Algorithms in *Everyday Mathematics*

What is an algorithm?

An algorithm is a well-defined procedure or set of rules guaranteed to achieve a certain objective. You use an algorithm every time you follow the directions to put together a new toy, use a recipe to make cookies, or defrost something in the microwave.

In mathematics, an algorithm is a specific series of steps that will give you the correct answer every time. For example, in grade school, you and your classmates probably learned and memorized a certain algorithm for multiplying. Chances are, no one knew why it worked, but it did!

In *Everyday Mathematics*, students first learn to understand the mathematics behind the problems they solve. Then, quite often, they come up with their own unique working algorithms that prove that they "get it." Through this process, they discover that there is more than one algorithm for computing answers to addition, subtraction, multiplication, and division problems. Having students become comfortable with algorithms is essential to their growth and development as problem solvers.

How do students learn to use algorithms for computation?

Ideally, students should develop a variety of computational methods and the flexibility to choose the procedure that is most appropriate in a given situation. *Everyday Mathematics* includes a variety of standard computational algorithms, as well as students' invented procedures. The program leads students through three phases as they learn each mathematical operation (addition, subtraction, multiplication, and division).

Algorithm invention

In the early phases of learning an operation, students are encouraged to invent their own methods for solving problems. This approach requires students to focus on the meaning of the operation. They learn to think and use their common sense, as well as new skills and knowledge. Students who invent their own procedures:

- learn that their intuitive methods are valid and that mathematics makes sense.
- become more proficient with mental arithmetic.
- are motivated because they understand their own methods, as opposed to learning by rote.
- become skilled at representing ideas with objects, words, pictures, and symbols.
- develop persistence and confidence in dealing with challenging problems.

Alternative Algorithms

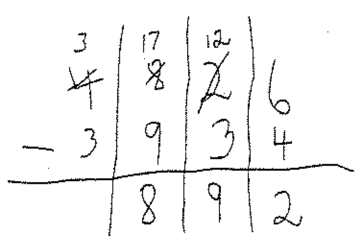
After students have had many opportunities to experiment with their own computational strategies, they are introduced to several algorithms for each operation. Some of these algorithms may be the same or similar to the methods students have already invented on their own. Others are traditional algorithms which have commonly been taught in the U.S. or simplifications of those algorithms. And others are entirely new algorithms that have significant advantages in today's technological world.

Students are encouraged to experiment with various algorithms and to become proficient with at least one.

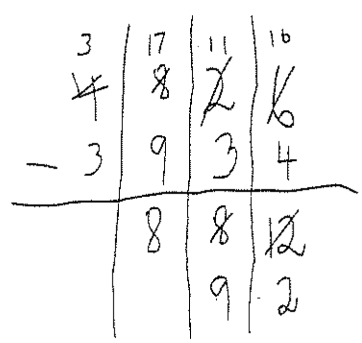
Demonstrating Proficiency

For each operation, the program designates one alternative algorithm as a "focus" algorithm. Focus algorithms are powerful, relatively efficient, and easy to understand and learn. They also provide common and consistent language, terminology, and support across grade levels of the curriculum.

All students are expected to learn and demonstrate proficiency with the focus algorithm. Once they can reliably use the focus algorithm, students may use it or any alternative they prefer when solving problems. The aim of this approach is to promote flexibility while ensuring that all students know at least one reliable method for each operation.



trade-first subtraction with columns



trade-first subtraction with an unnecessary trade

Addition Algorithms

This section presents just a few of the possible algorithms for adding whole numbers.

Focus Algorithm: Partial-Sums Addition

You can add two numbers by calculating partial sums, working one place-value column at a time, and then adding all the sums to find the total.

Example: Partial-Sums Addition

	268
	+ 483
Add the hundreds (200 + 400).	600
Add the tens (60 + 80).	140
Add the ones (8 + 3).	+ 11
Add the partial sums (600 + 140 + 11).	751

Column Addition

To add using the column-addition algorithm, draw vertical lines to separate the ones, tens, hundreds, and so on. Add the digits in each column, and then adjust the results.

For some students, the above process becomes so automatic that they start at the left and write the answer column by column, adjusting as they go without writing any of the intermediate steps. If asked to explain, they might say something like this:

"200 plus 400 is 600. But (looking at the next column) I need to adjust that, so I write 7. 60 and 80 is 140. But that needs adjusting, so I write 5. 8 and 3 is 11. With no more to do, I can just write 1."

Example: Column Addition

Add the digits in each column.

hundreds	tens	ones
2	6	8
+ 4	8	3
6	14	11

Since 14 tens is 1 hundred plus 4 tens, add 1 to the hundreds column, and change the number in the tens column to 4.

hundreds	tens	ones
2	6	8
+ 4	8	3
7	4	11

Since 11 ones is 1 ten plus 1 one, add 1 to the tens column, and change the number in the ones column to 1.

hundreds	tens	ones
2	6	8
+ 4	8	3
7	5	1

Opposite-Change Rule

If you add a number to one part of a sum and subtract the same number from the other part, the result remains the same. For example, consider:

$$8 + 7 = 15$$

Now add 2 to the 8, and subtract 2 from the 7:

$$(8 + 2) + (7 - 2) = 10 + 5 = 15$$

This idea can be used to rename the numbers being added so that one of them ends in zeros.

> ***Example:* Opposite-Change Rule**
>
> Rename the first number and then the second.
>
> | | Add 2. → | | Add 30. → | |
> |---|---|---|---|---|
> | 268 | | 270 | | 300 |
> | + 483 | | + 481 | | + 451 |
> | | Subtract 2. → | | Subtract 30. → | 751 |
>
> Rename the second number and then the first.
>
> | | Subtract 7. → | | Subtract 10. → | |
> |---|---|---|---|---|
> | 268 | | 261 | | 251 |
> | + 483 | | + 490 | | + 500 |
> | | Add 7. → | | Add 10. → | 751 |

Subtraction Algorithms

There are even more algorithms for subtraction than for addition, probably because subtraction is more difficult. This section presents several subtraction algorithms.

Focus Algorithm: Trade-First Subtraction

This algorithm is similar to the traditional U.S. algorithm except that all the trading is done before the subtraction, allowing children to concentrate on one thing at a time.

> ***Example:* Trade-First Subtraction**
>
> Examine the columns. You want to make trades so that the top number in each column is as large as or larger than the bottom number.
>
> | hundreds | tens | ones |
> |---|---|---|
> | 9 | 3 | 2 |
> | − 3 | 5 | 6 |
> | | | |
>
> To make the top number in the ones column larger than the bottom number, borrow 1 ten. The top number in the ones column becomes 12, and the top number in the tens column becomes 2.
>
> | hundreds | tens | ones |
> |---|---|---|
> | | 2 | 12 |
> | 9 | ~~3~~ | ~~2~~ |
> | − 3 | 5 | 6 |
> | | | |
>
> To make the top number in the tens column larger than the bottom number, borrow 1 hundred. The top number in the tens column becomes 12, and the top number in the hundreds column becomes 8.
>
> | hundreds | tens | ones |
> |---|---|---|
> | | 12 | |
> | 8 | ~~2~~ | 12 |
> | ~~9~~ | ~~3~~ | ~~2~~ |
> | − 3 | 5 | 6 |
> | | | |
>
> Now subtract column by column in any order.
>
> | hundreds | tens | ones |
> |---|---|---|
> | | 12 | |
> | 8 | ~~2~~ | 12 |
> | ~~9~~ | ~~3~~ | ~~2~~ |
> | − 3 | 5 | 6 |
> | 5 | 7 | 6 |

Counting Up

To subtract using the counting-up algorithm, start with the number you are subtracting (the subtrahend), and "count up" to the number you are subtracting from (the minuend) in stages. Keep track of the amounts you count up at each stage. When you are finished, find the sum of the amounts.

Example: **Counting Up**

To find 932 – 356, start with 356 and count up to 932.

356
Add 4 to count up to the nearest ten.
360
Add 40 to count up to the nearest hundred.
400
Add 500 to count up to the largest possible 100.
900
Add 32 to count up to 932.
932

Now find the sum of the numbers you added.

$$\begin{array}{r} 4 \\ 40 \\ 500 \\ +\ 32 \\ \hline 576 \end{array}$$

So, 932 – 356 = 576.

Left-to-Right Subtraction

To use this algorithm, think of the number you are subtracting as a sum of ones, tens, hundreds, and so on. Then subtract one part of the sum at a time.

Example: **Left-to-Right Subtraction**

To find 932 – 356, think of 356 as the sum 300 + 50 + 6. Then subtract the parts of the sum one at a time, starting from the hundreds.

	932
Subtract the hundreds.	– 300
	632
Subtract the tens.	– 50
	582
Subtract the ones.	– 6
	576

Same-Change Rule

If you add or subtract the same number from both parts of a subtraction problem, the results remain the same. Consider, for example:

$$15 - 8 = 7$$

Now add 4 to both the 15 and the 8:

$$(15 + 4) - (8 + 4) = 19 - 12 = 7$$

Or subtract 6 from both the 15 and the 8:

$$(15 - 6) - (8 - 6) = 9 - 2 = 7$$

The same-change rule algorithm uses this idea to rename both numbers so the number being subtracted ends in zeros.

***Example:* Same-Change Rule**

Add the same number.

	Add 4.		Add 40.	
932	→	936	→	976
− 356		− 360		− 400
			Subtract.	576

***Example:* Same-Change Rule**

Subtract the same number.

	Subtract 6.		Subtract 50.	
932	→	926	→	876
− 356		− 350		− 300
			Subtract.	576

Partial-Differences Subtraction

The partial-differences subtraction algorithm is a fairly unusual method, but one that appeals to some students.

The procedure is fairly simple: Write partial differences for each place, record them, and then add them to find the total difference. A complication is that some of the partial differences may be negative.

***Example:* Partial-Differences Subtraction**

		932
		− 356
Subtract 100s.	900 − 300	600
Subtract 10s.	30 − 50	− 20
Subtract 1s.	2 − 6	− 4
Add the partial differences.		576

Multiplication Algorithms

Students' experiences with addition and subtraction algorithms can help them invent multiplication algorithms. For example, when estimating a product mentally, many students begin to compute partial products: "Ten of these would be. . ., so 30 of them would be. . ., and we need 5 more, so. . ." Beginning in *Third Grade Everyday Mathematics*, this approach is formalized as the partial-products multiplication algorithm. This algorithm and others are discussed in this section.

Focus Algorithm: Partial Products

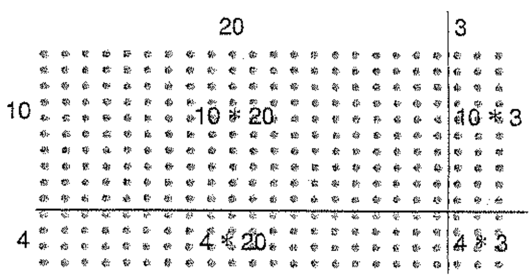
To use the partial-products algorithm, think of each factor as the sum of ones, tens, hundreds, and so on. Then multiply each part of one sum by each part of the other, and add the results.

Rectangular arrays can be used to demonstrate visually how the partial-products algorithm works. The product 14 * 23 is the number of dots in a 14-by-23 array. The diagram below shows how each of the partial products is represented in the array.

Example: Partial Products

To find 67 * 53, think of 67 as 60 + 7 and 53 as 50 + 3. Then multiply each part of one sum by each part of the other, and add the results.

	67
	* 53
Calculate 50 * 60.	3,000
Calculate 50 * 7.	350
Calculate 3 * 60.	180
Calculate 3 * 7.	+ 21
Add the results.	3,551



14 × 23 = (10 + 4) * (20 + 3)
= (10 * 20) + (10 * 3) + (4 * 20) + (4 * 3)
= 200 + 30 + 80 + 12
= 322

Modified Repeated Addition

Many students are taught to think of whole-number multiplication as repeated addition. However, using repeated addition as a computation method is inefficient for anything but small numbers. For example, it would be extremely tedious to add fifty-three 67s in order to compute 67 * 53. Using a modified repeated addition algorithm, in which multiples of 10, 100, and so on, are grouped together, can simplify the process.

Example: Modified Repeated Addition

Think of 53 * 67 as fifty 67s plus three 67s. Since ten 67s is 670, fifty 67s is five 670s.

So, 53 * 67 is five 670s plus three 67s.

67
* 53
670
670
670
670
670
67
67
67
3,551

Lattice Multiplication

Everyday Mathematics initially included the lattice method for its recreational value and historical interest (it has been used since A.D. 1100 and appeared in the first printed arithmetic book, published in 1478) and because it provided practice with multiplication facts and adding single-digit numbers. This method has become a favorite of many students in *Everyday Mathematics*.

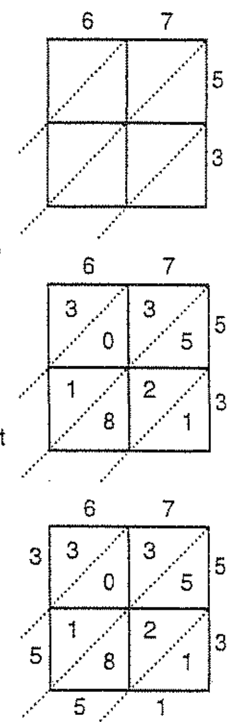
The following example shows how the method is used to find 67 * 53.

Example: Lattice Multiplication

Follow these steps to find 67 * 53.

- Draw a 2-by-2 lattice, and write one factor along the top of the lattice and the other along the right. (Use a larger lattice to multiply numbers with more digits.)
- Draw diagonals from the upper-right corner of each box, extending beyond the lattice.
- Multiply each digit in one factor by each digit in the other. Write each product in the cell where the corresponding row and column meet. Write the tens digit of the product above the diagonal and the ones digit below the diagonal. For example, since 6 * 5 = 30, write 30 in the upper-left box with the 3 above the diagonal and the 0 below.
- Starting with the lower-right diagonal, add the numbers inside the lattice along each diagonal. If the sum along a diagonal is greater than 9, carry the tens digit to the next diagonal.

  The first diagonal contains only 1, so the sum is 1. The sum on the second diagonal is 5 + 2 + 8 = 15. Write only the 5, and carry the 1 to the next column. The sum along the third diagonal is then 1 + 3 + 0 + 1, or 5. The sum on the fourth diagonal is 3.
- Read the product from the upper left to the lower right. The product is 3,551.



Division Algorithms

One type of division situation involves making as many equal-size groups as possible from a collection of objects: How many dozens can you make with 746 eggs? How many 5-passenger cars are needed for 37 people? Such problems ask, "How many of these are in that?" More generally, a / b can be interpreted as "How many bs are in a?" This idea forms the basis for the division algorithms presented in this section.

Focus Algorithm: Partial Quotients

The partial-quotients algorithm uses a series of "at least, but less than" estimates of how many bs are in a.

Example: Partial Quotients

Estimate the number of 12s in 158.

You might begin with multiples of 10 because they are simple to work with. There are at least ten 12s in 158 (10 * 12 = 120), but there are fewer than twenty (20 * 12 = 240). Record 10 as a first estimate, and subtract ten 12s from 158, leaving 38.

Now estimate the number of 12s in 38.

There are more than three (3 * 12 = 36) but fewer than four (4 * 12 = 48). Record 3 as the next estimate, and subtract three 12s from 38, leaving 2.

Since 2 is less than 12, you can stop estimating. The final result is the sum of the estimates (10 + 3 = 13) plus what is left over (the remainder of 2).

12)158	10	first guess
120		
38	3	second guess
36		
2	13	sum of guesses

158 / 12 → 13 R2

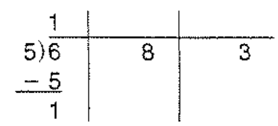
Column Division

Column division is a simplification of the traditional long division algorithm you probably learned in school, but it is easier to learn. To use the method, you draw vertical lines separating the digits of the divisor and work one place-value column at a time.

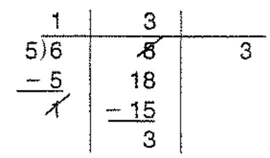
Example: Column Division

To find 683 / 5, imagine sharing $683 among 5 people. Think about having 6 hundred-dollar bills, 8 ten-dollar bills, and 3 one-dollar bills.

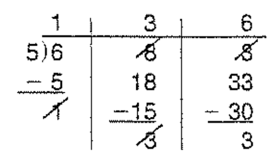
First, divide up the hundred-dollar bills. Each person gets one, and there is one left over.



Trade the leftover hundred-dollar bill for 10 ten-dollar bills. Now you have a total of 18 ten-dollar bills. Each person gets 3, and there are 3 left over.



Trade the 3 leftover ten-dollar bills for 30 one-dollar bills. You now have a total of 33 one-dollar bills. Each person gets 6, and there are 3 left over.



So, when you divide $683 among 5 people, each person gets $136, and there are $3 left over. So, 683 / 5 = 136 R3.