Session 3

Functions and Algorithms

Key Terms for This Session

Previously Introduced

- variable [Session 2]

New in This Session

- algorithm
- function
- function machine
- network
- iterate
- whole number
- factor
- prime number
- polygon

Introduction and Review

In Session 1, we looked at patterns in pictures, charts, and graphs to determine how different quantities are related. In Session 2, we used patterns and variables to describe relationships in tables and in situations like toothpicks and triangles. This session extends the exploration of relationships to include the concepts of algorithm and function. **[SEE NOTE 1]**

Learning Objectives

In this session, we will explore algorithms and functions. We will:

- Understand the importance of doing and undoing in mathematics
- Determine when a process can or cannot be undone
- Use function machines to picture and undo algorithms
- Understand that functions produce unique outputs

NOTE 1. In this session, we will explore algorithms and functions. We will use function machines to illustrate function as algorithm, to picture operations as machines, and to provide a visual image of inputs and outputs.

When people begin to move from arithmetic to algebra, they start thinking about properties of operations—specifically about "what undoes what." Function machines are used both to build algorithms and to provide experience in thinking about how to "undo," or create inverses of, those algorithms.

We will also focus on the concept of uniqueness, looking at non-numerical examples to illustrate the uniqueness of functions. A key idea in the study of functions is that inputs must give unique outputs, but outputs may not have unique inputs. If the latter is the case, these functions cannot be undone.

By the end of the session, we will see that the same function can be expressed by different rules or algorithms, something that we have seen informally in Sessions 1 and 2.

NOTE 1 cont'd. next page

Part A: Doing and Undoing (15 MINUTES)

There are many situations, both in and outside of mathematics, where the process of doing and undoing helps you organize your activities and figure out how to reverse what you've done. In mathematics, it is often important to know how to undo an operation. Here are some examples from everyday life and mathematics: **[SEE NOTE 2]**

- School buses pick up children every morning and then drop them off in the same spots every afternoon. Routes are usually organized by a "first on, last off" routine.
- You put on socks and then shoes every morning, and you take off shoes and then socks every night.
- If you added 3 to a number and got 724, you can get your original number back by subtracting 3.

Sometimes you do things that can't be undone:

- If the cover comes off the hot pepper shaker while you're sprinkling it on the pizza, there's not much you can do to undo the process.
- If you mix blue laundry detergent and water, you'd have a hard time separating them back into their original components.
- If you subtracted 10 from a number, then multiplied the result by itself, you wouldn't be able to find, with certainty, the original number just from undoing the steps.

Problem A1. How can you tell that you wouldn't be able to definitively find the original number in the numerical rule given above, in which 10 is subtracted from a number and then that number is multiplied by itself? **[SEE TIP A1, PAGE 81]**

Write and Reflect

Problem A2. Give some examples from teaching, mathematics, or anywhere else where doing and undoing comes into play.

Problem A3. Give an example of something you wish you could undo, but the undoing is impossible.

NOTE 1, CONT'D.

Review

Groups: Discuss any questions about the homework. Look at the descriptions of the rules found for Problem H1, and, if it has not already been done, write the rule using symbols. Discuss the notation in $x^2 + 1$, especially reviewing the meaning of the exponent. This session contains a brief problem on the function $y = x^4$, and later, Session 7 contains extensive work with exponential functions, so it's good to preview some of these ideas.

Another interesting question on the homework is how we can know that 102 wouldn't appear in the "output" column of the table (as long as the "inputs" remained integers). One explanation: The input of 10 gives 101, and the input of 11 gives 122. Since the function seems to be only increasing, it has skipped 102. Since each number in the table is 1 more than a perfect square, another good explanation would be that 101 is not a perfect square, so 102 cannot appear in the table. **Groups:** Consider sharing solutions and representations for the frog in the well problem, as well.

NOTE 2. This exercise on "doing and undoing" will be our first look at functions.

Read through the examples of things that can be done and undone, and things that cannot be undone. **Groups:** Discuss the examples as a large group. You may want to discuss the number puzzle that cannot be undone. Then, take five minutes to discuss Problems A2 and A3 in pairs.

Part B: Undoing Algorithms (20 MINUTES)

An algorithm is a recipe or a description of a mechanical set of steps for performing some task. For example, you can have an algorithm for making a peanut butter and jelly sandwich.

Mathematical algorithms are increasingly important in the computer age. Computer programs are essentially algorithms written in a language that computers understand.

Here's a mathematical algorithm (let's call it Algorithm A): [SEE NOTE 3]

Algorithm A

- Pick a number (that's the input)
- Double it
- Add 2 to the answer
- Divide that answer by 2
- Subtract 7 from what you get
- Multiply the result by 4 (that's the output)

Problem B1. Use Algorithm A for these problems.

a. If the input is 9, what is the output?

b. If the input is 10, what is the output?

c. If the input is *n*, what is the output? [SEE TIP B1, PAGE 81]

d. If the output is 28, what is the input?

e. If the output is 32, what is the input?

f. What input produces an output of 48?

g. What input produces an output of 36?

Write and Reflect

Problem B2. What strategies did you use to answer parts (d) through (g) of Problem B1?

Problem B3. Describe, in language similar to the way we described Algorithm A, an algorithm (call it Algorithm B) that undoes Algorithm A. This means that if you put a number into Algorithm A, then put that output into Algorithm B, you should end up with the original input.

Algorithm B

NOTE **3. Groups:** Discuss your responses to Problems B2 and B3. Then discuss the video segment of Dr. Fuji monitoring medication for a newborn. How is math being used at Boston Medical Center? Where else might the concepts of doing and undoing have real-world applications?

Part B, cont'd.

Video Segment (approximate times: 7:29-9:04): You can find this segment on the session video approximately 7 minutes and 29 seconds after the Annenberg/CPB logo. Zero the counter on your VCR clock when you see the Annenberg/CPB logo.

In this segment, Professor Cossey works with Lolita and Deanna to find the algorithm that undoes Algorithm A, then they discuss why such an algorithm would have its steps reversed. Watch it after you have completed Problem B3. If you get stuck on the problem, you can watch the video segment to help you.

Professor Cossey refers to "inverse operations." Give some examples. Do all operations have inverse operations?

Problem B4. Does Algorithm A undo Algorithm B? That is, if you put a number into Algorithm B and then put that output into Algorithm A, do you get back to your starting number?

Video Segment (approximate times: 21:52-24:36): You can find this segment on the session video approximately 21 minutes and 52 seconds after the Annenberg/CPB logo. Zero the counter on your VCR clock when you see the Annenberg/CPB logo.

In this segment, Dr. Fujii of the Boston Medical Center describes the importance of doing and undoing in prescribing medication to newborns. Watch this segment after you have completed Part B-Undoing Algorithms. The segment is taken from the "real world" example at the end of the Session 3 video.

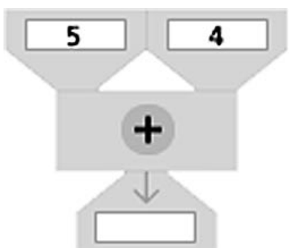
What inverse operations does Dr. Fujii use? How are the questions he answers similar to those you answered in Problem B1?

Part C: Function Machines (35 MINUTES)

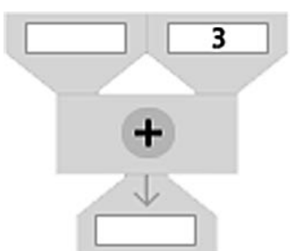
About Function Machines

When thinking about inputs and outputs, you are thinking about algorithms as *functions*: You *input* a number into the algorithm, follow the prescribed steps, and get an *output*. To be a function, there are two requirements. First, the algorithm must be consistent—that is, every time you give it the same input, you get the same output. Second, each input must produce exactly one possible output. **[SEE NOTE 4]**

Some people picture the steps in an algorithm or function as little machines. An addition machine would look like this:

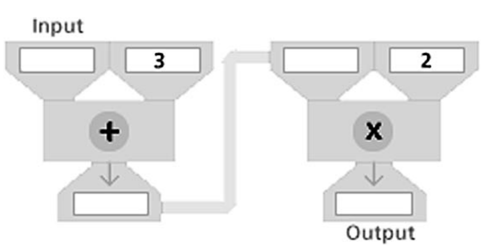


If you put a 5 in the left hopper and a 4 in the right hopper, what would come out of the bottom? Take a few minutes to practice drawing your own machines.



Sometimes we want our machine to add the same thing each time. If we wanted our machine to add 3 each time, we could represent it with the 3 locked in position, as shown at left.

We can also connect two different machines together to make a network, so that the output hopper from one machine goes right into the input hopper on the next machine. Consider the network of function machines below.



Try running a few numbers through this network. Remember to perform each step in order. For example, if you start with 0, you will get 3 as an output from the first machine. After running 3 though the second machine, your output will be 6, which is the output of the network.

Now try pulling a number back up through the machine in the reverse order—that is, begin with the output and work backward to the input.

NOTE 4. We mentioned earlier that moving from arithmetic to algebra involves a focus on operations rather than numbers. Think about an operation like addition. What do you see when you think about addition? Don't think of an addition problem like 4 + 5, but just plain addition with no numbers—focus on the operation rather than on the numbers themselves. Next, go through the descriptions of input/output machines. After looking at the first example, be creative in coming up with drawings of machines. **Groups:** Share drawings with the whole group.

Some machines add the same thing each time. Machines like this can be connected together, so that the output hopper from one machine goes right into the input hopper on the next machine. The two connected machines are really a single function that takes an input, runs it through the network inside, and produces an output. Read through the next written description of an algorithm and go through the exercises. **Groups:** Answer Problems C1-C11 in pairs or small groups.

Part C, cont'd.

Running a Function Machine

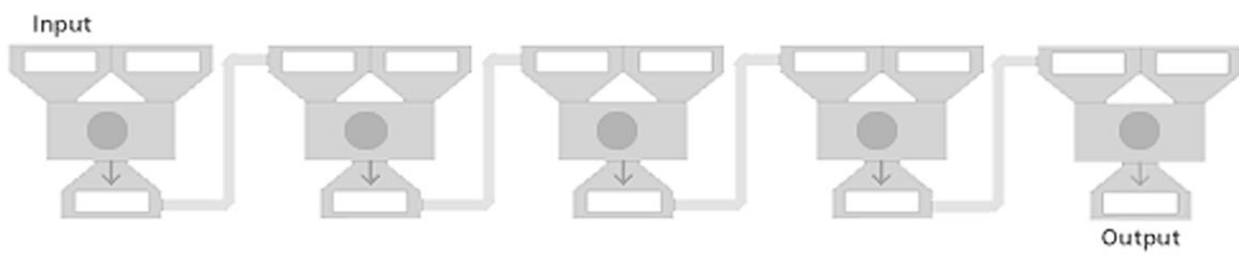
Our connected machine is really a single function that takes an input, runs it through the network inside, and produces an output.

Take a look at this algorithm (let's call it Algorithm C):

Algorithm C

- Pick a number (that's the input)
- Multiply it by 2
- Add 3 to the answer
- Multiply that answer by 10
- Add 6
- Finally, divide by 2

Fill in the function machine network below with Algorithm C to help you answer Problems C1-C5.



Try It Online!

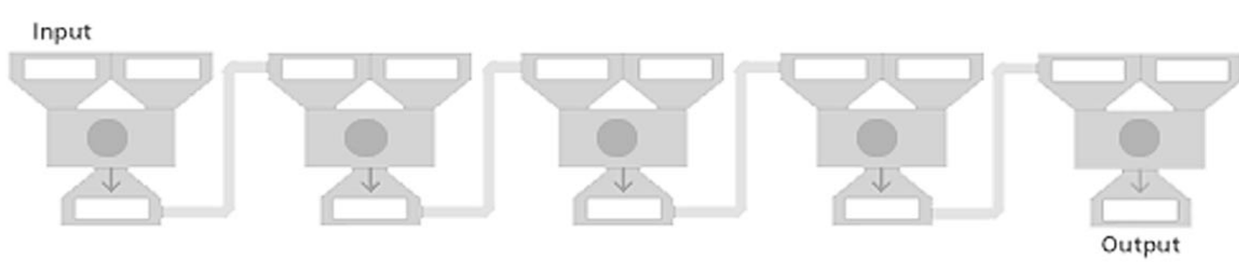
This problem can be explored online as an Interactive Activity. Go to the ***Patterns, Functions, and Algebra* Web site** at www.learner.org/learningmath and find Session 3, Part C, Running a Function Machine.

Problem C1. If the input to Algorithm C is 3, what is the output?

Problem C2. If the output to Algorithm C is 88, can you determine what the input was? How did you do it?

Problem C3. Build Algorithm D, which undoes Algorithm C, below. How would this help with Problem C2?

Algorithm D



Part C, cont'd.

Problem C4. Imagine that Algorithm C and Algorithm D are connected to form one large network. What would happen to a number used as an input in this network? Think about it for a few minutes, and try a few numbers if you need to.

Connecting the two networks gets you back to your original number. Going through all these steps turns out to be a complicated way of doing nothing to a number!

Take It Further

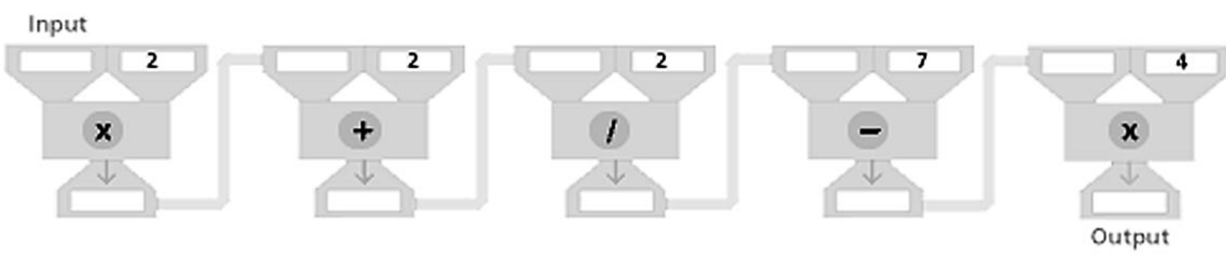
Problem C5. Use the function machine network and create your own algorithm to produce the following outputs:

- 59
- 216
- 15,625
- 7,280
- 0.12345

As an additional challenge, use the fewest possible machines to do so.

Function Machines and Undoing

Think back for a minute to Algorithm A **(SEE PAGE 63)**, which we worked with in Part B. Take a look at the following picture of Algorithm A and answer Problems C6-C10 below.



Problem C6. Imagine dropping a 7 into this network. What comes out the bottom?

Problem C7. Imagine that an 8 came out the bottom. Pull it back through the network and figure out what had to go in the top. **[SEE TIP C7, PAGE 81]**

Problem C8. Imagine that a 100 came out the bottom. Pull it back through the network and figure out what had to go in the top.

Problem C9. Draw a picture using machines to show Algorithm B, the algorithm that "undoes" Algorithm A.

Part C, cont'd.

Problem C10. Imagine connecting the output spout of Algorithm A to the input hopper of Algorithm B. Now you have a huge network. What does it do to a number? **[SEE TIP C10, PAGE 81]**

VIDEO SEGMENT (approximate times: 13:01-13:42): You can find this segment on the session video approximately 13 minutes and 1 second after the Annenberg/CPB logo. Zero the counter on your VCR clock when you see the Annenberg/CPB logo.

In this segment, Deanna and Lolita present the combined network built by connecting Algorithms A and B. Watch the segment after you have completed Problem C10. If you get stuck on the problem, you can watch the video segment to help you.

Do you think every network built by function machines can be "undone"? Suppose a network's steps could all be reversed, step by step. Could every algorithm of this type be "undone"?

Write and Reflect

Problem C11. Suppose someone hands you an algorithm. Describe a general process that will allow you to construct a new algorithm that undoes the one you are given. Can you imagine an algorithm for which your method doesn't work?

Part D: Number Games (35 MINUTES)

[SEE NOTE 5]

Here's a number game:

- Pick a number
- Add 3
- Double your answer
- Subtract 4
- Finally, multiply by 3

Problem D1. Draw a picture of this algorithm and use your picture to answer the following questions:

a. If you pick 3, what will your final number be?

b. If you pick 5, what will your final number be?

c. If you pick 12, what will your final number be?

d. If you pick 38, what will your final number be?

e. If you pick *n*, what will your final number be? **[SEE TIP D1(E), PAGE 81]**

f. If Mary's final result is 48, what number did she pick? **[SEE TIP D1(F), PAGE 81]**

g. If Joe's final result is 30, what number did he pick?

Problem D2. Build an algorithm that lets you find the number someone picked in Problem D1 if you know his or her final result. Try it out.

NOTE **5. Groups:** Work on Problems D1-D7 and discuss what everyone finds.

The problem with the algorithm in Problem D3 is the step "add your original number." It's not clear how to model that with machines. Even if we are able to adapt the machine metaphor to work, we can't "undo" that step of the process.

Consider going through a symbolic representation of the algorithm to see that it does not share this weakness with the machine, because you can combine terms, essentially creating an equivalent algorithm that you can undo:

$[(n + 3) \times 2 + n - 4] \times 3 = (3n + 6 - 4) \times 3$ This is just one of many equivalent algorithms.

Part D, cont'd.

Problem D3. Here's another number game. Use the algorithm to answer the questions below.

- Pick a number
- Add 3
- Double your answer
- Add your original number
- Subtract 4
- Finally, multiply by 3

a. If you pick 3, what will your final number be?

b. If you pick 5, what will your final number be?

c. If you pick 12, what will your final number be?

d. If you pick 38, what will your final number be?

e. If you pick *n*, what will your final number be? **[SEE TIP D3(E), PAGE 81]**

f. If Mary's final result is 48, what number did she pick? **[SEE TIP D3(F), PAGE 81]**

g. If Joe's final result is 30, what number did he pick?

Problem D4. Try to draw a picture of the algorithm in Problem D3. How is it different from the algorithm in Problem D1?

Problem D5. Here's another number game. Use the algorithm to answer the questions below.

- Pick a number
- Multiply by 3
- Add 5
- Multiply by 2
- Subtract 6 times your original number
- Add 4
- Finally, divide by 2

a. If you pick 3, what will your final number be?

b. If you pick 5, what will your final number be?

c. If you pick 12, what will your final number be?

d. If you pick 38, what will your final number be?

e. If you pick *n*, what will your final number be? Can you explain why?

f. If Nancy's final result is 7, what number did she pick?

Write and Reflect

Problem D6. What strategies did you use on Problems D3(f) and D3(g)? Why are these different from the strategies used in Problems D1(f) and D1(g)?

Problem D7. Describe another number game that has the same kind of outputs as the game of Problem D5.

Part E: Other Kinds of Functions

(45 MINUTES)

Functions and Non-Functions

So far you have been thinking about functions as algorithms or machines. They take an input—in the cases you have seen, a number—and give an output. **[SEE NOTE 6]**

A function is really any relationship between an input variable and an output variable in which there is exactly one output for each input. Not all functions have to work on numbers, nor do functions need to follow a computational algorithm. Below are some examples of functions and non-functions. Read through them, then answer Problems E1-E4. **[SEE NOTE 7]**

The following relationships are functions.

Input: an integer
Output: classification of the input as even or odd

Input: a person's Social Security number
Output: that person's birth date

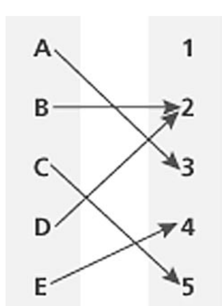
Input: the name of a state
Output: that state's capital

Input: the side length of a square
Output: the area of that square

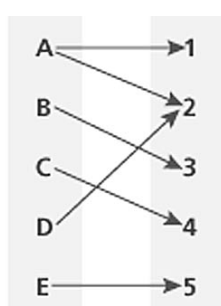
Input: a word
Output: the first letter of that word

NOTE 6. The final part of this session introduces a more general notation of function, rather than just algorithmic functions with numeric inputs and outputs.

Groups: Work in pairs to describe what you think a function is. Some people may recall struggling with learning or teaching about functions using diagrams like these:



A function



Not a function

NOTE 7. Read the definition of a function and take a look at the examples in the course text.

Groups: Work on Problems E1-E4 in small groups, then as a whole group. Discuss the other examples of functions and non-functions before moving on.

Part E, cont'd.

Problem E1. For each function described in the previous list, make a table of five or six input/output pairs. Explain why for every possible input there is only one possible output.

Problem E2. In any of your tables, do you have repeated outputs? That is, do you have two different inputs that give the same output?

The following relationships are *not* functions.

Input: a number
Output: some number less than the input

Input: a whole number
Output: a factor of the input

Input: a person
Output: the name of that person's grandparent

Input: a city name
Output: the state in which that city can be found

Input: the side length of a rectangle
Output: the area of that rectangle

Input: a word
Output: that word with the letters rearranged

Part E, cont'd.

Problem E3. For each relationship described in the previous list, make a table of five or six input/output pairs. Explain why for some inputs there may be more than one possible output. **[SEE TIP E3, PAGE 81]**

Problem E4. Come up with three more examples of relationships that are functions, and three examples of relationships that are not functions. For each relationship, explain why it is or is not a function.

Problems in Part E are taken from IMPACT Mathematics Course 3, *developed by Education Development Center, Inc. (New York: Glencoe/McGraw-Hill, 2000).*

Part E, cont'd.

More Functions

The next function we will explore is called the "Prime?" function. Most of you will remember that a prime number is a whole number that has only itself and 1 as factors. A few examples of prime numbers are 7, 13, and 29. Can you come up with some other examples? **[SEE NOTE 8] [SEE TIP "PRIME?", PAGE 81]**

The "Prime?" function takes positive whole numbers as inputs and produces the outputs *yes* and *no*—*yes* if the input is a prime, and *no* if the input is not a prime. Use what you know about functions and prime numbers to answer Problems E5-E11.

Problem E5.

a. If the input is 3, what is the output?

b. If the input is 2, what is the output?

c. If the input is 100, what is the output?

d. If the input is 1, what is the output? **[SEE TIP E5, PAGE 81]**

Problem E6. If the output of the "Prime?" function is *yes,* what could the input have been? **[SEE TIP E6, PAGE 81]**

Write and Reflect

Problem E7. Explain why "Prime?" is a function.

Problem E8. If possible, describe a function that would undo the "Prime?" function. That is, if you put an input into the "Prime?" function and then put the output into your new function, you get back your original input. **[SEE TIP E8, PAGE 81]**

NOTE 8. Read about the "Prime?" function and review the definition of a prime. Think about some examples of primes and non-primes and how you could test to see if a number is prime if you aren't sure. Work on Problems E5-E8. These problems address common confusion about both prime numbers and functions. **Groups:** Summarize these problems in a discussion as everyone completes their work.

Here are some points to consider:

- 2 is a prime number. It is the only even prime.
- 1 is not a prime. This is a convention. The number 1 fits the definition of prime we have given, since it is only divisible by itself—1—and 1. However, an important theorem in mathematics, called the fundamental theorem of arithmetic, says that every integer greater than 1 is either prime or can be expressed as a *unique* product of prime numbers. If 1 is considered a prime, this would no longer be the case. Consider: $10 = 2 \times 5$. But $10 = 1 \times 2 \times 5$. But $10 = 1 \times 1 \times 1 \times 1 \times 1 \times 2 \times 5$.

  The fundamental theorem of arithmetic is essential for proving many mathematical results, so it would never do to allow 1 to be a prime!
- Two inputs to a function may give the same output. In this case, many numbers produce the output "yes," and many will produce the output "no."
- Not every function can be undone. In this case, if the output is "yes," for example, there's no way of knowing what the input was. (You may want to discuss how this is related to the point above.)

Part E, cont'd.

Problem E9. The "3" function takes real numbers as inputs and always outputs the number 3. **[SEE NOTE 9]**

a. If the input is 17, what is the output?

b. If the input is -2, what is the output?

c. If the input is 1.5, what is the output? **[SEE TIP E9, PAGE 81]**

Problem E10. If the output is 3, what could the input have been?

Write and Reflect

Problem E11. Explain why "3" is a function.

Problem E12. If possible, describe a function that would undo the "3" function. That is, if you put an input into the "3" function and then put the output into your new function, you get back your original input. **[SEE TIP E12, PAGE 81]**

A Geometric Function

Take It Further

Sometimes, functions can be based on an algorithm but still not use numbers as inputs. See Algorithm M on the following page. **[SEE NOTE 10]**

NOTE 9. Read about the "3" function and work through Problems E9-E12. These problems may reinforce many of the points in Note 8.

NOTE 10. Look at how Algorithm M works by going through the steps with a pentagon. After finishing the drawing, consider if there was any other way you could have followed the directions. For example, you could connect the midpoints in a different order. No matter how you connect the midpoints, however, the output will be the same. Once this is clear, work on Problems E13-E16. **Groups:** If there is time, compare results of this geometric algorithm.

There are several surprising things that some people may notice:

- No matter what shape triangle you start with, you end up with 4 identical triangles inside your original triangle. Three are oriented the same way as the original triangle, and one is upside down.

* The 4 triangles are all similar to the original. For example, if you connect the midpoints of a right triangle, you will end up with right triangles inside.

- The areas of each of the triangles are 1/4 the area of the original (since there 4 of them, and they are identical).

- The inside figure of the quadrilateral is a parallelogram. That is, opposite sides are parallel. It doesn't necessarily resemble the outer figure at all.

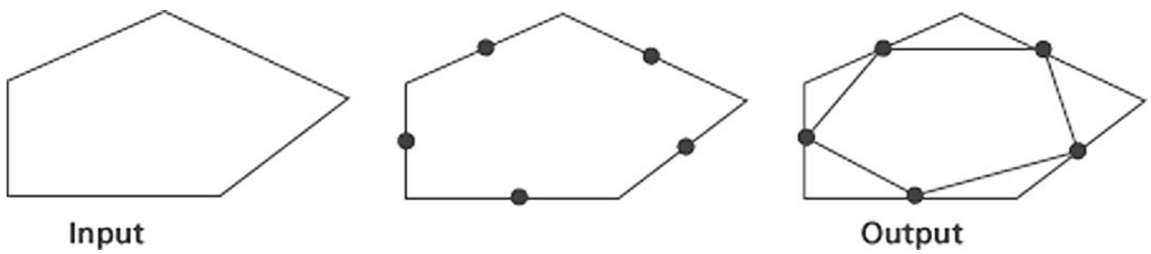
- The inside figure of a quadrilateral contains 1/2 the original area (this may be more difficult to see).

Part E, cont'd.

Algorithm M

- Start with a polygon
- Find the midpoint of each side of the polygon
- Connect each midpoint to the two midpoints on either side of it

Here's what the algorithm does to a pentagon:



Problem E13. Try Algorithm M on three different triangles. Describe in words how the output is related to the input. **[SEE TIP E13, PAGE 81]**

Problems in Part E are taken from IMPACT Mathematics Course 3, *developed by Education Development Center, Inc. (New York: Glencoe/McGraw-Hill, 2000).*

Part E, cont'd.

Problem E14. How does any new triangle created by Algorithm M relate to the original, in size and in shape?

Problem E15. Try Algorithm M on several different quadrilaterals. Describe anything you notice about the outputs. **[SEE TIP E15, PAGE 82]**

Problem E16. Does Algorithm M describe a function? Explain how you know.

Homework

Problem H1. Invent a number game like the one below from Problem D1. Draw a network that lets you figure out the output for any number. Draw another network that lets you find someone's original number if you know his or her final result. Try out your game on someone else.

Here's a number game:

- Pick a number
- Add 3
- Double your answer
- Subtract 4
- Finally, multiply by 3

Problem H2. Tell whether each example below is a function, and explain how you decided.

a. Input: a circle. Output: the ratio of the circumference to the diameter.

b. Input: a soccer team. Output: a member of the team.

c. Input: a CD. Output: a song on the CD. **[SEE TIP H2, PAGE 82]**

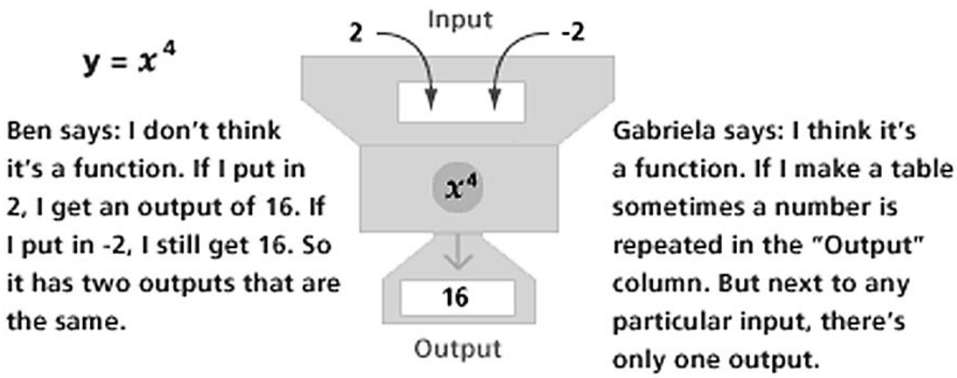
Problems H2 and H3 are taken from IMPACT Mathematics Course 3, *developed by Education Development Center, Inc. (New York: Glencoe/McGraw-Hill, 2000).*

Homework, cont'd.

Problem H3. Gabriela and Ben are trying to decide whether the rule $y = x^4$ is a function. Represented as an algorithm, the equation $y = x^4$ is equivalent to starting with a number, then multiplying that number by itself 4 times. For example:

Input: $x = 2$

Output: $y = 2 \times 2 \times 2 \times 2 = 16$



Input	Output
0	0
1	1
2	16
3	81
2/5	16/625
-1	1
-2	16

Who is correct, Ben or Gabriela? Is $y = x^4$ a function? Explain how you know.

Problems H4-H7 involve *iteration,* a process that can be done to any function where the output is the same type as the input. When you iterate a function, you apply it again and again, each time using the previous output as the new input. Iteration is a very important technique for solving equations approximately when typical algebraic methods can't be used, and it is also used to model many real-world problems like population growth and the change of weather. **[SEE TIP H4-H7, PAGE 82]**

Here's an example:

Input: a real number
Output: half that number

Start with an input of 20. The first output is 10. Now, use that output as the new input. The second output is 5. Use 5 as the next input, and you get an output of 2.5. And so on.

Problem H4. Use the function described above, starting with an input of -16. What are the first 5 outputs as you iterate the function? What will happen to the value if the iteration continues forever? **[SEE TIP H4, PAGE 82]**

Homework, cont'd.

Problem H5. Build a network of function machines that adds 1 to an input and then divides by 3. Now iterate the function. Try three different original inputs, and iterate the function at least 10 times for each input. What is going on? **[SEE TIP H5, PAGE 82]**

Take It Further

A "fixed point" of an iteration is a value of the input that produces itself as an output. For example, if your algorithm subtracts 3 from an input, then multiplies by 2, the input value 6 has output 6, so it is a fixed point.

Problem H6. Find any fixed points for these algorithms. There may not be any, or there may be more than one.

a. Add 1, and divide by 3

b. Subtract 6, and multiply by 3

c. Add 6

d. Square the input

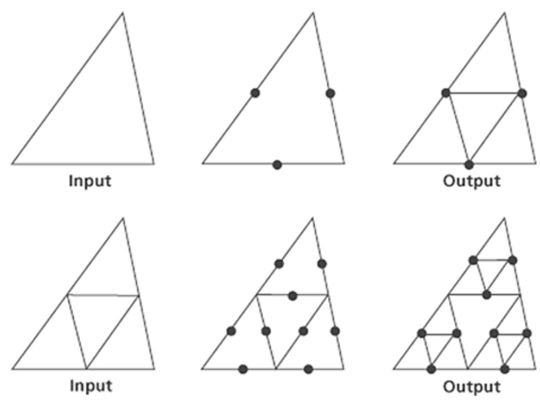
e. Add 12, and divide by 4

f. Square the input, then subtract 6 **[SEE TIP H6, PAGE 82]**

You can also iterate functions that act on geometric shapes. Here's a new function:

- Start with a rightside-up triangle; that is, a triangle with one base horizontal.
- Find the midpoints of each side of the triangle, and connect them to each other.
- Repeat this process on each rightside-up triangle in the output.

To the right is the start of the iteration for this function:



Problem H7. Draw your own triangle. Iterate the function above at least 3 times. Describe how the outputs relate to each other.

Tips

Part A: Doing and Undoing

TIP A1. See if you can find two different inputs whose outputs are the same. How would that make it impossible to find the original number?

Part B: Undoing Algorithms

TIP B1. Try to run the algorithm's steps using the variable n rather than a particular value. After the first step, the result is $2n$. After the second step, the result is $2n + 2$.

Part C: Function Machines

TIP C7. Try to think of an efficient way of solving this problem. You can also use the Interactive Activity on the course Web site to model Algorithm A. Go to the *Patterns, Functions, and Algebra* Web site at www.learner.org/learningmath and find Session 3, Part C, Running a Function Machine.

TIP C10. Problem C10 is a lot like Problem C4.

Part D: Number Games

TIP D1(E). After the first step, the result is $n + 3$.

TIP D1(F). You will have to undo the algorithm for this and for D1(g).

TIP D3(E). What would you use to represent "your original number"?

TIP D3(F). This is more difficult than the seemingly identical question in Problem D1. Why? What can be done about it? Does your work in D3(e) help?

Part E: Other Kinds of Functions

TIP E3. Be sure to generate pairs of inputs and outputs that show that the relationship is not a function. What property would those pairs have?

TIP "PRIME?". The number 6 is *not* a prime number, because it has 2 and 3 as factors. The number 11 *is* prime, because its only factors are 1 and itself. A prime number must have *exactly* 2 factors—no more, no less.

TIP E5. Remember, the output is either *yes* or *no*.

TIP E6. How many answers are there?

TIP E8. Note the relationship between Problems E6 and E8.

TIP E9. The answers to all the parts of Problem E9 are all the same number.

TIP E12. Note the relationship between Problems E10 and E12.

TIP E13. Be sure to select triangles that are different in a significant way: acute, obtuse, scalene, isosceles.

Tips, cont'd.

TIP E15. As with Problem E13, select quadrilaterals that are different in a significant way. You might also concentrate on a specific type of special quadrilateral to determine if Algorithm M does something similar to all quadrilaterals of that type.

Homework

TIP H2. Remember, a function is any relationship in which each input leads to exactly one output, and the same output may be repeated more than once for different inputs. A rule is *not* a function when the same input can lead to multiple outputs.

TIP H4-H7. Pick a number. Add 1. Add 1 again. Again. Again. By doing this, you're iterating the "add 1" function. The "his father before him" function is frequently iterated in conversation.

TIP H4. The first output is -8. Then use -8 as the next input.

TIP H5. What happens to the values "in the end," after many iterations? For what numbers does this happen? Can you explain why?

TIP H6. If a fixed point's input is *n*, what would its output have to be?

Solutions

Part A: Doing and Undoing

Problem A1. The inputs 12 and 8 each lead to the output 4. If you only knew that the output was 4, it would be impossible to determine which of 12 and 8 was the correct input.

Problem A2. Some examples:

- Driving directions. Telling someone how to get somewhere usually allows them to figure out how to get back (if there are no one-way streets involved). In particular, to drive back you must take each road the opposite direction, and in reverse order (last road first).
- Packing and unpacking. Especially with commercially shipped packages, it can be difficult to re-pack a box without knowing where things were located before you unpacked it.
- In mathematics, many algebra problems involve "undoing" steps. For example, if $3x = 12$, you can find x by undoing the multiplication step.

Problem A3. Lots of things can't be undone easily, like throwing a water balloon, using gasoline in a car engine, or exploding fireworks.

Part B: Undoing Algorithms

Problem B1.

a. The output is 12.

b. The output is 16.

c. For an input n, the algorithm performs n >> $2n$ >> $2n + 2$ >> $n + 1$ >> $n - 6$ >> $4n - 24$.

d. Knowing that the output is $4n - 24$ means that we have to solve $4n - 24 = 28$, an equation that is solved by $n = 13$. This means that the input must be 13.

e. The input is 14.

f. The input is 18.

g. The input is 15.

Problem B2. Working backwards from the end of the algorithm to the beginning will undo it. There are other strategies, such as writing equations for each of (d) through (g).

Problem B3. Algorithm B "undoes" everything that Algorithm A does.

- Take the input (the output of Algorithm A) and
- Divide it by 4
- Add 7
- Double it
- Subtract 2
- Halve that
- There's your output (of Algorithm B)

Solutions, cont'd.

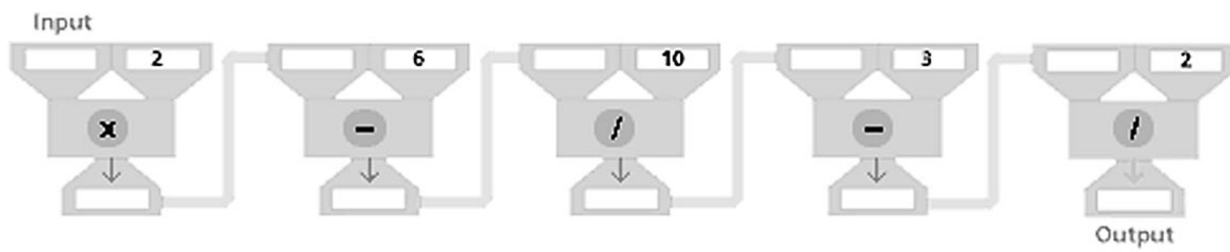
Problem B4. Test it out. See if you can figure out why it works by thinking of the algorithms as driving directions. Where would you end up if someone gave you directions like Algorithms A and B, and asked you to do them both? In short, yes, they undo each other.

Part C: Function Machines

Problem C1. The number 48 will come out at the bottom.

Problem C2. Try to undo the steps. The original number was 7.

Problem C3. Sketch. The order of operations is: multiply by 2, subtract 6, divide by 10, subtract 3, divide by 2. Algorithm D is the inverse of Algorithm C, so using 88 as the input for Algorithm D would answer Problem C2.



Problem C4. The huge Algorithm CD doesn't do anything; its output numbers will equal its input numbers.

Problem C5. All of them are possible in multiple ways.

$59 = 10 \times 5 + 9$

$216 = 6 \times 6 \times 6$

$15{,}625 = 5 \times 5 \times 5 \times 5 \times 5 \times 5$

$7{,}280 = 9 \times 9 \times 9 \times 10 - 10$

$0.12345 = [(1 / 9) / 9] \times 10$

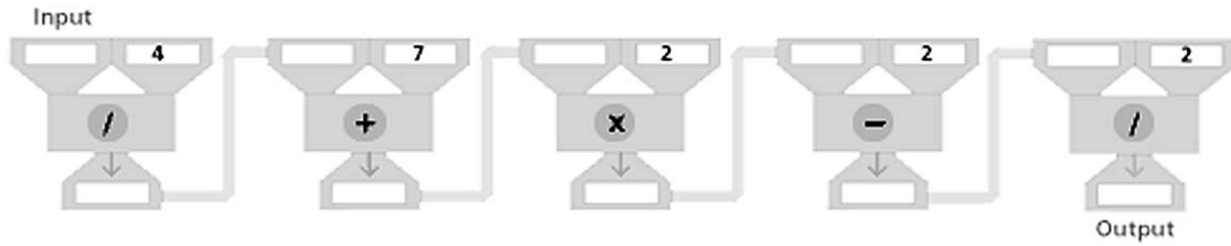
Problem C6. The output will be 4.

Problem C7. Use Algorithm B, which undoes Algorithm A. The input was 8.

Problem C8. Using Algorithm B, the input was 31.

Solutions, cont'd.

Problem C9.



Problem C10. It will leave the number unchanged, since A and B undo each other.

Problem C11. If you can list the algorithm as a series of steps involving unchanging numerical operations (like "add 6"), then they can be undone by an algorithm which performs the inverse operation, and where the steps are performed in reverse order. Unfortunately, some operations do not have inverses, like squaring or throwing a water balloon. Think of Mr. Lewis's rule from Session 2 (see page 50)—this is a rule that cannot be undone.

Part D: Number Games

Problem D1.

a. If you pick 3, the final number will be 24.

b. Picking 5 gives a final number of 36.

c. Picking 12 gives 78.

d. Picking 38 gives 234.

e. Picking n gives $6n + 6$ [n >> $n + 3$ >> $2n + 6$ >> $2n + 2$ >> $6n + 6$].

f. Mary's original number is 7. Solving $6n + 6 = 48$ gives this, or you could run 48 through the algorithm in reverse.

g. Joe started with 4.

Problem D2. This algorithm would undo the steps of the original by using inverse operations taken in reverse order. That would be:

- Pick a number
- Divide it by 3
- Add 4
- Cut it in half
- Subtract 3

You should find that this algorithm undoes the one in Problem D1. Inverse algorithms like this are frequently incorporated in mathematical magic tricks.

Solutions, cont'd.

Problem D3.

a. The final number will be 33.

b. The final number will be 51.

c. The final number will be 114.

d. The final number will be 348.

e. The final number will be $9n + 6$ [$n >> n + 3 >> 2n + 6 >> 3n + 6 >> 3n + 2 >> 9n + 6$].

f. This is difficult without completing D3(e), but solving $9n + 6 = 48$ suggests that Mary's original number was $4\ ^2/_3$.

g. Solving $9n + 6 = 30$ shows that Joe's original number was $2\ ^2/_3$.

Problem D4. It's pretty tough to draw a picture of the machine for "add the original number"! Specifically, the original number is a variable, so the results of that machine vary depending on what you put in to start. A machine labeled "+ *n*" might work, but notice how different that is from one that reads "+ 3".

Problem D5.

a. The final number will be 7.

b. The final number will be 7.

c. The final number will be 7.

d. The final number will be 7.

e. The final number will always be 7. The algorithm, starting with n, proceeds as $n >> 3n >> 3n + 5 >> 6n + 10 >> 10 >> 14 >> 7$. Note the step "subtract 6 times the original number" removes n completely, so the answer does not depend on n.

f. Since the answer is always 7, there is no way to determine what number Nancy was thinking of.

Problem D6. You almost need to know a formula for the entire algorithm before solving D3(f) and D3(g). In Problem D1, a "reverse" strategy will work, taking the algorithm step by step in reverse. But in Problem D3, the "add the original number" step is impossible unless you know the original number! So you could guess until you get it, or come up with a formula for the entire algorithm.

Problem D7. Here's one: Pick a number, multiply it by 9, add 18, divide by 3, add 6, divide by 3 again, then subtract the original number. The answer will always be 4.

Solutions, cont'd.

Part E: Other Kinds of Functions

Problem E1. Here are some possible completions for the tables.

Integer	1	2	3	4	5	10	15
Odd or Even?	odd	even	odd	even	odd	even	odd

SSN	590-14-6017	024-33-3467	024-33-3568	024-33-7146	036-89-0831
DOB	6/2/75	10/27/70	10/27/70	8/10/74	6/8/84

State	Massachusetts	Texas	Washington	North Dakota	West Virginia
Capital	Boston	Austin	Olympia	Bismarck	Charlestown

Side Length	1	2	3	4	5	10	15
Area	1	4	9	16	25	100	225

Word	Word	Hey	Wow	Math	Is	Very	Cool
First Letter	W	H	W	M	I	V	C

In each case there are clear reasons that there can only be 1 answer. For example, a state can have only 1 capital city. A word can only have 1 first letter.

Problem E2. Sure, but not always. The odd-or-even, date of birth, and letter functions have the possibility of matching outputs.

Problem E3. More tables!

Number	10	10	15	17	21	21	0
Smaller Number	7	8	10	12	12	-5	-100

Number	15	20	24	24	30	45	100
Factor	3	5	3	4	10	9	20

Person	Abbey	Abbey	Megan	Megan	Brian
Grandparent	Mary	John	Mary	Alice	Henry

City Name	New York	Chicago	Salem	Salem	Portland	Portland
State Name	New York	Illinois	Massachusetts	Oregon	Oregon	Maine

Side Length	5	10	20	1	5	10	100
Area	20	20	20	1/4	15	50	250,000

Word	ear	ear	mare	toilets	relation	listen	Elvis
Anagram	are	era	ream	T. S. Eliot	oriental	silent	lives

For certain (not necessarily all!) inputs, there can be more than 1 correct output. Note how different this is from Algorithms A and B.

Solutions, cont'd.

Problem E4. Other functions: a circle's circumference is a function of its radius; the average temperature is a function of the time of year; a TV program's rating is a function of the number of people watching the show. For each function, there can only be 1 output for a given input, while a non-function may have more than 1 output for the same input. For example, people of more than 1 age can wear size 11 shoes.

Problem E5.

a. The output is yes, 3 is a prime number.

b. The output is yes, 2 is a prime number.

c. No, 100 is not a prime (it has lots of factors).

d. No, 1 is not a prime (it needs to have exactly 2 factors).

Problem E6. It could be any prime number: 2, 3, 5, 7, 11, 13, 17, 19, ...

Problem E7. It's a function because there is exactly 1 output. The answer is always *yes* or *no,* never both.

Problem E8. There is no such function. The outputs are only *yes* or *no,* so if such a function existed, it would have to guarantee the specific prime number picked from *yes,* which is impossible. Put another way, the undoing rule cannot be a function, because *yes* would return *all* the prime numbers, and *no* would return *all* the non-primes.

Problem E9.

a. The output is 3.

b. The output is 3.

c. The output is still 3.

Problem E10. It could be any number at all. Since the output is always 3, telling us that the output is 3 doesn't give any new information. This is the same situation as Problem D5.

Problem E11. There is exactly 1 value for the output. It's always 3, but that doesn't keep it from being a function.

Problem E12. No such function exists.

Problem E13. The output is a triangle whose sides are 1/2 the sides of the original, and parallel to the original sides.

Problem E14. All the sides are 1/2 as long, and the new triangle's area is 1/4 of the original.

Problem E15. All the formed quadrilaterals are parallelograms.

Problem E16. Yes, because there is exactly 1 output polygon for any starting polygon.

Solutions, cont'd.

Homework

Problem H2.

a. Yes, the output is always π (about 3.14), but there is exactly 1 output for any input.

b. No, the same team can have lots of different members.

c. No, the same CD can have many different songs.

Problem H3. Gabriela is correct. The number of matching outputs is not important (witness the "3" function!), only that there is exactly 1 output each time.

Problem H4. The outputs are -8, -4, -2, -1, and -1/2. The iterates get closer and closer to 0.

Problem H5. No matter what you start with, the inputs will get closer and closer to 1/2.

Problem H6. In each case, the output value must equal the input, so writing the rule and making it equal n is one way to solve the problem.

a. The fixed point is $x = 1/2$.

b. The fixed point is $x = 9$.

c. There is no fixed point.

d. There are two fixed points, $x = 0$ and $x = 1$.

e. The fixed point is $x = 4$.

f. There are two fixed points, $x = 3$ and $x = -2$.

Problem H7. Each output is 3 half-sized copies of the input, with a hole in the middle. The pattern can repeat itself indefinitely.

Notes