



# A Ternary State of Affairs

Robert T. Kurosaka

## *The base-3 system offers an automatic solution to a classic puzzle*

This month we'll investigate the mathematics of balance scales. Our study begins with the natural language of computers, binary arithmetic, but soon moves on to the more exotic ternary or base-3 system. Even there, we'll find that computers come in handy. Consider the balance scale and measuring weights shown in figure 1. We intend to weigh some commodity (say, coffee) to the nearest ounce. The coffee goes in the right pan, and we add sufficient weights to the left pan until the scales balance, giving us the weight of the coffee.

What would be the minimum set of measuring weights needed to weigh at least 30 ounces of coffee? Before proceeding, you are urged to try and solve this warm-up question.

Thinking in familiar decimal terms, our first guess might be a set of eight weights: 10, 10, 5, 1, 1, 1, and 1 oz., respectively. An obvious refinement gives six weights: 10, 10, 5, 2, 2, and 1. However, further experimentation leads to the ideal solution of five weights: 16, 8, 4, 2, and 1, with which we can actually measure any amount up to 31 oz.

### First There Were Two

If you're thinking that the last sequence is composed of powers of 2, and thus suggests that the problem is binary in nature, you're right. To see why, look at the binary representation of our 31-oz. maximum weight: 11111<sub>2</sub>, which represents the number  $1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ . Each of the binary digits in the number corresponds to one of our weights. Smaller amounts are weighed by removing one or more of the binary counterweights.

We now have an automatic method for determining which weights are needed to offset a given amount: First convert the amount into binary notation, and then read off the digits. For each 1 digit, we take the correspondingly marked weight. For instance,  $28 = 11100_2$ , meaning that we use only the 16-, 8-, and 4-oz. counterweights

to offset a 28-oz. portion of coffee.

Our binary analysis also gives us the useful information that with  $n$  weights valued at 1, 2, 4, . . . ,  $2^{n-1}$ , we can counterbalance any amount up to  $2^n - 1$ .

Now we'll alter the weighing method to allow weights to be placed in either pan. For instance, to weigh out 3 oz. of coffee, we place the 4-oz. counterweight in the left pan and the 1-oz. counterweight in the right pan, plus enough coffee to balance the scale. The mathematical expression of this is  $4 = 1 + c$ , with  $c$  being the amount of coffee.

What's the minimum set of weights using this "bilateral" weighing method? Experiment with this one and come up with your own guess. (Hint: We can get by with fewer weights than with the "unilateral" system.)

### And Then There Were Three

Let's apply a little computer logic to the question. In the unilateral system, a weight could have two possible "states": on the scale or off the scale; that's why our binary model works so well. But in the new system, a weight can have three possible states: in the right pan, in the left pan, or off the scale. This leads us to try a ternary model for the counterweight values: 1, 3, 9, and so forth. By trial and error, we find that using just the weights 1, 3, 9, and 27, we can counterbalance any weight up to 40.

Can we apply the automatic method again for determining which weights will be needed to counterbalance a given amount? Let's try to weigh out 22 oz. of coffee. First convert 22 to ternary notation:  $22 = 2 \times 3^2 + 1 \times 3^1 + 1 \times 3^0 = 211_3$ . So we place two 9-oz. weights, one 3-oz. weight, and one 1-oz. weight in the left pan, and we place sufficient coffee in

the right pan to balance the scale. The corresponding equation is  $9+9+3+1=c$ .

But there's a problem—we don't have two 9-oz. weights. Furthermore, we're supposed to be placing some of the weights

in the right pan; that's the bilateral method.

We don't give up, though. Instead, we mentally add another 9-oz. weight to each pan (even though we really don't have any more 9-oz. weights). The scales still balance, and the equation is  $9+9+9+3+1=c+9$ . But the left side can be rewritten as  $27+3+1$ , and it suggests a real solution to the problem: On the left scale, place 27-, 3-, and 1-oz. weights; on the right scale, place a 9-oz. weight and the coffee to be weighed.

Fortunately, it is not necessary to perform this mental juggling act every time with the bilateral method. Ternary notation gives us another automatic process for figuring out which weights to use.

First convert the number (the amount to be weighed out) to ternary. For example,  $22 = 211_3$ .

Examine the digits of the ternary number from right to left. Each time we encounter a 2, change it to a -1 and add 1 to the next digit on the left. Continue moving to the left until no more 2s remain. In our example, we'll use delimiters to separate each of the individual digits:  $211_3 = (1) (-1) (1) (1)_3$ .

The resulting modified ternary "number" is a sequence of 1s, 0s, and -1s. For each 1, place the corresponding weight on the left pan. For each -1, place the corresponding weight on the right pan. Now add enough coffee to the right pan until the scale is balanced.

For practice, apply this automatic

*continued*

*Robert T. Kurosaka teaches mathematics in the Massachusetts State College system. He can be reached c/o BYTE, One Phoenix Mill Lane, Peterborough, NH 03458.*

MAKE YOUR PC  
SEEM LIKE AN AT!

MAKE YOUR AT  
SEEM LIKE A  
DREAM MACHINE!

**ANSI-<sup>tm</sup>**  
**CONSOLE**

The Integrated Console Utility™  
**FAST, POWERFUL**  
**ANSI.SYS REPLACEMENT**

For the IBM-PC, AT, and clones

New Version 2.00 is MUCH FASTER!  
Now blink free scrolling on CGA!

Now uses EMS/EEMS for Scroll Recall  
New Menu Program for Changing Options

GET A BOX FULL OF UTILITIES!  
MAKE LIFE EASIER FOR ONLY \$75!

- Speed up your screenwriting 2-6x
- Extend your ANSI.SYS to full VT100
- Add many more escape sequences
- Scroll lines back onto screen
- Save scrolled lines into a file
- Add zip to your cursor keys
- Free your eyes from scroll blinking
- Easy installation
- Get a 43 line screen w/EGA
- Get a 50 line screen w/CGA
- No more annoying typeahead beep
- Prevent screen phosphor burnin
- Control many programs' use of color
- Generate breakpts from keyboard
- Shorten that annoying bell
- Over 50 other useful options

"The psychological difference is  
astonishing"

—Lotus June 85 pg 8.

"So many handy functions rolled into  
one unobtrusive package"

—PC-World Feb 86 pg 282.

"The support provided by the  
publishers is extraordinary"

—Capital PC Monitor May 86 pg 25.

"... the best choice for improving your  
console..."

—Capital PC Monitor June 86 pg 26.

460p Manual (w/slip case) & disks \$75.

Satisfaction Guaranteed!  
Order Yours Today!

HERSEY MICRO CONSULTING  
Box 8276, Ann Arbor, MI 48107  
(313) 994-3259 VISA/MC/Amex

DEALER INQUIRIES INVITED



method to the number 46. (Hint: You should end up using a bilateral combination of 81, 27, 9, and 1.)

In general, given  $n$  weights valued at 1, 3, 9, ...,  $3^{n-1}$ , the bilateral weighing method will handle any weight up to  $(3^n-1)/2$ .

The program in listing 1 incorporates this ternary arithmetic to "weigh" any amount up to  $(3^{12}-1)/2$ .

With this warm-up completed, we're ready to take on a classic puzzle using a computer-aided approach.

### The Counterfeit Coin

We have a set of apparently identical coins containing one counterfeit. The counterfeit is off-weight (either light or heavy). We are to identify the bad coin in

*continued*

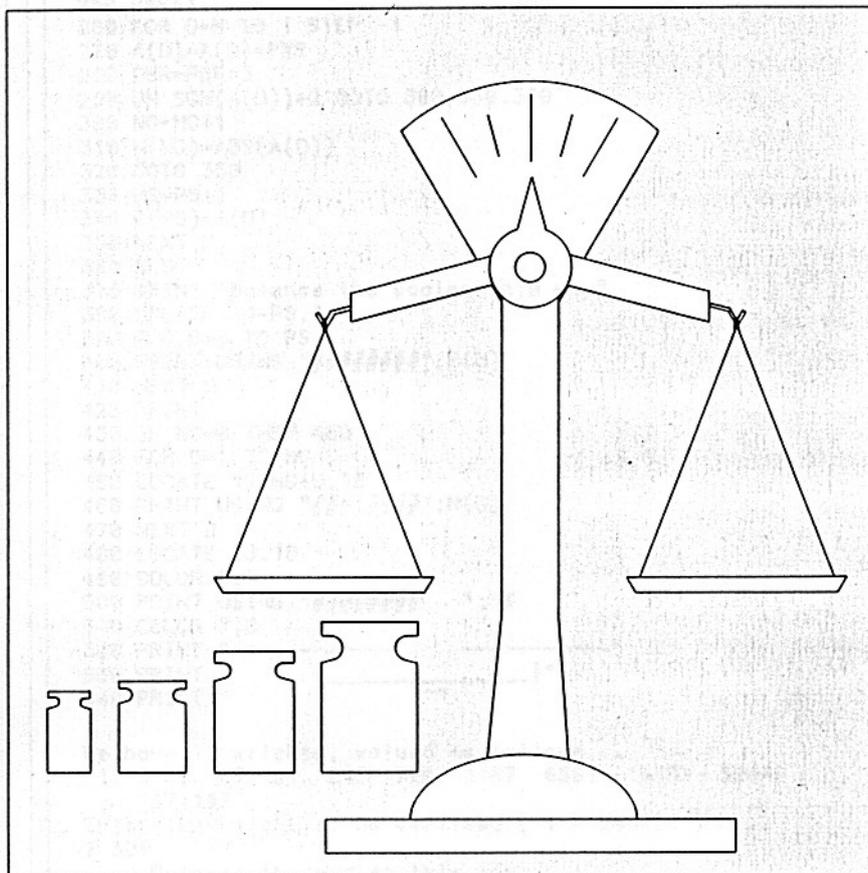


Figure 1: A set of balance scales with standard counterweights.

Listing 1a: BASIC program to verify a weight using ternary-power counterweights. A sample run is also shown.

```

10 DIM A(12),P(12),N(12)
20 N=12
30 CLS
40 PRINT "We have 12 weights, valued as follows..."
50 FOR W=0 TO N-1
60 PRINT 3^W;
70 NEXT W
80 LARGEST=INT((3^N-1)/2)
90 PRINT "Enter the weight to be verified ( 1 -"; LARGEST;
  " )"
100 INPUT W: W=INT(W)
110 IF W<1 OR W>LARGEST THEN 90
120 SW=W
130 FOR D=N TO 1 STEP -1
140 Q=INT(W/3)

```

*continued*

# LOOKING FOR "NET" RESULTS?

**EARTH COMPUTERS**  
 has the  
 solution  
 to your  
 Networking  
 problems.

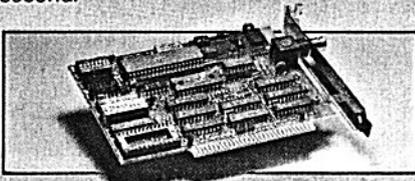


## EARTHNET-PC.™

EARTHNET-PC is the most flexible networking card on the market. It has been designed for high performance and maximum functionality.

EARTHNET-PC is fully compatible with SMC networking cards and runs popular networking software such as NOVELL's NETWARE, ViaNet, and TurboDOS, all of which support the new LAN Standard and DOS 3.1 record locking.

EARTHNET-PC's 5-1/2 inch card fits in any short slot of an IBM-PC/XT or compatible system and uses advanced Token-Passing technology. Data transfers are made at 2.5 Megabytes per second.



YOU CAN STOP SEARCHING...EARTHNET-PC IS THE SOLUTION TO YOUR NETWORKING PROBLEMS! Order your EARTH-NET-PC today! Call EARTH COMPUTERS, the company that's setting the standard for LAN standards.

**ATTENTION DEALERS!** If you've been searching for ways to increase your Networking profits, call EARTH COMPUTERS and find out about our attractive, profit-generating dealer program.

EARTHNET-PC is a trademark of EARTH COMPUTERS  
 NETWARE is a trademark of Novell  
 ViaNet is a trademark of Vlanetix, Inc.  
 TurboDOS is a trademark of Software 2000  
 IBM-PC/XT is a trademark of International Business Machines, Inc.



EARTH COMPUTERS

P.O. Box 8067, Fountain Valley, CA 92728  
 TELEX: 910 997 6120 EARTH FV

**(714) 964-5784**

Ask about EARTH COMPUTERS' other fine PC and S-100 compatible products.

```

150 A(D)=W-3*Q
160 W=Q
170 NEXT D
180 FOR D=N TO 1 STEP -1
190 IF A(D)<2 THEN 220
200 A(D)=A(D)-3
210 A(D-1)=A(D-1)+1
220 NEXT D
230 NG=0
240 PS=0
250 PWR=1
260 FOR D=N TO 1 STEP -1
270 A(D)=A(D)*PWR
280 PWR=PWR*3
290 ON SGN(A(D))+2 GOTO 300,350,330
300 NG=NG+1
310 N(NG)=ABS(A(D))
320 GOTO 350
330 PS=PS+1
340 P(PS)=A(D)
350 NEXT D
360 CLS
370 PRINT "Balance the scales this way"
380 LOCATE 14-PS,1
390 FOR D=1 TO PS
400 PRINT USING "#####";P(D)
410 NEXT D
420 PRINT
430 IF NG=0 THEN 480
440 FOR D=1 TO NG
450 LOCATE 12-NG+D,18
460 PRINT USING "#####";N(D)
470 NEXT D
480 LOCATE 13,18
490 COLOR 0,7
500 PRINT USING "##### ";SW
510 COLOR 7,0
520 PRINT "-----"
530 PRINT " |-----|"
540 PRINT " |-----|"
    
```

We have 12 weights, valued as follows...

1 3 9 27 81 243 729 2187 6561 19683 59049  
 177147

Enter the weight to be verified ( 1 - 265720 )  
 ? 301

Balance the scales this way

```

      1
      3
      81
      243
-----
|-----|
      ^
    
```

Ok

Listing 1b: Alternate lines with simplified I/O for BASICs without the LOCATE feature.

```

380 PRINT "Left side: ";
400 PRINT P(D),
430 PRINT "Right side: ";
450 REM line deleted
460 PRINT N(D),
480 REM line deleted
490 REM line deleted
500 PRINT "??";SW;"??"
510 REM line deleted
520 REM line deleted
530 REM line deleted
540 REM line deleted
    
```

a specified maximum number of weighings on a balance scale. No standard weights are available; the coins are to be weighed against each other. We are not allowed to add or remove coins during a weighing.

The simplest version of this problem involves eight coins, of which one is known to be heavy. In only two weighings, find the bad coin. Trying to solve this one will give you a greater understanding and appreciation of what follows.

Now for the big one, involving 12 coins, of which one is light or heavy (we don't know which in advance). Using three weighings, we are to find the bad coin and state whether it is light or heavy. Again, you are urged to give this one a try.

Start by numbering the coins from 1 to 12 for reference. Suppose we express those 12 reference numbers in ternary. Further

suppose that the result of each weighing (left pan down, right pan down, no difference) could give a new ternary digit, so that after three weighings we are left with a ternary number identifying the bad coin. That would be too easy!

In fact, the method we've sketched out does work, but it's not easy. The preparation is quite complicated (enter the computer to help out).

**Preparation Phase**

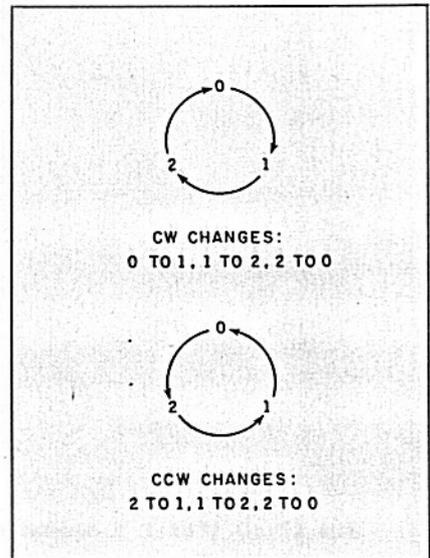
First, for each numbered coin, we need the ternary equivalent and the two's complement (also in ternary). To get the two's complement, subtract each digit from 2. (Equivalently, change each 0 to a 2, each 2 to a 0, and leave 1s unchanged.) For instance, coin #1 is 001<sub>3</sub>, which has a two's-complement representation of 221<sub>3</sub>. Table 1 lists the ternary and two's-complement

representations for all 12 coins.

The next step is to classify each of our ternary and two's-complement numbers as either "clockwise" or "counterclockwise." To do so, we read a number's digits from left to right and note the first change of digits. If the change is 0 to 1, 1 to 2, or 2 to 0, the number is clockwise. Otherwise, it is counterclockwise. The "clocks" in figure 2 should clarify these directions. In table 1, clockwise numbers are marked with an asterisk. Now consider the clockwise ternaries only. continued

**Table 1:** In preparation for solving the counterfeit-coin problem, the coins are numbered in decimal, ternary, and two's-complement ternary. ("Clockwise" coins are marked with an asterisk.)

Decimal	Ternary	Two's complement
1	*001	221
2	002	*220
3	*010	212
4	*011	211
5	*012	210
6	020	*202
7	021	*201
8	022	*200
9	100	*122
10	101	*121
11	102	*120
12	110	*112



**Figure 2:** Illustration of clockwise and counterclockwise digit changes, for use in classifying numbers. A number is clockwise if its first digit change (starting at the left) is clockwise; otherwise, it is counterclockwise.

# DataSaver® Standby UPS

**90** Watt. The original micro-UPS that brought standby AC power to Apple users in 1980. Still the POWER SOLUTION for serious applications of the MAC, small micros, and portables. Model 9012060 \$350. FCC (15A&B).

Now, POWER PROTECTION and MASTER CONTROL for full-size micros. Filters electrical noise/spikes and provides up to 10 minutes of AC power for the AT and supermicros. Model 40012060 \$695. UL-E101268.

**200** Watt. This DataSaver standby UPS offers compact, cool and quiet line conditioning and backup power to PC, XT, and COMPAQ markets. Data and hardware are protected. Model 20012060 \$495. CSA-LR51253.



**Cuesta Systems Corporation**  
3440 Roberto Court  
San Luis Obispo, California 93401

© 1986. Made with pride in the U.S.A.  
805/541-4160 FAX: 4949381 CUESTA  
Dealer, VAR, & OEM Inquiries Invited.

# C Programmers!

**db\_VISTA™: high-speed DBMS**  
written exclusively for C  
**NOW offers SQL-based query**

High-Speed data retrieval and access... just two benefits of using Raima's network model DBMS, db\_VISTA. Combine these design benefits with those of C—speed, portability, efficiency, and you begin to understand db\_VISTA's real measure... performance.

Says, **Dave Schmitt,**  
**President of Lattice, Inc.**

"If you are looking for a sophisticated C programmer's database, db\_VISTA is it. It lets you easily build complex databases with many interconnected record types. Raima's customer support and documentation is excellent. Source code availability and a royalty-free run-time is a big plus."

**db\_QUERY™: new simplicity**  
retains performance!

db\_QUERY, our new C-linkable, SQL-based, ad-hoc query and report writing facility... provides a simple, relational view of db\_VISTA's complex network database. No longer will you give up performance for simplicity... combine db\_QUERY with db\_VISTA... you have both!

- **Written in C**
- **Royalty-Free**
- **Source Code Available**
- **Multi-user and LAN capability**
- **Fast B-tree indexing method**
- **Transaction processing**
- **Faster without Data Redundancy**
- **Complete Documentation**
- **Operating systems:** MS-DOS, PC-DOS, UNIX, XENIX, SCO XENIX, UNOS, ULTRIX, VMS
- **C compilers:** Lattice, Microsoft, IBM, DeSmet, Aztec, Computer Innovations, XENIX and UNIX

**30-day Money-Back Guarantee**

### Royalty Free Price Schedule

	db_VISTA	db_QUERY
Single-user	\$ 195	\$ 195
Single-user w/Source	\$ 495	\$ 495
Multi-user	\$ 495	\$ 495
Multi-user w/Source	\$ 990	\$ 990
<b>NEW:</b>		
VAX Multi-user	\$ 990	\$ 990
VAX Multi-user w/Source	\$1980	\$1980

**FREE Technical Support**  
For 60 days

**Call Toll-Free Today!**

Order Line . . . . 1-800-327-2462  
Information Line 1-206-828-4636



**RAIMA™**  
CORPORATION

High-Speed Programming Tools,  
Designed for Portability

3055-112 N.E., Bellevue, WA 98004 USA  
(206) 828-4636 Telex: 6503018237 MCI UW

0287B

## MATHEMATICAL RECREATIONS

Listing 2a: *BASIC program to find a single off-weight coin using a balance scale with no standard weights.*

```

10 DIM A(120,5),R(5)
20 REM
30 REM
40 CLS: PRINT "Bad Coin Finder"
50 INPUT "How many weighings are to be allowed (2 TO 5)";N
60 IF N<2 OR N>5 THEN 50
70 C=INT((3^N-3)/2)
80 PRINT "Out of "; C; "coins, exactly 1 is bad (light or
heavy). I'll find it."
90 PRINT "Numbering the coins";
100 FOR K=1 TO C
110 PRINT ". ";
120 D=K
130 FOR J=N TO 1 STEP -1
140 Q=INT(D/3)
150 A(K,J)=D-3*Q
160 D=Q
170 NEXT J
180 NEXT K
190 PRINT
200 FOR K=1 TO C
210 J=0
220 J=J+1
230 DF=A(K,J)-A(K,J+1)
240 IF DF=0 THEN 220
250 IF DF=-1 OR DF=2 THEN 290
260 FOR L=1 TO N
270 A(K,L)=2-A(K,L)
280 NEXT L
290 NEXT K
300 PRINT "Pick out the bad coin number ( 1 to"; C; ") and
write it down."
310 INPUT "Press Return to start weighing";RT$
320 FOR W=1 TO N
330 CLS
340 PRINT "Weighing #"; W
350 CI=1
360 FOR J=1 TO C
370 IF A(J,W)=0 THEN LOCATE 12-((CI-1) MOD 10), 1+INT((CI-
1)/10)*8: PRINT J:CI=CI+1
380 NEXT J
390 CI=1
400 FOR J=1 TO C
410 IF A(J,W)=2 THEN LOCATE 12-((CI-1) MOD 10),41+INT((CI-
1)/10)*8: PRINT J:CI=CI+1
420 NEXT J
430 LOCATE 13,1
440 PRINT "-----" TAB(40)
"-----"
450 PRINT TAB(16) "|" TAB(56) "|"
460 PRINT TAB(16)
"-----"
470 PRINT TAB(36);"^"
480 LOCATE 18,1
490 PRINT "Which side is heavier? L=left, R=right,
N=neither: ";
500 WH$=INPUT$(1)
510 PRINT WH$;
520 V=INT((INSTR(1,"LINRr",WH$)+1)/2)
530 IF V=0 THEN 480 ELSE R(W)=V-1
540 NEXT W
550 COIN=0
560 FOR K=1 TO N
570 COIN=3*COIN+R(K)
580 NEXT K

```

continued

```

590 LOCATE 18,1
600 PRINT STRING$(80,32)
610 LOCATE 18,1
620 IF COIN>C THEN COIN=INT(3^N-1-COIN)
630 IF COIN<1 OR COIN>C THEN PRINT"Impossible! Please try
again.": END
640 PRINT "Coin "; COIN; "is ";
650 D=0
660 D=D+1
670 DF=R(D)-R(D+1)
680 IF DF=0 THEN 660
690 IF DF=-1 OR DF=2 THEN PRINT "HEAVY" ELSE PRINT "LIGHT"
700 END
    
```

**Bad Coin Finder**

How many weighings are to be allowed (2 TO 5)? 5  
 Out of 120 coins, exactly 1 is bad (light or heavy).  
 I'll find it. Numbering the coins.....

Pick out the bad coin number ( 1 to 120 ) and  
 write it down. Press Return to start weighing?

**Weighing # 1**

14	33	43	53	23	60	70	80
13	32	42	52	22	59	69	79
12	31	41	51	21	58	68	78
11	30	40	50	20	57	67	77
10	29	39	49	19	56	66	76
9	28	38	48	18	55	65	75
5	27	37	47	8	54	64	74
4	17	36	46	7	26	63	73
3	16	35	45	6	25	62	72
1	15	34	44	2	24	61	71

Which side is heavier? L=left, R=right, N=neither: R

**Listing 2b: Alternate lines with simplified I/O for BASICs without the LOCATE feature.**

```

350 PRINT: PRINT"Left side: ";
370 IF A(J,W)=0 THEN PRINT J;
390 PRINT: PRINT "Right side: ";
410 IF A(J,W)=2 THEN PRINT J;
430 PRINT
440 REM Line deleted
450 REM Line deleted
460 REM Line deleted
470 REM Line deleted
480 PRINT
510 REM Line deleted
590 PRINT
600 REM Line deleted
610 REM Line deleted
    
```

The complicated preparation is over, and we are ready to perform some ternary magic.

**The Weighing-in**

For the first weighing, we place in the left pan every coin whose first (leftmost) digit is 0. In the right pan we place every coin whose first digit is 2. If the left pan goes down (is heavy), we write a 0. If the pans balance, we write a 1. If the right pan goes down, we write a 2.

The second weighing is similar, except we look at each coin's second digit. Into the left pan go the coins whose second digit is 0; into the right pan go the coins whose second digit is 2. After checking the balance, we write down a 0 (left pan down), 1 (balanced), or 2 (right pan down), as before.

For the third weighing, coins whose third (rightmost) digit is 0 go in the left pan; coins whose third digit is 2 go in the right pan. Write down a 0, 1, or 2, as before.

We have now generated a three-digit ternary number. We find the number in table 1 and read off the corresponding coin number from column 1. That's our bad coin. If the ternary number is clockwise, the coin is heavy; if the ternary number is counterclockwise, the coin is light.

Simple? Not at all. But the method works, and it can be generalized to handle problems allowing  $n$  weighings of  $(3^n-3)/2$  coins.

For instance, if seven weighings are allowed, we can find the one bad coin among 1092 coins. But applying the method manually would be virtually impossible because of the paperwork involved. Which explains why you won't find reference to a 1092-coin problem in any puzzle book.

Nevertheless, the program in listing 2 brings computer power to the task, and it lets us handle any arbitrary number of weighings. The program sets an upper limit of 5 to allow simulation of a scale on the screen.

To raise the limit, replace the value 5 with a larger number in lines 10, 50, and 60. In this case, you will also need to include the alternate lines in listing 2b.

In the program, the computer asks you to pick out the bad coin and write down its number. The computer then simulates each weighing, asking you to tell it which "pan" is heavier. At the end of  $n$  weighings, the computer will identify the bad coin by number and specify whether it is light or heavy.

I would appreciate hearing from readers who have comments on these ternary puzzles or can suggest other mathematical recreations involving alternate number systems. ■