## ETHIOPIAN MULTIPLICATION 

Sometimes called "Russian Multiplication" or "Peasant Multiplication," the curious multiplication technique we discuss here has close connections to an ancient multiplication technique employed more than three-and-a-half millennia ago in Egypt. That older technique is described in the famous Rhind Papyrus and is sometimes called "Egyptian Multiplication" or even "Ethiopian Multiplication," and we'll describe it too!

```
EXPLODING DOTS Topic:
Experience 2: Understanding the 1\leftarrow2 machine.
Suggested Grade Level:
Middle-school, High School
```


## ETHIOPIAN MULTIPLICATION

Demonstrate the following multiplication technique, often called "Peasant Multiplication," on the classroom board.

Here's a curious way to perform long multiplication.

To illustrate the technique, let's compute $37 \times 10$, whose answer we know.

- Start by heading two columns with the two numbers in the product.
- Repeatedly halve the numbers in the left column and double the numbers in the right - and to make life easy, ignore any fractions that arise. Stop when you see 1 on the left.

- Delete any row that has an even number on the left and sum the numbers that survive on the right.
- The sum of the surviving numbers is the desired product!

Practice: Use this technique to show that $10 \times 37$ also gives the answer 370 .

Have your students practice this technique with some more examples.

Of course, today's question is: Why does this technique work?

A Nudge: You could perhaps ask your students to examine the products $8 \times N$ and $64 \times N$ via this technique where $N$ represents some general (unspecified) number.

## Some Things Students Might Notice or Ask

1. This is weird!
2. Is it really okay to ignore remainders?
3. Working out $10 \times 37$ is awkward.
4. You can work out $10 \times 37$ in an Exploding Dots way.

5. To work out $8 \times N$ we end up doubling $N$ three times to get $8 N$, and only this survives on the right side. To work out $64 \times N$ we end up double $N$ six times to get $64 N$, and only this survives on the right side.

6. You don't actually have to stop at 1 on the left side: you can keep going to get zeros. But since zero is an even number, these rows get crossed out in any case.


## SOLVING THE PUZZLE

## See EXPERIENCE 2 of EXPLODING DOTS: Understanding the $1 \leftarrow 2$ machine.

Scholars of the region on ancient Egypt some 3500 years ago were aware that it is not too taxing mentally to repeatedly double a number, and from this, one can perform multiplication. For example, to compute 37 times a number $N$ start by listing the numbers that arise from repeatedly doubling $N$.

and since $37=32+4+1$ we can compute $37 \times N$ simply by adding $32 N+4 N+N$.
For example,

$$
\begin{aligned}
37 \times 10 & =(32+4+1) \times 10 \\
& =32 \times 10+4 \times 10+1 \times 10=370
\end{aligned}
$$

This ancient technique of adding repeated doubles of a given number to compute a product is sometimes called today "Egyptian Multiplication." It feels closely related to the multiplication technique described in today's puzzle.

To work out a product $M \times N$, Egyptian multiplication relies on writing $M$ as a sum of powers of two. That is-in Exploding Dots thinking-it relies on knowing the $1 \leftarrow 2$ machine code for $M$.


And how do we find the $1 \leftarrow 2$ code for a number $M$ ? Well, we just put in $M$ dots in the right box of a $1 \leftarrow 2$ machine!

Here's the thing to note: In a $1 \leftarrow 2$ machine, if the count of dots in a box is

EVEN, then all the dots explode leaving 0 behind and half the number of dots appear in the box one place to the left,

ODD, then all but one of the dots explode leaving 1 behind and just under half the number of dots appear in the box one place to the left.


That is, in working out the $1 \leftarrow 2$ machine code for 37 , say, we place 37 dots in the machine, and repeatedly halve the count on dots as we move to the left (ignoring remainders) and note that a dot remains (the remainder!) at each position that the count was odd.

The positions of the odd counts match the powers of two that appear in the binary code of the number.


And we thus we see that computing $37 \times N$ matches the Peasant Multiplication method presented.


Of course, there is nothing special about the number 37 here.

To compute $M \times N$, repeatedly halve $M$ (ignoring remainders). Focusing on the odd entries gives that powers of two that appear when writing $M$ as a sum of powers of two. Thus $M \times N$ is the sum of these powers of two each multiplied by $N$. We see these multiples of two multiplied by $N$ when we repeatedly double $N$.

## EXTENSIONS

Every solved problem, of course, is an invitation to explore and play more. Might your students enjoy these explorations?

Wild Exploration 1: We have a quick method for finding the binary code of number.

Write the number to the right of the page and, working left, repeatedly halve the number ignoring remainders. Writing a 1 under every odd entry and 0 under every even entry gives the binary code of the number.

$$
\begin{aligned}
& 1 \leftarrow 3 \leftarrow 6 \leftarrow 13 \leftarrow 26 \\
& 1 \quad 1 \begin{array}{llll}
1 & 0 & 1 & 0
\end{array} \\
& \text { The binary code of } 26 \text { is } 11010 .
\end{aligned}
$$

Devise a similar method for swiftly finding the base-three code of a number. Can you create new multiplication technique based on these base-three codes?

Wild Exploration 2: Consider a $-1 \leftarrow 2$ machine in which two dots in a box explode to be replaced by an antidot one place to their left, and two antidots in a box explode to be replaced by a dot one place to their left. This is a base negative-two machine.


This machine has the added feature that any antidot can be replaced by a dot along with a second dot one place to its left.

a) Put ten dots in the machine and show you obtain the code $1|1| 1|1| 0$ for it.
b) What is the code for negative ten in this machine using only 0 s and 1 s ?

One can prove that every positive integer has a unique code in this machine using only the digits 0 and 1. (Care to prove this?) These codes are called negabinary codes.
c) Can you find a swift way to compute the negabinary code of a number along the lines of this puzzle? If so, can you create an interesting multiplication method from it?

