COMPUTER RECREATIONS

The search for an invisible ruler that will help radio astronomers to measure the earth

by A. K. Dewdney

A simple ruler one foot long, bearing 13 inch marks, can measure 12 discrete lengths. Is it possible to improve this familiar device so that it measures more lengths than there are marks on the ruler? The answer is yes: it is possible to remove all but five marks from the standard ruler and still measure 10 distances with it. Each distance will be found between some pair of marks as the difference between the integers that label them. It is even possible to achieve the same result with an 11-inch ruler. Readers who puzzle over this exercise and finally succeed will have created a Golomb ruler.

The search for such rulers is an engaging task in which the computer can be useful. What elevates the project from a curiosity to a first-class conundrum is that the need for Golomb rulers emerges from a variety of scientific and technical disciplines.

The devices are the invention of Solomon W. Golomb, professor of mathematics and electrical engineering at the University of Southern California. For two decades he and a handful of colleagues have sought the rulers and studied their properties. The rulers may be applied in coding theory, X-ray crystallography, circuit layout and radio astronomy.

Among the investigators whose work Golomb rulers enlarge and expedite is Douglas S. Robertson, a geophysicist who works for the U.S. National Geodetic Survey of the National Oceanic and Atmospheric Administration in Rockville, Md. He uses the radio-astrometric technique known as very-long-baseline interferometry (or VLBI for short) not to map radio sources but to make finely tuned measurements of the earth. Having sought the rulers himself for a number of years, he unhesitatingly appeals to readers to widen the search. The result may be both a more accurate determination of the size, shape and motion of our planet and a more intriguing time spent thereon.

Before trying to answer Robertson's appeal it is worth mastering the principles that underlie Golomb rulers. Although the rulers come in all sizes, only the smaller ones are known. The first three rulers can be described somewhat abstractly by three sequences of numbers:

0, 1
0, 1, 3
0, 1, 4, 6

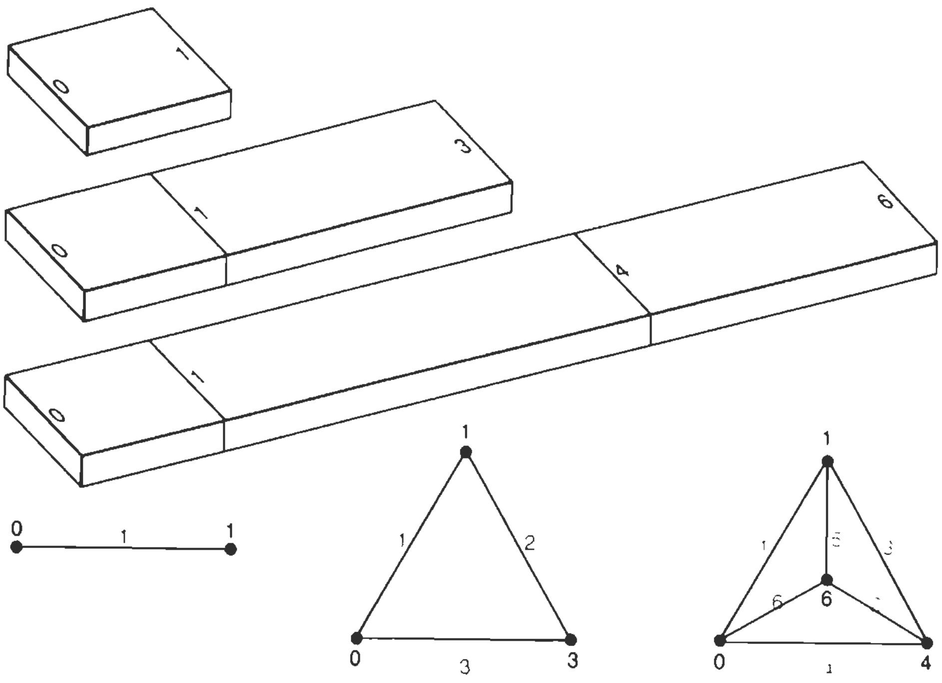
To construct the physical ruler mark the left end of a blank ruler that is n units long with the smallest number (0) in the sequence. Inscribe the right end with the largest number (n). The largest number can be 1, 3 or 6. Intervening integers should accompany marks placed at appropriate intervals from the 0 end of the ruler [*see illustration on this page*].

A simple way to check which possible distances from 0 to n can be measured by one of these small rulers is to draw the ruler's distance diagram. For each integer on the ruler mark a point on a sheet of paper and label it with the integer. Then join each pair of points by a line that is labeled with the difference between the integers at its endpoints. If every integral distance encompassed by 0 through n appears on only one line of the distance diagram, the Golomb ruler is said to be perfect. The three rulers in the illustration are all perfect, a fact that can be verified by a glance at their distance diagrams. In each diagram no distance appears more than once and every possible distance between 0 and n is present.

Perfection is rare among Golomb rulers. In fact, the only perfect rulers that exist have just been described. For values of n higher than 6 imperfection is manifested in one of two ways: either a distance occurs more than once or it does not occur at all.

This is a cruel reality to face so early in the search for bigger (and better) rulers. How do we know that no larger perfect rulers exist? Golomb has supplied a proof that is short as well as charming.

His suggestion is that one consider not the marks on a ruler but the intervals between them. If the ruler is perfect, it turns out that the intervals between consecutive marks must provide the distances 1, 2, 3,... up to $m - 1$ in some order (m is the number of marks). Golomb asks: "Where is the one-unit space?" If it is next to any space whose length is less than $m - 1$, then the two spaces together yield a distance that is less than m. This dis-



The three perfect Golomb rulers and their distance diagrams

A3022

nce must already occur as a space sewhere because all distances from 1 – 1 occur between consecutive . Such reasoning forces us to ac-pt the somewhat startling conclusion at the one-unit space is next to the pace that is $m - 1$ units long. More-ver, there is no space on the other side f the unit space. It lies at one end of e ruler.

The foregoing argument constitutes ne turn of the crank on what some athematicians call a sausage ma-hine. "Where is the two-unit space?" olomb asks. The crank is turned gain: if the two-unit space is next to ny space whose length is less than $m - 2$, the two spaces together pro-uce a distance that already occurs lsewhere. This time we cannot con-lude that the two-unit space is next to space of length $m - 2$. Their com-ined length, m, is already measured ointly by the one- and $(m - 1)$-unit paces at one end of the ruler.

The sausage machine grinds to a alt, producing the conclusion that here is only one space the two-unit pace may lie next to, namely the one whose length is $m - 1$. Since any ruler has only two ends, a perfect ruler has at most three spaces, 1, $m - 1$ and 2. The proof is complete when we realize t hree spaces require four marks: 4. The spaces are therefore 1, 3, 2; the corresponding marks are 0, 1, 4, 6.

Faced with a complete lack of perfect rulers that have more than four marks, a mathematician will cut the losses by constructing a new definition. What might be called the "next-best syndrome" is thereby demonstrated: the next-best thing to an impossible perfect five-mark ruler might be one that contains each distance only once but does not contain all the distances a perfect ruler of the same length would have. Since this condition is easily met by allowing a ruler to be long enough, a rider is attached. Among all five-mark rulers that contain each distance at most once, determine the shortest one. Such a ruler is called a Golomb ruler of order five. Golomb rulers of order m are defined in the same way. Since the definition includes the first three rulers as a special case, it bridges the awkward discontinuity in perfection beyond four marks.

Herbert Taylor, a colleague of Golomb's, has summarized the current state of information about Golomb rulers in a table [*see illustration on this page*]. From two to 24 marks there th certain knowledge and some guesswork about the size of Golomb rulers. What I call the zone of perfection extends from two to four marks. Thereafter the zone of knowledge embraces the Golomb rulers having up to 13 marks. All the rulers here are known to be minimum. That is to say, in each case there is no shorter Golomb ruler that has the same number of marks. A Golomb ruler of five marks has length 11. A Golomb ruler of 13 marks has length 106.

Beyond 13 marks lies what I call the twilight zone. Dignified as the zone of research, it contains only rulers not yet known to be Golomb. For each number of marks there is a ruler that has the length given in the table. But shorter rulers may exist. Indeed, there is a formula that provides a lower limit for these lengths. A steadily widening gap between formula values and rulers so far found attests either to a weakness in the formula or to increasingly poor rulers.

Robertson is responsible for extending the knowledge zone to include 13-mark rulers. In a computer run that lasted for a month his program exhaustively searched through all potential Golomb rulers bearing 13 marks and found the shortest one. It would probably interest very few readers to search for Golomb rulers that have 14 or more marks if runs longer than this are needed.

Instead it seems reasonable to suggest some probing techniques, computational raids into the research zone that promise some return in the form of better rulers. Basic to any such effort is a program called CHECKER. CHECKER addresses the following task: Given an array of integers, what is the most efficient way to determine whether the differences between them are all unique? The simpleminded approach generates all possible pairs of integers and stores their differences in another array. Then it checks the file for duplicates relying on an awkward and time-consuming algorithm.

Rarely does the faster way to do a job require a shorter program, but here is a case. Since the differences themselves are supposed to be distinct, they can be used as addresses in a special array called *check*. Initially only 0's are stored in *check*. Each time a new difference is calculated the value stored at the appropriate address is changed from 0 to 1. Thus as CHECKER proceeds with its computations it may find a 1 already stored at a particular address, implying that the "new" difference has actually been seen before. In such a case the ruler cannot be Golomb, because it does not pass the fundamental test of Golombicity: each distance must be generated only once.

The technique of using differences as addresses constitutes a primitive form of what computer scientists and programmers call hashing. In many information-retrieval settings, hashing is the fastest way for a computer to recall a file.

	NUMBER OF MARKS	SHORTEST RULER KNOWN	LOWER BOUND
ZONE OF PERFECTION	2	1	
	3	3	
	4	6	
ZONE OF KNOWLEDGE	5	11	
	6	17	
	7	25	
	8	34	
	9	44	
	10	55	
	11	72	
	12	85	
	13	106	
ZONE OF RESEARCH	14	127	114
	15	155	133
	16	179	154
	17	199	177
	18	216	201
	19	246	227
	20	283	254
	21	333	283
	22	358	314
	23	372	346
	24	425	380

Lengths of Golomb and near-Golomb rulers

In more detail for those who require it, here is the essence of CHECKER. Two nested loops are used to generate all possible pairs of integers from the input array. If the first loop generates i and the second loop generates j, the program computes the absolute value of their difference and stores it in a variable called *diff*. In the next step CHECKER uses the value of *diff* as a kind of hash code: in algorithmic language one can write the following:

if *check*(*diff*) = 1
  then output "non-Golomb" and exit
  else *check*(*diff*) ← 1.

If the program never says "non-Golomb," the ruler has passed the main test. But how short is it? There are a number of ways to find out.

First, it is possible to use CHECKER in the stand-alone mode. I can imagine a reader hunched over the keyboard running only that program. He or she is exploring the research zone at an altitude of 14 marks, looking for a ruler shorter than 127 units, the best ruler currently known. The reader, flying in IFR weather, has no idea which way to turn. He has just submitted a sequence of 14 marks. The largest integer in the sequence is 124 and the excitement is almost too much as the display screen springs to life: "Congratulations. The set is OK." In programming this message he vowed never to try CHECKER on anything but potentially record-breaking sets.

Perhaps the reader found his record-breaking set by following Golomb's advice and exploring only those rulers in which the largest space appears in the middle. The spaces on such a ruler become smaller toward the ends of the ruler but they do so at the reader's discretion. Golomb assures us that many good rulers, if not necessarily the best ones, follow this pattern.

CHECKER can be modified to suit a more tentative style of inquiry. In STEP CHECKER the integers are typed in one at a time. After each entry the program generates the differences between the integer just entered and those already stored. In fact, STEP CHECKER is simply a version of CHECKER in which an input statement replaces the outer loop. The sequence is successful if the last integer has been digested and the program has not printed "non-Golomb."

Finally, the program STEP CHECKER can be incorporated into an automated search of the kind undertaken by Robertson. His program (which I may as well call EXHAUST because it is exhaustive) generates new rulers by adding one space at a time systematically. After each addition STEP CHECKER decides whether the ruler currently under construction is valid.

Robertson constructed his program by visualizing a ruler to which new spaces (and so new marks) are added left to right. Readers who followed this trail of prose through the byway of Golomb's argument (proving the nonexistence of perfect rulers) will remember that the spaces that must occur had the lengths 1, 2,..., $m - 1$ in some order. Although this is true only of perfect rulers, something similar is true of Golomb rulers in general. Most but not all of the lengths from 1 to $m - 1$ between consecutive marks on less than perfect rulers occur in some order. Yet some spaces even longer than $m - 1$ can be found within such rulers.

Robertson generates new spaces in a stepwise manner. He maintains them in an array I shall call, appropriately, *spaces*. EXHAUST traverses the array adding one space after another. Naturally there are some simple tests that ease the labors of EXHAUST. One of these is to be sure that when a new space is generated it does not already occur in *spaces*. A second test is to sum all the spaces making up the current ruler to confirm that their sum does not exceed the shortest length known.

The EXHAUST program in operation seems eager to find rulers. It sets the first element of *spaces* to 1 and adds units to the second space so that it is different from the first. Then it adds units to the third space so that this distance not only is different from the first two distances but also satisfies the requirement set by STEP CHECKER, namely that all distances contained in the ruler must be different from one another. Each time a new entry of *spaces* is decided in this way, EXHAUST adds up the array and compares the sum with the length of the shortest ruler yet known. If the sum is less, the program continues to the next entry. If it is not less, EXHAUST returns to the preceding entry and continues to add spaces there.

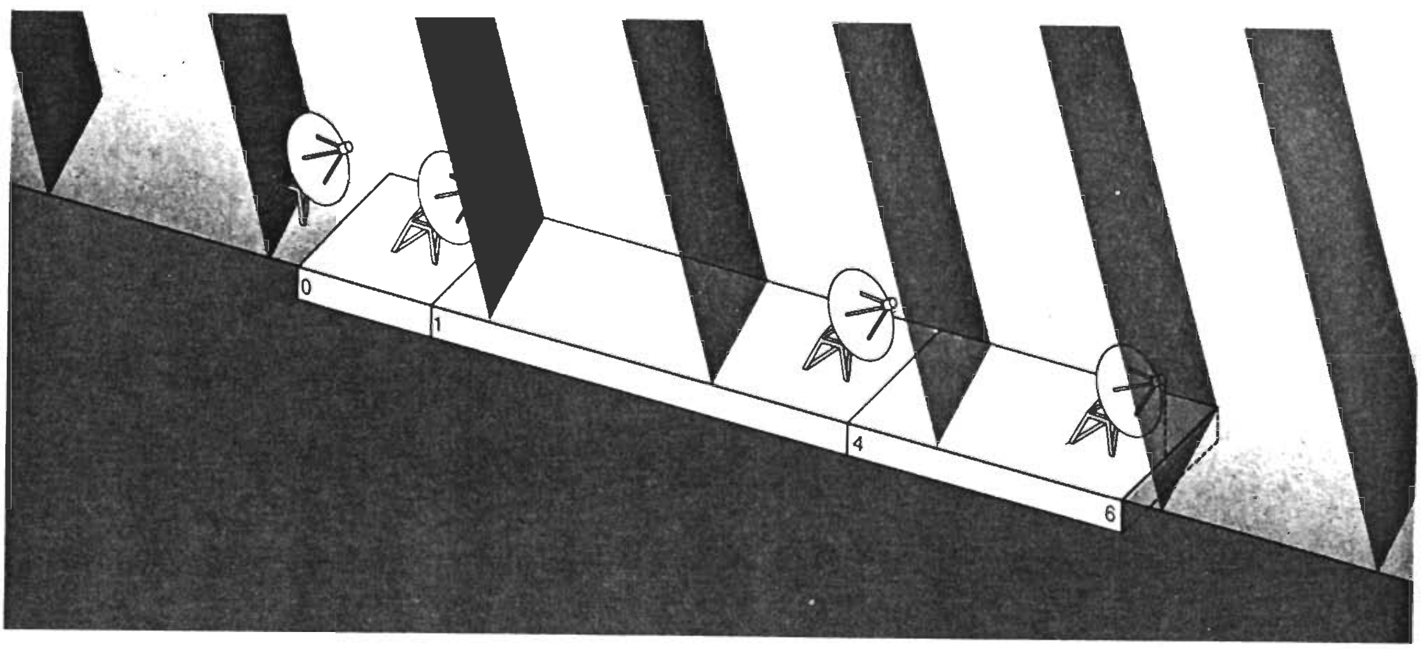
Robertson's program will run marginally faster if the first element of *spaces* is set to 2 instead of 1. Indeed, a one-day run will be shortened by a few hours. Readers may want to ponder why the search is still exhaustive.

Surely the effectiveness of an exhaustive search program depends on the inclusion of further tests and heuristics. Additional limitations on the values assumed by various entries in the *spaces* array would particularly enhance efficiency. Perhaps there is an incrementing procedure that uses much smaller ranges of such values. Processing the array is akin to counting. The count is reached much sooner if the number of possibilities at each entry of *spaces* is reduced. In any event, the readers who found busier beavers, new glider guns and other benefits to research will doubtless make their own way into the Golomb research zone.

New rulers should be sent to Golomb at the University of Southern California, University Park, Los Angeles, Calif. 90089. The most remarkable finds will be published in a future column.

Golomb has offered a prize of $100 to the first person who finds two different rulers that have more than six marks and yet measure the same set of distinct distances. Rulers that are mirror images of each other are not regarded as different.

A positive result would ring the death knell for a "theorem" propounded by Sophie Piccard, a Swiss mathematician, in 1939. Piccard's theorem states that two rulers measuring the same set of distinct distances must be the same rulers. The theorem was embraced by X-ray crystallographers because it helped them to resolve ambiguities in diffraction patterns. Unfortunately the theorem fails for numerous pairs of rulers that have six marks. Perhaps it is true for all rulers



Pairs of radio-telescope antennas set up on a Golomb ruler can reveal phase differences between incoming signals of higher order. Readers may pursue Golomb's prize without venturing into the research zone. It is a question that can be investigated for rulers bearing as few as seven marks.

How does all this relate to helping Robertson? Radio astronomy makes occasional use of Golomb rulers in the resolution of distant radio sources and in the measurement of our own planet. In the first case a number of antennas are placed along a straight line several kilometers long. The antenna positions correspond to the marks on a Golomb ruler [*see illustration on page 22*]. To locate a distant radio source, it is essential to determine the angle between the antenna baseline and the direction of wave fronts arriving from the source. The antennas are all observing at a given wavelength. The precise time at which each wave in the incoming signal arrived at each antenna can be determined by analysis of the tape that captures the incoming signal. The total number of wavelengths between a given pair of antennas is called the total phase difference. It is normally composed of an integer and a fractional part called the phase difference. If the total phase difference can be reconstructed, the sought-for angle between the source and the baseline is easily calculated from the observing wavelength and *c*, the speed of light. Each pair of antennas, however, can only yield the phase difference itself, not the total phase difference.

In truth, it is Fourier analysis that recaptures the total phase difference from the many pairs of antenna recordings. But if the distance between one pair of antennas is the same or nearly the same as the distance between another pair, the two pairs provide the same phase-difference information. Redundancy of information means its loss. The accuracy of the source-angle computation is greatest if each antenna pair records a different phase difference; this condition is achieved by in effect placing the antennas on the marks of a Golomb ruler.

Another way to locate a distant radio source is to use just two receivers, each scanning an entire set of wavelengths simultaneously. Observing with two antennas a distant source at several different wavelengths yields the same information about the total phase difference between the antennas as would the use of several antennas tuned to the same wavelength. Here the same threat of redundancy looms again. No two pairs of scanning wavelengths should be the same distance apart, so to speak. The Golomb ruler is invisible but nonetheless present.

Robertson uses the second technique not to map radio sources but to locate the antennas themselves. For his purpose it is not enough to know that the second antenna is in Westford, Mass. He needs to know its position to within a few centimeters. The precision of such location is possible if a very distant, pointlike radio source such as a quasar is used. To locate a single point on the earth's surface with respect to a distant radio source is tantamount to the precise determination of such fundamental earthly variables as diameter, spin orientation and length of day. At the level of centimeters or microseconds such variables are truly that, posting annual, seasonal and even meteorological variations that are sometimes meaningful and at other times mysterious.

The September "Computer Recreations" column described CRABS, a benign terminal illness. Hordes of marine crustaceans occasionally and without warning descend hungrily on the Blit terminals of scientists at the AT&T Bell Laboratories in Murray Hill, N.J.

Almost every aspect of the Blit terminal's multiprogramming environment relies on an important instruction called bitblt. Bitblt transfers a rectangular set of bits from one area of the Blit terminal's memory to another. Two such rectangular sets can be combined by various logic operators before the transfer is effected.

Readers were challenged to solve two bitblt puzzles: erasing a picture from the screen and rotating a picture by 90 degrees. The first puzzle is easy to solve. To erase a picture occupying a given rectangular set, combine the set with itself using the XOR operator. The result consists of nothing but zeros. A single bitblt command replaces the original picture with a blank screen.

Rotating a picture is much harder than erasing it. So far only one reader has submitted a workable solution. Thomas Witelski, a high school student in New York City, has found a way to rotate an $n \times n$ picture on a $2n \times 2n$ screen. Using only $3n - 1$ bitblt operations, he slides $n - 1$ rows, copies an $n - 1 \times n - 1$ subpicture, slides $n - 1$ columns, copies another subpicture and then slides $n - 1$ more rows. Witelski's method is faster than the standard rotation algorithm, which requires $4n - 2$ operations.

There have been heartfelt cries from readers who do not feel competent to write their own version of the MANDELZOOM program but would like to run it on their personal computer. When readers write to ask for listings or disks of the programs I describe, there is a temptation to comply. But it would simply take too long to fill all the orders I get. Moreover, the time a reader spends learning the few elements of programming necessary for most of the projects I describe is time well spent. The teacher in me is pleased at the thought.

When other readers generously offer their own programs for distribution, however, I am tempted beyond my ability to resist. Following are the names and addresses of six individuals who are willing to supply programs under varying conditions and in varying states of accuracy and beauty. Caveat emptor:

Mark W. Bolme
Token Software
P.O. Box 3746
Bellevue, Wash. 98009
(IBM PC and Apple II family)

Pete Gwozdz
21865 Regnart Road
Cupertino, Calif. 95014
(IBM PC)

Will Jones
609 Rochester Avenue
Coquitlam, British Columbia
CANADA V3K 2V3
(IBM PC)

Bradley Dyck Kliewer
3001 East 24 Street
Minneapolis, Minn. 55406
(IBM PC with or without 8087 chip)

Charles Platt
9 Patchin Place
New York, N.Y. 10011
(IBM PC)

Richard A. Tilden
10 Thurston Street
Somerville, Mass. 02145
(Zenith-100)

Just as some readers have trouble getting started in the fine art of programming, so others, more expert, have an urge to share. It has occurred to me that a form of computer buddy system might be implemented with a minimum of administrative detail. It would link two classes of people: readers who expect to need help with the programs in "Computer Recreations" and readers who are willing to be of help. Am I wrong in supposing that the latter are in adequate supply? Send me a postcard with your name, full address, telephone number and category (tyro or adviser). Readers willing to advise should specify the maximum number of tyros they feel capable of helping. I shall endeavor to make matches within reasonable geographic distances.

Dewdney, postscript A3022

puter program that can score at the genius level on an I.Q. test. Does the score on the test measure the intelligence of the computer? If it does not, just how does one go about measuring the intelligence of a computer, whether it is made of silicon and plastic or of carbon and tissue? The answer: Probably not by running some I.Q. program through a battery of tests.

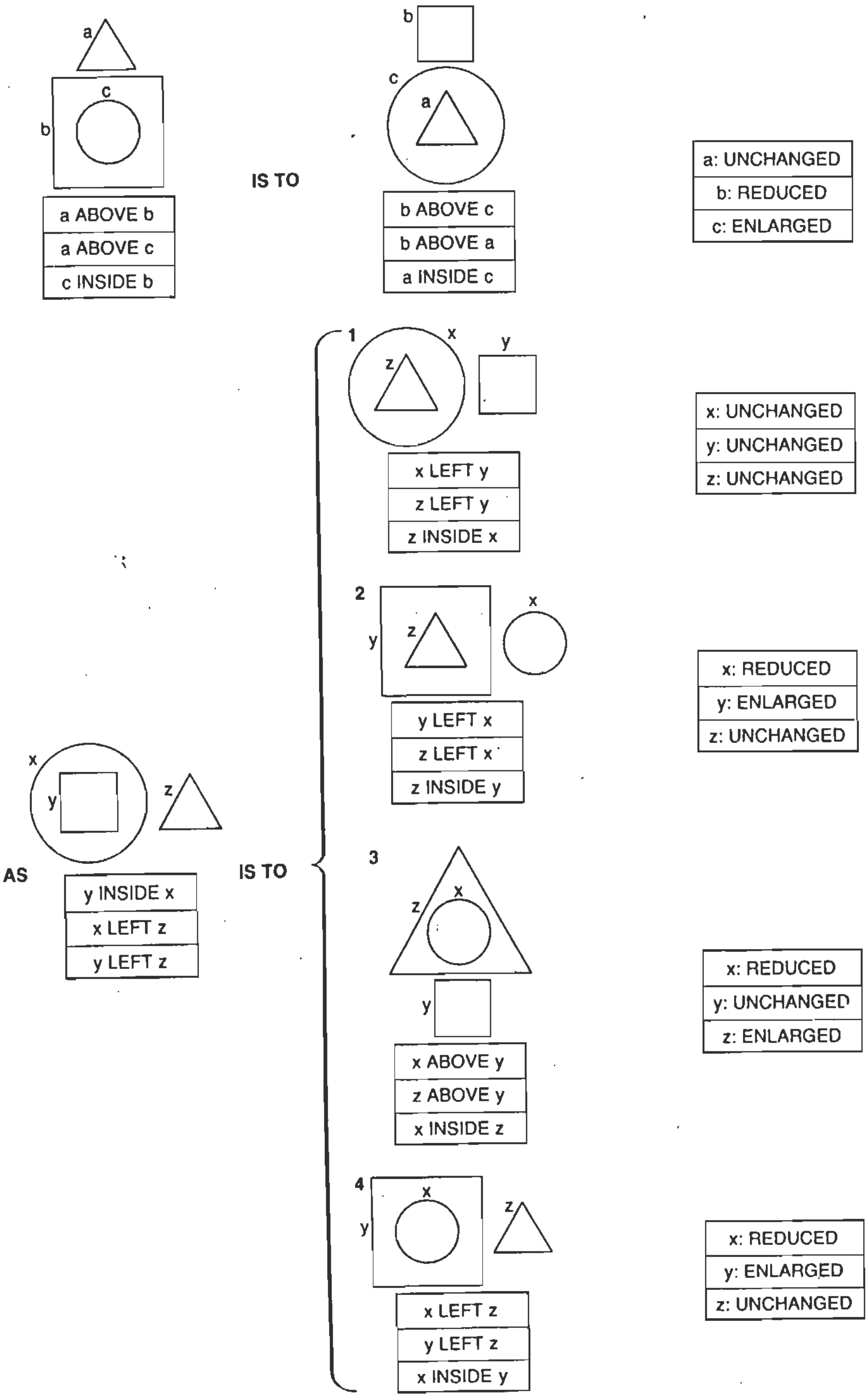
Golomb rulers, the subject of last December's column, turned out to be the toughest project that readers have yet faced. Many were called but few were chosen, so to speak. Several readers even sought to claim a $100 prize offered by the inventor of the rulers, Solomon W. Golomb of the University of Southern California.

A Golomb ruler with *n* marks is the shortest ruler possible with the following properties: it bears *n* distinct marks (including the endpoints) at integer positions, and it measures as many integral lengths as possible from 1 to the length of the ruler, each length in at most one way. A distance can be measured by the ruler only if it is the distance between some pair of marks. If the same distance can be measured between more than one pair of marks, the ruler is not a Golomb ruler.

At the time the December column was published, no Golomb rulers were known with more than 13 marks, and the shortest ruler known with 15 marks was 155 units long. Soon thereafter Douglas S. Robertson of the National Oceanic and Atmospheric Administration discovered a shorter 15-mark ruler only 153 units long. Then during the Christmas holidays James B. Shearer of the IBM Thomas J. Watson Research Center programmed an idle computer to search exhaustively for rulers, and the computer has now turned up Golomb rulers with 14 and 15 marks. The 14-mark Golomb ruler is 127 units long and has marks at 0, 5, 28, 38, 41, 49, 50, 68, 75, 92, 107, 121, 123 and 127. The 15-mark Golomb ruler is 151 units long and has marks at 0, 6, 7, 15, 28, 40, 51, 75, 89, 92, 94, 121, 131, 147 and 151. Shearer writes that he saved much computing time by assuming the middle mark on the ruler is to the left of the geometric middle.

Another problem posed by Golomb has generated the claims for the $100 prize. The claims made so far are invalid, apparently because they are based on misunderstandings of the problem. Golomb has urged me to clarify matters by restating it. Find two different rulers (whether of minimal length or not), each having the same number of marks for some number greater than 6, that measure the same set of distances; again, no distance on either ruler can be measured between more than one pair of marks. Reflections, such as the ruler with marks at 0, 2, 5, 6 and the ruler with marks at 0, 1, 4, 6, are not counted as different. There are infinitely many known pairs of rulers, almost all of them nonminimal, that solve the analogue of Golomb's problem for six marks. For example, one such pair have marks respectively at 0, 1, 4, 10, 12, 17 and at 0, 1, 8, 11, 13, 17. They are nonreflecting, essentially different rulers, but they both measure all distances between 1 and 17 except 14 and 15. The prize will go to the first person who discovers such a pair of rulers with more than six marks each.

The advisory network to help programming novices with projects stemming from this department has run into unforeseen difficulties: there are hundreds of advisers but almost no tyros. The name tyro may have been ill-advised. Has it put people off who program with little success? It is time to send me a card bearing your name, address and telephone number, in care of this magazine.



The solution of visual analogies in Thomas G. Evans' dissertation