contribute to the odd sex ratio.

• Women are more attuned to social pressures than men. Everyone knows that scientists and engineers aren't cool. Quickly, now, name a cartoon character besides Dilbert that makes fun of the social skills of an entire field. Rex Morgan, M.D.? Zippy the Pinhead? Prince Valiant? If that's not convincing, ask the next five women you meet whether they'd rather sleep with Bill Gates or Michael Jordan.

One friend, when she was in graduate school, quickly tired of the reactions she got from young men she met at taverns and on the dance floor.

"A Ph.D. student in math? Oh. That must be interesting. Well. Will you look at the time? I really have to go."

Inspiration finally hit. She just lied.

"A sex therapist? Really? Um. Well. See, I have this friend who has a problem and I was wondering what you…."

Why wouldn't this drive the same number of men as women out of computing? Perhaps men are just more willing to be socially unacceptable. We know more men who are loners than women. Just ask Ted Kaczynski–but hope he doesn't answer by mail.

We have complete confidence that you have your own pet theories that are just as silly as ours and that you will mail them to us. If you're male.

Subtracting Performance

Enough editorializing and idle speculation, let's talk code. We wrote a program to do an exhaustive search for solutions to Zoe's problem, and find beginning numbers that generated a cycle of six or more steps. We actually ended up writing several versions of programs to generate solutions.

The first one was in Perl. Even though it's an arithmetic problem, the arithmetic isn't difficult. Add to this the fact that it uses lots of lists and we like to program in Perl. Our Perl solution turned out to be ugly, slow as the dickens and made our hard disk sound like our computer was home to a click beetle convention.

OK, maybe C is still good for something. Besides, we didn't have a FORTRAN compiler. (And anyway, as Copeland's wife has been heard to observe, "A good FORTRAN programmer can write FORTRAN in any language.")

Listing 1 shows our second cut.

We weren't content to leave well enough alone (is any programmer?), so we decided to look for ways to speed it up.

Our first optimization came from realizing that the problem can be solved recursively, like factorials. We can figure out the number of steps for (1,1,1,3) by figuring out that it takes one more step than (0,0,2,2).

Not only can we write a recursive version, but we can use the trick used in factorial functions of caching away each answer to prevent having to calculate it again.

A second optimization cuts our work even further. All circular permutations of a list take the same number of steps. So does the reverse of each of these. As soon as we calculate and store the number of steps for (1,2,3,4), we can store the number of steps for seven others: (2,3,4,1), (3,4,1,2), (4,1,2,3), (4,3,2,1), (3,2,1,4), (2,1,4,3) and (1,4,3,2).

Clever, eh? (Reader quiz: There is a second set of eight that can also be stored. What is it?)

Listing 2 shows our clever, optimized version.

But how clever is it? Not very, as it turns out. When the integers in the lists are small, either program is fast enough. But by the time we permit large numbers in the four-tuples–say, up to 100–the original, brute-force method is actually faster! Why? The chattering of our disk reminds us that with integers this big, the array we're allocating to cache the steps for each four-tuple is 4 x 100 x 100 x 100 x 100 bytes. We're pretty cavalier about space on today's machines–an old programmer is

Listing 1

```
/* $ID: steps.c,v 1.1 1998/12/01 04:54:06 jsh Exp jsh $ */

#include <stdio.h>
#include <stdlib.h>

int nmax = 10;
int lim = 6;

int
steps(int a, int b, int c, int d)  {
 int n = 0;
 int a0;

 while(1)   {
  if ((a == b) && (a == c) && (a == d))
   return n;
  a0 = a;
     a = abs(b-a); b = abs(c-b); c = abs(d-c); d = abs(a0-d);
  n++;
 }
}

int
main(int argc, char *argv[]) {

 int i, j, k, l, n;

 int lim = 6;  /* don't print anything with under lim steps */
 int nmax = 9; /* maximum integer in a 4-tuple */

 if (argc > 3)   {
  fprintf(stderr, "usage: %s [cutoff] [maxint]\n", argv[0]);
  exit(1);
  }
  if (argc == 3)
   nmax = atoi(argv[2]);
  if (argc > 1)
   lim = atoi(argv[1]);

  for (i = 0; i<=nmax; i++)
   for (j = 0; j<=nmax; j++)
    for (k = 0; k<=nmax; k++)
     for (l = 0; l<=nmax; l++) {
      if ((n = steps(i, j, k, l)) >= lim)
       printf("%d %d %d %d : %d \n",
        i, j, k, l, n);
     }
  exit(0);
}
```

Listing 2

```
/* $ID: optimized_steps.c,v 1.1 1998/12/01 04:54:34 jsh Exp jsh $ */

#include <stdio.h>
#include <stdlib.h>

#define UNDEF -1
int ****S;

/* create a 4d cube of width "len" */
int ****
make_4d_cube(int len)
{
 int i, j, k, l;
 int *s;
 int **r;
 int ***q;
 int ****p;

 p = malloc(len * sizeof(int ***));
 for (i=0; i<len; i++)   {
  q = malloc(len * sizeof(int **));
  p[i] = q;
  for (j=0; j<len; j++)   {
   r = malloc(len * sizeof(int *));
   q[j] = r;
   for (k=0; k<len; k++)    {
    s = malloc(len * sizeof(int));
    r[k] = s;
    for (l=0; l<len; l++)    {
     s[l] = UNDEF; /* initialize all entries to UNDEF */
    }
   }
  }
 }
 return p;
}

int
steps(int a, int b, int c, int d)
{
 int x = S[a][b][c][d];
 if (x != UNDEF) return x;

 /* all circular permutations of this 4-tuple and their reverses
  have the same number of steps */
 x = S[a][b][c][d] = S[b][c][d][a] = S[c][d][a][b] = S[d][a][b][c] =
     S[d][c][b][a] = S[c][b][a][d] = S[b][a][d][c] = S[a][d][c][b] =
         steps(abs(b-a), abs(c-b), abs(d-c), abs(a-d)) + 1;
 return x
}

int
main(int argc, char *argv[])
{
 int i, j, k, l, n;
 int lim = 6;  /* don't print anything with under lim steps */
 int nmax = 9; /* maximum integer in a 4-tuple */
```

someone who remembers when a Kilobyte wasn't just an archaic name for a millimeg (and why Kilobytes were used to measure something called "core")–but a 400-MB array is still big enough to cause paging.

Minus the Answers

We can use either version of this program to find four-tuples that give us large numbers of steps. (0,1,4,9) is the first four-tuple with seven steps. To get the first four-tuple with eight steps, you need to allow integers up to 11: (0,2,5,11). You want nine steps? The first four-tuple with nine steps is (0,2,6,13).

Here's a table of the lowest integers that you need to permit in four-tuples to get specific numbers of steps:

Steps	Integers
1	1
2	1
3	1
4	3
5	3
6	4
7	9
8	11
9	13
10	31
11	37
12	44
13	105
14	125
15	149

Doesn't it seem odd that it's easy to find seven-, eight- and nine-step examples with low integers, yet you have to go all the way up to four-tuples containing 31 to find a 10-step example? And look at the jump it takes to go from 12 to 13. And why do these seem to cluster in groups of three? We don't know.

Can we just keep going? Are there four-tuples with arbitrarily large numbers of steps? If we go high enough, can we find four-tuples that don't stabilize at all and have an infinite number of steps? We don't know.

Computers, with their brute-force solutions, often raise as many questions as they answer. This started out as a math problem. Can we answer our questions by deriving a mathe-

```
 if (argc > 3)   {
  fprintf(stderr, "usage: %s [cutoff] [maxint]\n", argv[0]);
  exit(1);
 }
 if (argc == 3)
  nmax = atoi(argv[2]);
 if (argc > 1)
  lim = atoi(argv[1]);

 S = make_4d_cube(nmax+1);

 /* entries like (3,3,3,3) have 0 steps */
 for (i = 0; i<=nmax; i++)
  S[i][i][i][i] = 0;

 /* now figure out the number of steps for each 4-tuple */
 for (i = 0; i<=nmax; i++)
  for (j = 0; j<=nmax; j++)
   for (k = 0; k<=nmax; k++)
    for (l = 0; l<=nmax; l++) {
     if ((n = steps(i, j, k, l)) >= lim)
      printf("%d %d %d %d : %d \n",
       i, j, k, l, n);
    }
 exit(0);
}
```

matical foundation for all this–say, from the calculus of finite differences? We don't know.

In fact, why don't we say anything about four-tuples that take more than 15 steps? Because we haven't been able to generate one yet. No, wait. That is, "no one has ever found an example that takes more than 15 steps."

We'll report the sex ratio of reader responses in a future column. Until then, 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`.