NOTTINGAM WEST ELEMENTARY SCHOOL

Everyday Mathematics Parent Handbook

Algorithms and Arithmetic in Everyday Mathematics

(Developed by Ann Arbor Michigan Curriculum & Instruction Department)

An algorithm is a set of rules for solving a math problem which, if done properly, will give a correct answer each time.

Algorithms generally involve repeating a series of steps over and over, as in the borrowing and carrying algorithms and in the long multiplication and division algorithms. The Everyday Mathematics program includes a variety of suggested algorithms for addition, subtraction, multiplication and division. Current research indicates a number of good reasons for this — primarily, that students learn more about numbers, operations, and place value when they explore math using different methods.

Arithmetic computations are generally performed in one of three ways: (1) mentally, (2) with paper and pencil, or (3) with a machine, e.g. calculator or abacus. The method chosen depends on the purpose of the calculation. If we need rapid, precise calculations, we would choose a machine. If we need a quick, ballpark estimate or if the numbers are "easy," we would do a mental computation.

The learning of the algorithms of arithmetic has been, until recently, the core of mathematics programs in elementary schools. There were good reasons for this. It was necessary that students have reliable, accurate methods to do arithmetic by hand, for everyday life, business, and to support further study in mathematics and science. Today's society demands more from its citizens than knowledge of basic arithmetic skills. Our students are confronted with a world in which mathematical proficiency is essential for success. There is general agreement among mathematics educators that drill on paper/pencil algorithms should receive less emphasis, and that more emphasis be placed on areas like geometry, measurement, data analysis, probability and problem solving, and that students be introduced to these subjects using realistic problem contexts. The use of technology, including calculators, does not diminish the need for basic knowledge, but does provide children with opportunities to explore and expand their problem solving capabilities beyond what their pencil-and-paper arithmetic skills may allow.

Sample Algorithms: Below are examples of a few procedures that have come from children's mental arithmetic efforts. Each is a legitimate algorithm, that is, a set of rules that if properly followed yields a correct result. As parents, you need to be accepting and encouraging when your children attempt these computational procedures. As they experiment and share their solution strategies, please allow their ideas to flourish. If you are not comfortable with the vocabulary of arithmetic, you may want to review the glossary entries for addition, subtraction, multiplication and division before reading the sample algorithms.

Addition Algorithms

1. Left-to-right Algorithm

A. Starting at the left, add column-by-column, and adjust the result.

	2	6	8
	+4	8	3
1. Add	6	14	11
2. Adjust 10's and 100's	7	4	11
3. Adjust 1's and 10's	7	5	1

B. Alternate procedure: For some students this process becomes so automatic that they start at the left and write the answer column by column, adjusting as they go without writing any in-between steps. If asked to explain, they say something like this:

2	6	8
+4	8	3
6^1	4^1	1
7	5	1

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

This technique easily develops from experiences with manipulatives, such as base-10 blocks and money, and exchange or trading games, and is consistent with the left-to-right patterns learned for reading and writing.

2. Partial-Sums Algorithm

Add the numbers in each column. Then add the partial sums.

Students who use this type of algorithm often show more awareness of place value than those who learned the traditional method. This procedure works well for larger numbers too.

	268
	+483
1. Add 100's	600
2. Add 10's	140
3. Add 1's	+11
4. Add partial sums	751

3. Rename-Addends Algorithm (Opposite Change)

If a number is added to one of the addends and the same number is subtracted from the other addend, the result remains the same. The purpose is to rename the addends so that one of the addends ends in zeros.
This strategy indicates a good number sense and some understanding of equivalent forms.

A. Rename the first addend, and then the second.

268	->	(+2)	->	270	->	(+30)	->	300
+483	->	(-2)	->	+481	->	(-30)	->	+451
							Add	751
Explanation: Adjust by 2, and then by 30.								

B. Rename the first addend, and then the second.

268	->	(-7)	->	261	->	(-10)	->	251
+483	->	(+7)	->	+490	->	(+10)	->	+500
							Add	751
Explanation: Adjust by 7, and then by 10.								

4. Counting-On Algorithm

A. Rename the first addend, and then the second.

268 + 483

Begin at 268 and count by 100’s, 4 times: 368, 468, 568, 668; then count by 10’s, 8 times: 678, 688, 698, 708, 718, 728, 738, 748; continue to count by l’s, 3 times: 749, 750, 751.

B. Counting-on algorithm alternate method

With larger numbers children may use a combination of counting on and counting back. Begin at 268 and count by 100’s, 5 times: 368, 468, 568, 668, 768; then count back by 10’s, twice: 758, 748; continue to count by 1’s, 3 times: 749, 750, 751.

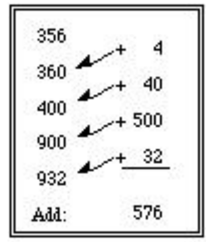
Subtraction Algorithms

1. Add-Up Algorithm

Add up from the subtrahend (bottom number) to the minuend (top number).

932
-356

Students may mentally keep track of the numbers that are added or use paper to record the addends on the side. Most of us often use some form of this method when making change.



2. Left-to-Right Algorithm

Starting at the left, subtract column by column.

932
-356

1. Subtract 100's	932 -300
2. Subtract 10's	632 -50
3. Subtract 1's	582 -6
	576

3. Rename Subtrahend Algorithm (also called Same Change)

If the same number is added to or subtracted from both the minuend (top number) and subtrahend (bottom number), the result remains the same. The purpose is to rename both the minuend and the subtra-hend so that the subtrahend ends in zero.

This type of solution method shows a strong ability to hold and manipulate numbers mentally.

A. Add the same number

932	->	(+4)	->	936	->	(+44)	->	976
-356	->	(+4)	->	-360	->	(+40)	->	-400
							Subtract	576
Explanation: Adjust by 4, and then by 40.								

B. Add the same number

932	->	(-6)	->	930	->	(+54)	->	976
-356	->	(-6)	->	-350	->	(+50)	->	-400
							Subtract	576
Explanation: Adjust by 6, and then by 50.								

4. Two Unusual Algorithms

A. Subtract by adding column-by-column with adjustments. (Same problem as above.) Some students who use the add-up algorithm extend that to subtraction. They just write the answer with no other remarks. Asked to explain, they say something like this:
"To get to 900 from 300, add 600; but the tens need help, so make it 5 [for 500]. To get to 130 from 50, add 80; but the ones need help, so write 7 [for 70]. To get to 12 from 6, add 6. No more to do."

B. Write partial differences, negative if necessary, and adjust. A few students who love negative numbers use some variation of the procedure shown here.
This method may be less common than some of the others. Yet, some students seem to have an informal sense of working with negatives (deficits).

	932
	-356
1. Subtract 100's: 900-300	600
2. Subtract 10's: 30-50	-20
3. Subtract 1's: 2-4	-4
4. Add the partial differences	576
(600-20-4, done mentally)	

Multiplication Algorithms

In Third Grade Everyday Mathematics, a "partial-products" algorithm is the initial approach to solving multiplication problems with formal paper-and-pencil proce-dures. This algorithm is done from left to right, so that the largest partial product is calculated first. As with left-to-right algorithms for addition, this encourages quick estimates of the magnitude of products without neces-sarily finishing the procedure to find exact answers. To use this algorithm efficiently, students need to be very good at multiplying multiples of 10, 100, and 1000. The fourth-grade program contains a good deal of practice and review of these skills, which also serve very well in making ballpark estimates in problems that involve multiplication or division, and introduces the * as a symbol of multiplication.

1. Partial-Product Algorithm

In the partial-product multiplication algorithm, each factor is thought of as a sum of ones, tens, hundreds, and so on. For example, in 67 * 53, think of 67 as 60 + 7, and 53 as 50 + 3. Then each part of one factor is multiplied by each part of the other factor, and all of the resulting partial products are added together.

	67
	*53
50 x 60	3000
50 x 7	350
3 x 60	180
3 x 7	+21
	3551

This method reinforces the understanding of place value and emphasizes the multiplication of the largest product first.

2. Modified Standard U.S. Algorithms

Example *a* is the standard U.S. algorithm.
Example *b* replaces the blank with a zero, which makes it clear that for the second partial product, we are multiplying by 50 (five 10's) and not just by 5.
Example *c* works from left to right, but is otherwise the same as the standard algorithm with zero in place of the blank. This method may be less common than some of the others yet students using it are able to do arithmetic easily and apply it in many different situations.

a.		b.		c.	
	67		67		67
	*53		*53		*53
	201		201		3350
	+335		+3350		+201
	3551		3551		3551

3. The Lattice Method

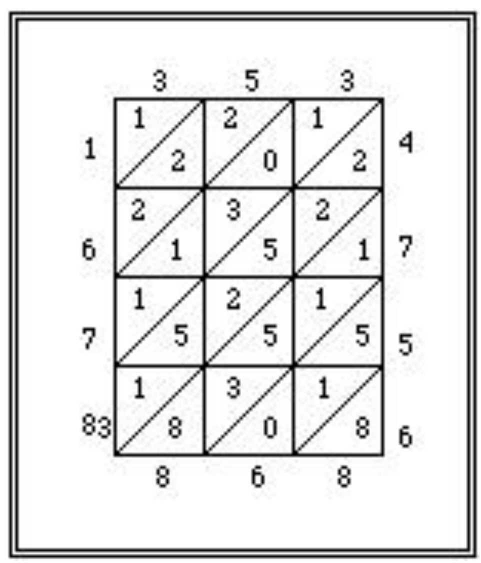
This algorithm is included mainly for its historical interest, and the fact that it provides fine practice with the multiplication facts and adding single-digit numbers. It is not easy to explain exactly why it works, but it does have the reliability that all algorithms must have. It is also very efficient, no matter how many digits are in the factors, as indicated by the second example below.

The lattice method appeared in what is said to be the first printed arithmetic book, printed in Treviso, Italy, in 1478. It was in use long before that, with some historians tracing it to Hindu origins in India before 1100.

This is a student favorite because of the direct relationship to known multiplication facts and its easy expandability to very large numbers.

Lattice Method Explanation

- First, make a table of the right size: one column for each digit of the first factor and one row for each digit of the second factor. Divide each cell of the resulting table in half by drawing a diagonal line.
- Write the digits for the first factor across the top (one for each box).
- Write the digits for the second factor down the right side (one for each box).
- Multiply each digit across the top with the digit(s) down the right side, placing the tens digit of each product above the diagonal of the cell, and the ones digit below the diagonal. For example, the upper left hand cell contains the answer 12, the result of multiplying the 3 (first top digit) by 4 (first right-side digit). The second box in the top row is the result of multiplying 5 (second top digit) by 4 (first right-side digit). And so on.
- Start at the bottom right corner and add the digits upward along the diagonal. Place the sum at the bottom of the diagonal (outside the box) carrying the tens digit to the next diagonal, if needed. Move left one column, and do the next diagonal. Continue along the bottom, then turn the corner and do the diagonals up the left side.



The answer is read down the left side and across the bottom, left to right.

A Division Algorithm

The key question to be answered in many problems is, “How many of these are in that,” or “How many n's are in m?” This can be expressed as division: “m divided by n,” or “m/n.”

One way to solve division problems is to use an algorithm that begins with a series of “at least/less than” estimates of how many n’s are in m. You check each estimate. If you have not taken out enough n’s from the m’s, take out some more; when you have taken out all there are, add the interim estimates.

For example, 158/12 can be thought of as the question, “How many 12’s are in 158?” You might begin with multiples of 10, because they are simple to work with. A quick mental calculation tells you that there are at least ten 12’s in 158 (10 * 12 = 120), but less than twenty (since 20 * 12 = 240).

12)158	
-120	10
38	
-36	+3
2	13

You would record 10 as your first estimate and remove (subtract) ten 12’s from 158, leaving 38. The next question is, “How many 12’s are in the remaining 38?” You might know the answer right away (since three 12’s are 36), or you might sneak up on it: “More than 1, more than 2, a little more than 3, but not as many as 4.” Taking out three 12’s leaves 2, which is less than 12, so you can stop estimating.

To obtain the final result, you would add all of your estimates (10 + 3 = 13) and note what, if anything is left over (2). There is a total of thirteen 12’s in 158; 2 is left over. The quotient is 13, and the remainder is 2.

It is important to note that, in following this algorithm, students may not make the same series of estimates. In the example, a student could have used 2 as a second estimate, taking out just two 12’s and leaving 14 still not accounted for—another 12, and a remainder of 2. The student would reach the final answer in three steps rather than two. One way is not better than another.

12)158	
-120	
38	10
-24	
14	2
-12	
2	+1
	13

The examples show one method of recording the steps in the algorithm.

One advantage of this algorithm is that students can use numbers that are easy for them to work with. Students who are good estimators and confident of their extended multiplication facts will need to make only a few estimates to arrive at a quotient, while others will be more comfortable taking smaller steps. More important than the course a student follows is that the student understands how and why this algorithm works and can use it to get an accurate answer.

Another advantage of this algorithm is that it can be extended to decimals once students have a pretty good sense of "How many n's are in m?" Sometimes it may be desirable to express the quotient as a decimal. Sometimes n may be larger than m (the divisor larger than the dividend), or all the information is in decimal form. For the example 158 / 12, the estimates could be continued by asking, "How many 12's in the remainder 2?"

12)158.0	
-120.0	10.0
38.0	
-36.0	3.0
2.0	
-1.2	+0.1
.8	13.1

A student with good number sense might answer, "At least one-tenth, since 0.1 * 12 is 1.2, but less than two-tenths, since 0.2 * 2 = 2.4. The answer then could be l3.1 (12's) in 158, and a little bit left over."

The question behind this algorithm, "How many of these are in that?" also serves well for estimates where the information is given in "scientific notation" (see glossary). The uses of this algorithm with problems that involve scientific notation or decimal information will be explored briefly in grades 5 and 6, mainly to build number sense and understanding of the meanings of division.

Summary

An algorithm is any series of steps which, if followed properly, always yield a correct result. There are many ways to add, subtract, divide, and multiply that meet this definition. Your child will learn to compute accurately and quickly.

Children gain valuable confidence and insight when permitted to explore algorithms of their own invention. A given child may be more comfortable with this way or that. A given approach may be more useful for this problem or that one.

Although you probably learned only one or two algorithms for each kind of arithmetic, it is important that you support your child's use of many. In fact, if you closely observe your own computations in a variety of real-life settings — counting change, making estimates, balancing your checkbook, etc. — you will probably find that you use different algorithms at different times, and some of them are probably your own inventions.