

Utah State University

DigitalCommons@USU

Instructional resources

Integrating Elementary-Level Mathematics
Curricula with Expansively-Framed Computer
Science Instruction

9-2023

Cache Code Math Computer Lab Lesson Plans: Repeated Addition & Multiplication

Jody Clarke-Midura

Utah State University, jody.clarke@usu.edu

Jessica Shumway

Utah State University, jessica.shumway@usu.edu

Kimberly Beck

kimberly.beck@usu.edu

Umar Shehzad

Utah State University, umar.shehzad@usu.edu

Mimi Recker

Utah State University, mimi.recker@usu.edu

Follow this and additional works at: https://digitalcommons.usu.edu/eled_support_instructional



Part of the [Education Commons](#)

Recommended Citation

Clarke-Midura, Jody; Shumway, Jessica; Beck, Kimberly; Shehzad, Umar; and Recker, Mimi, "Cache Code Math Computer Lab Lesson Plans: Repeated Addition & Multiplication" (2023). *Instructional resources*. Paper 10.

https://digitalcommons.usu.edu/eled_support_instructional/10

This Curriculum is brought to you for free and open access by the Integrating Elementary-Level Mathematics Curricula with Expansively-Framed Computer Science Instruction at DigitalCommons@USU. It has been accepted for inclusion in Instructional resources by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



CACHE

CODE

MATH

Repeated Addition & Multiplication

Fall 2023



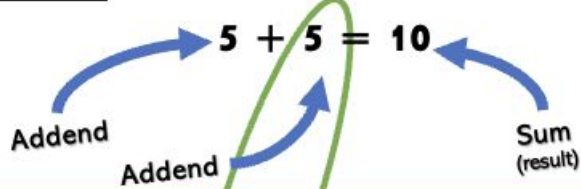
CACHE

CODE

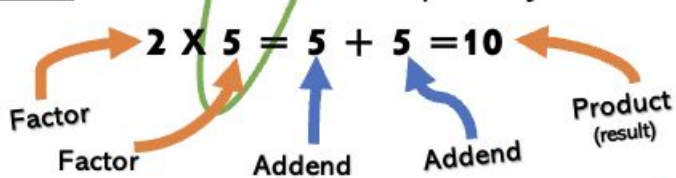
MATH

Math Vocabulary

Addend: a number that is added to another

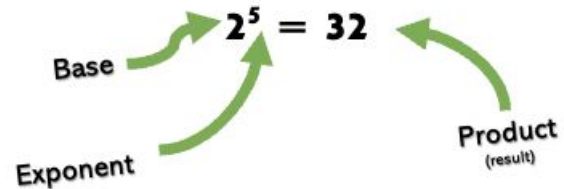


Factor: a number that is multiplied by another



Base: the number that is used as the repeated factor in exponential form

Exponent: the number that tells how many times the base is used as a factor



CACHE

CODE

MATH

Variable Block

VARIABLE

set

VARIABLE ▾

to

Sets the specified variable to the amount entered



when  clicked

set size to 30 %

 erase all

go to x: -230 y: -160

set ADDEND ▾ to 5

In this code, what is the value of the variable, ADDEND?

CACHE

CODE

MATH

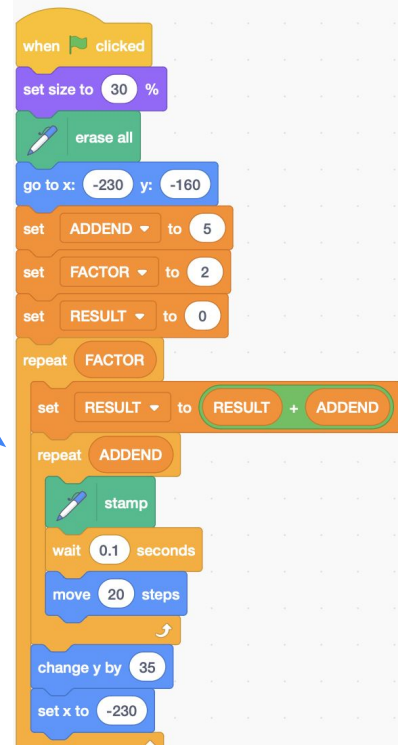
Stamp and Repeat Block



The stamp block is a pen block. It will stamp (copy) the sprite.



The repeat block will loop (or repeat) the code that is put within the block code.



*How many times will the 2nd repeat block loop?
How many stamps will there be?*

CACHE

CODE

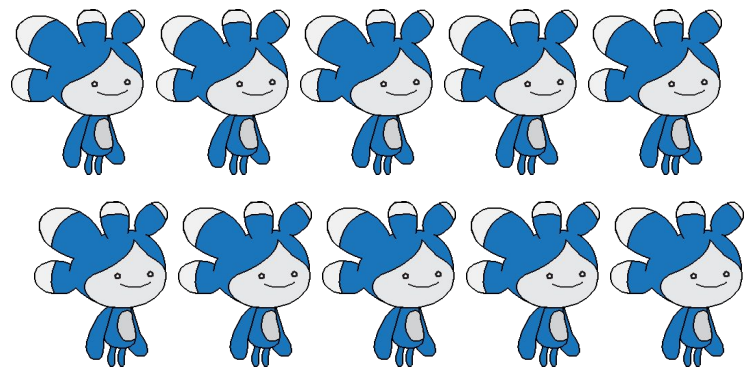
MATH

Repeated Addition

Repeated addition
of addends can be
represented by a
multiplication equation:

$$5 + 5 = 10$$

$$5 \times 2 = 10$$



CACHE

CODE

MATH

Read & Predict

```
when green flag clicked
  set size to 30 %
  erase all
  go to x: -230 y: -160
  set ADDEND to 5
  set FACTOR to 2
  set RESULT to 0
  repeat FACTOR
    set RESULT to RESULT + ADDEND
  repeat ADDEND
    stamp
    wait 0.1 seconds
    move 20 steps
  change y by 35
  set x to -230
```

What do you think this code will do?

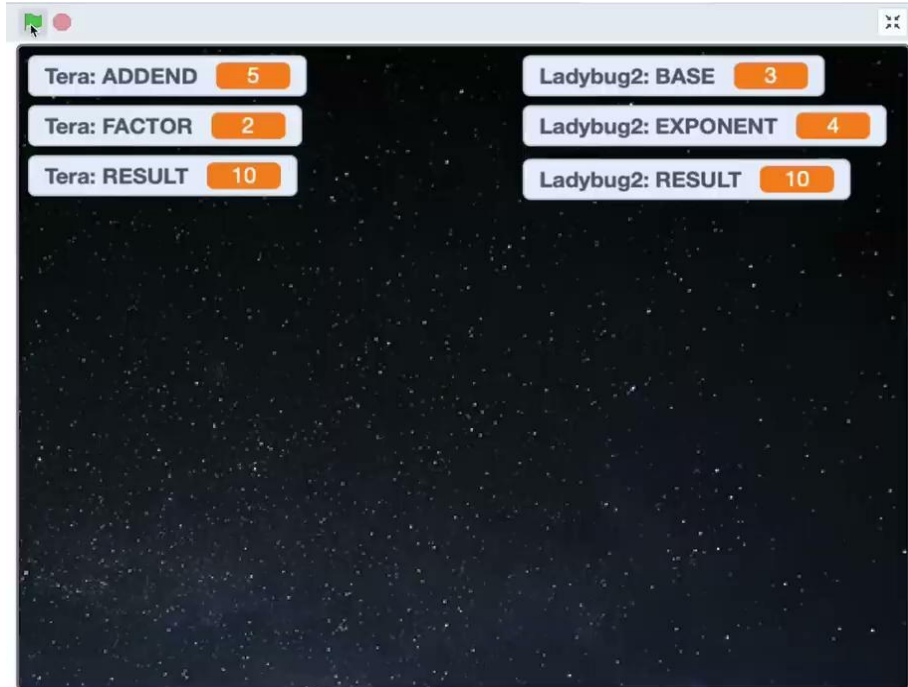
What will be the final value of the variable RESULT?

CACHE

CODE

MATH

Watch the Code



CACHE

CODE

MATH

See the Code

Go to the following URL & click “See Inside”

<https://scratch.mit.edu/projects/873533865>

 See inside

CACHE

CODE

MATH

Select Tera



Click on the
Tera sprite.

A screenshot of the Scratch project editor. The project title is "Repeated Addition & Mult..." by aub_codes. The interface shows a code editor on the left with a script area containing a sequence of blocks: "when clicked" (green flag), "set size to 30%", "erase all", "go to x: -230 y: -160", "set ADDEND to 5", "set FACTOR to 2", "set RESULT to 0", a "repeat" loop for FACTOR times containing "set RESULT to RESULT + ADDEND", "stamp", "wait 0.1 seconds", "move 20 steps", "change y by 35", "set x to -230", "change y by 10", and "set y to -125". The right side shows a stage with a dark starry background and several data monitors: Tera: ADDEND (5), Tera: FACTOR (2), Tera: RESULT (10), Ladybug2: BASE (3), Ladybug2: EXPONENT (4), and Ladybug2: RESULT (10). The bottom right shows the sprite area with "Tera" selected, "Ladybug2", and "Correct C..." as options. The Tera sprite's position is x: -230, y: -125, size: 30, and direction: 90.

CACHE

CODE

MATH

Run the Code



Click the
green flag to
run the code!

A screenshot of the Scratch code editor. The main workspace shows a script for a character named Tera. The script starts with a 'when clicked' event, followed by 'set size to 30 %', 'erase all', 'go to x: -230 y: -20', 'set ADDEND to 7', 'set FACTOR to 5', 'set RESULT to 0', a 'repeat FACTOR' loop containing 'set RESULT to RESULT + ADDEND', another 'repeat ADDEND' loop containing 'stamp', 'wait 0.1 seconds', 'move 20 steps', 'change y by 35', 'set x to -230', and 'change y by -35'. The right-hand side of the editor shows a stage with a black background and several data monitors: Tera: ADDEND (7), Tera: FACTOR (5), Tera: RESULT (35), Ladybug2: BASE (3), Ladybug2: EXPONENT (4), and Ladybug2: RESULT (35). A green flag icon is visible in the top right corner of the workspace, with a black arrow pointing to it from the text 'Click the green flag to run the code!'. The top of the interface shows the Scratch logo, settings, file, edit, and project page options, along with the title 'Repeated Addition & Mult...' by aub_codes.

CACHE

CODE

MATH

Understand the Code

```
when clicked
  set size to 30 %
  erase all
  go to x: -230 y: -160
  set ADDEND to 5
  set FACTOR to 2
  set RESULT to 0
  repeat FACTOR
    set RESULT to RESULT + ADDEND
    repeat ADDEND
      stamp
      wait 0.1 seconds
      move 20 steps
    change y by 35
    set x to -230
```

The sprite's size is set to 30% of its original size. It needs to be small because we are going to "repeat" many of them.

Tera (sprite) goes to the bottom left corner of the screen.

The rows of Tera are repeated FACTOR # of times (2) and the total is stored in the RESULT variable.

The answer is calculated by adding ADDEND to previous value of the RESULT variable.

An image of the sprite is "stamped" (copied) ADDEND # of times.

Tera moves up to the start of a new row.

CACHE

CODE

MATH

Understand the Code

Tera moves to a new row after leaving 5 “stamps” (copies) of itself underneath.

In one row, Tera is stamped **ADDEND** times (variable ADDEND is set to 5).

In math this would be represented as:

$1+1+1+1+1$

5 Teras are displayed in that row.



CACHE

CODE

MATH

Understand the Code

The rows of Tera are repeated **FACTOR** number of times (FACTOR is set to 2).

Total is stored in the **RESULT** variable (10).

$$5 \times 2 = 10$$

$$5 + 5 = 10$$



CACHE

CODE

MATH

Your Turn!

Change Tera's code to model

$$3 \times 4 = 12$$

$$3+3+3+3 = 12$$





CACHE

CODE

MATH

Your Turn!

Change the code to model a multiplication problem of your choice!

Share with your neighbor.

CACHE

CODE

MATH

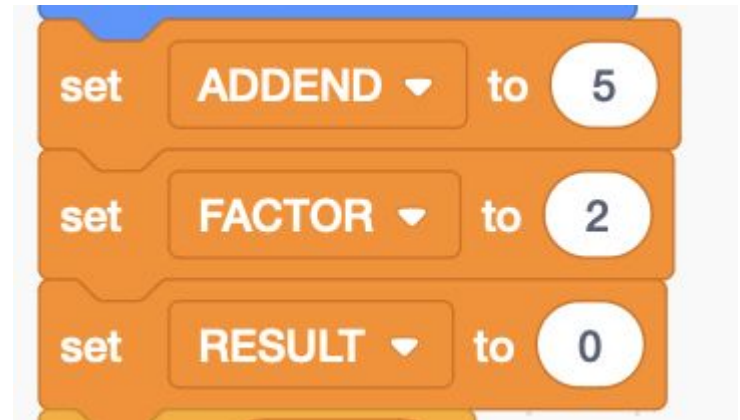
Reset Tera's Code

Click Tera and reset the variables in your Tera code.

Set ADDEND To 5.

Set FACTOR to 2.

Set RESULT to 0.



CACHE

CODE

MATH

Repeated Multiplication

Repeated multiplication of factors can be efficiently represented by **exponent** notation.

$$2^5 = 32$$

$$2 \times 2 \times 2 \times 2 \times 2 = 32$$

CACHE

CODE

MATH

Modifying Tera

We will modify the repeated addition code to program repeated multiplication.

$$2^5$$

$$2 \times 2 \times 2 \times 2 \times 2 = 32$$

CACHE

CODE

MATH

Select Ladybug



Click on the Ladybug2 sprite.

Scratch interface showing a script for a ladybug character. The script includes a 'when space key pressed' event, followed by 'erase all', 'set size to 30%', 'go to x: -230 y: -180', 'set ADDEND to 5', 'set FACTOR to 2', 'set RESULT to 0', a 'repeat FACTOR' loop containing a 'repeat ADDEND' loop with a 'stamp' block, 'wait 0.1 seconds', 'move 20 steps', 'change y by 35', and 'set x to -230'. The right side shows the stage with a starry background and a row of ladybug sprites. The 'Ladybug2' sprite is selected, and its properties are shown below: x: -230, y: -125, size: 30, direction: 90. A black arrow points from the 'Ladybug2' sprite in the stage area to the 'Ladybug2' sprite icon in the separate view on the left.

CACHE

CODE

MATH

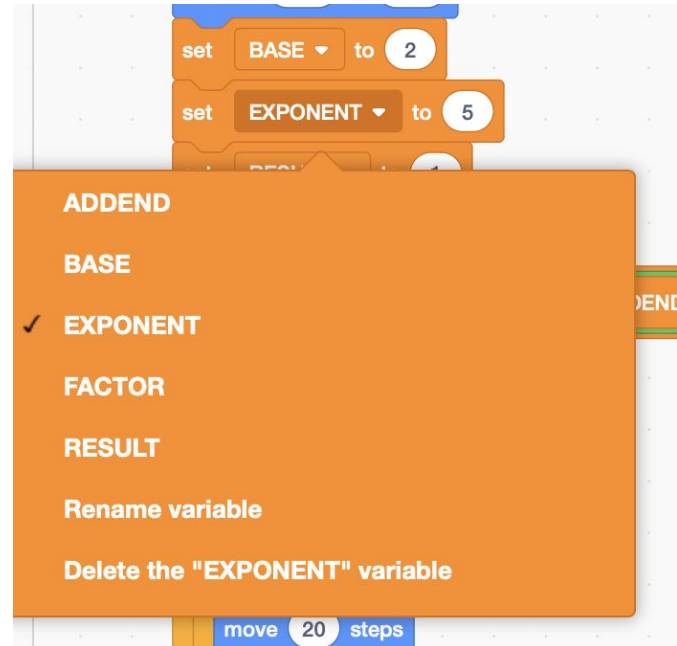
Change Variables

Change ADDEND to **BASE** and set value to **2**.

Change FACTOR to **EXPONENT** and set value to **5**.

Set **RESULT** value to **1**.

Chicks sprite has the correct code for exponents



CACHE

CODE

MATH

Repeat Blocks

In repeat blocks:

Change **FACTOR**

to **EXPONENT**

Change **ADDEND**

to **RESULT**

The image shows a Scratch code editor with a script for a Ladybug2 sprite. The script is as follows:

```
when space key pressed
  erase all
  set size to 30 %
  go to x: -230 y: -160
  set BASE to 2
  set EXPONENT to 5
  set RESULT to 1
  repeat EXPONENT
    set RESULT to RESULT + ADDEND
  repeat RESULT
    stamp
    wait 0.1 seconds
    move 20 steps
    change y by 35
    set x to -230
```

The right panel shows the following variable monitors:

Tera: ADDEND	5	Ladybug2: BASE	3
Tera: FACTOR	2	Ladybug2: EXPONENT	4
Tera: RESULT	10	Ladybug2: RESULT	10

CACHE

CODE

MATH

Operator Block

Remove addition operator block and replace with **multiplication** operator block



Scratch IDE interface showing a script for a ladybug. The script includes blocks for 'when space key pressed', 'erase all', 'set size to 30%', 'go to x: -230 y: -160', 'set BASE to 2', 'set EXPONENT to 5', 'set RESULT to 1', 'repeat EXPONENT times', 'set RESULT to RESULT * BASE', 'repeat RESULT times', 'stamp', 'wait 0.1 seconds', 'move 20 steps', 'change y by 35', and 'set x to -230'. The right side shows the stage with a ladybug sprite and a data monitor for variables: Tera: ADDEND (5), Tera: FACTOR (2), Tera: RESULT (10), Ladybug2: BASE (3), Ladybug2: EXPONENT (4), and Ladybug2: RESULT (10).

CACHE

CODE

MATH

Assign Operators

Add the RESULT and BASE variables to the multiplication operator



Chicks sprite has the correct code for exponents

Scratch IDE interface showing a project titled "Repeated Addition & Mult..." by aub_codes. The code area displays a script for a Ladybug2 sprite. A black box highlights the "Variables" panel on the left, where "BASE", "EXONENT", "FACTOR", and "RESULT" are checked. An arrow points from this box to the "set RESULT to RESULT * BASE" block in the code. The right panel shows the stage with variable monitors for "Tera" and "Ladybug2".

Variables:

- ADDEND
- BASE
- EXONENT
- FACTOR
- RESULT

Code:

```
when space key pressed  
  erase all  
  set size to 30 %  
  go to x: -230 y: -160  
  set BASE to 2  
  set EXONENT to 5  
  set RESULT to 1  
  repeat EXONENT  
    set RESULT to RESULT * BASE  
  repeat RESULT  
    stamp  
    wait 0.1 seconds  
    move 20 steps  
  change y by 35  
  set x to -230
```

Stage:

Tera: ADDEND	5	Ladybug2: BASE	3
Tera: FACTOR	2	Ladybug2: EXONENT	4
Tera: RESULT	10	Ladybug2: RESULT	10

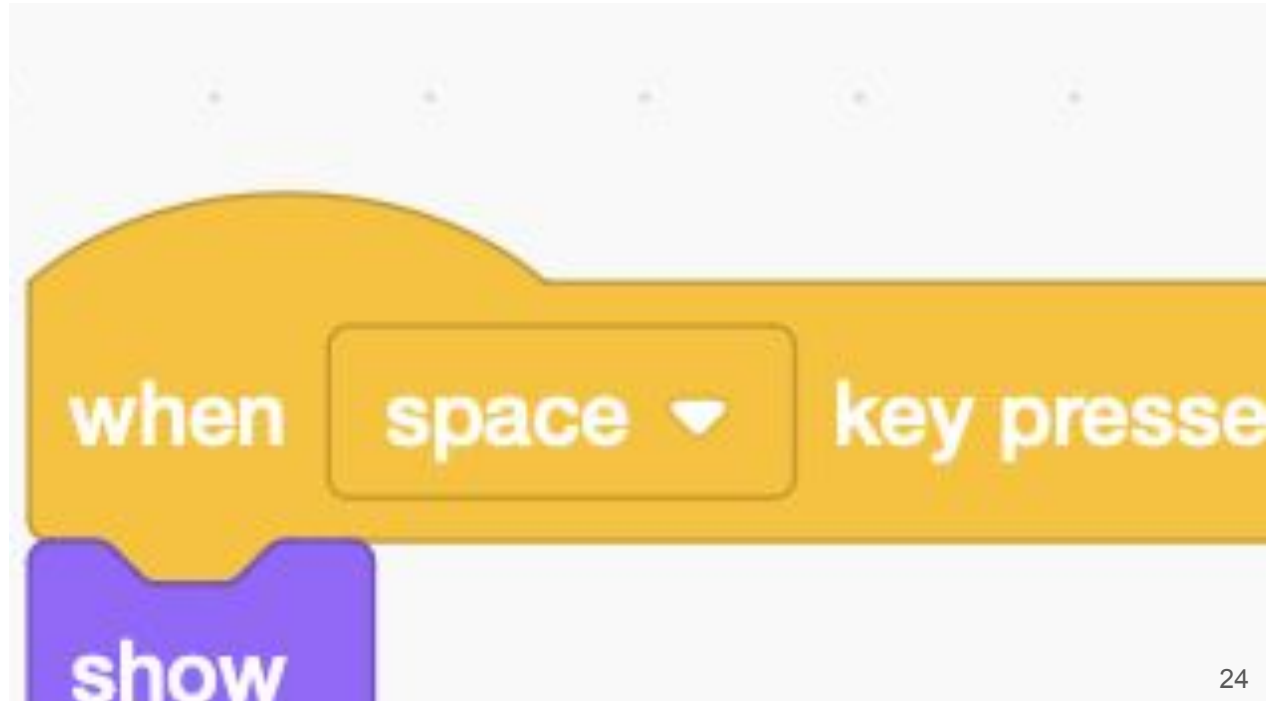
CACHE

CODE

MATH

Run the Code

Press
spacebar to
run the code!



CACHE

CODE

MATH

Uh Oh!

The ladybugs duplicate to a number where they start to duplicate outside the screen area



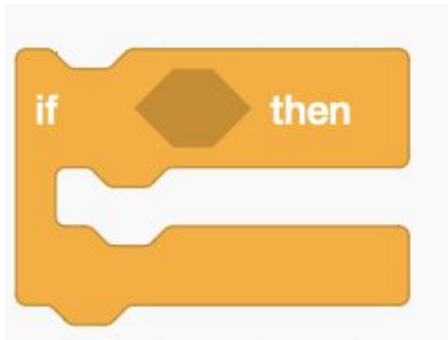
CACHE

CODE

MATH

Fix the Code

We can use an **“if, then”** control block and a sensing block to help the duplicating ladybugs stay within the screen.

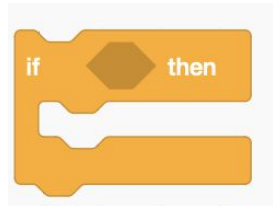


CACHE

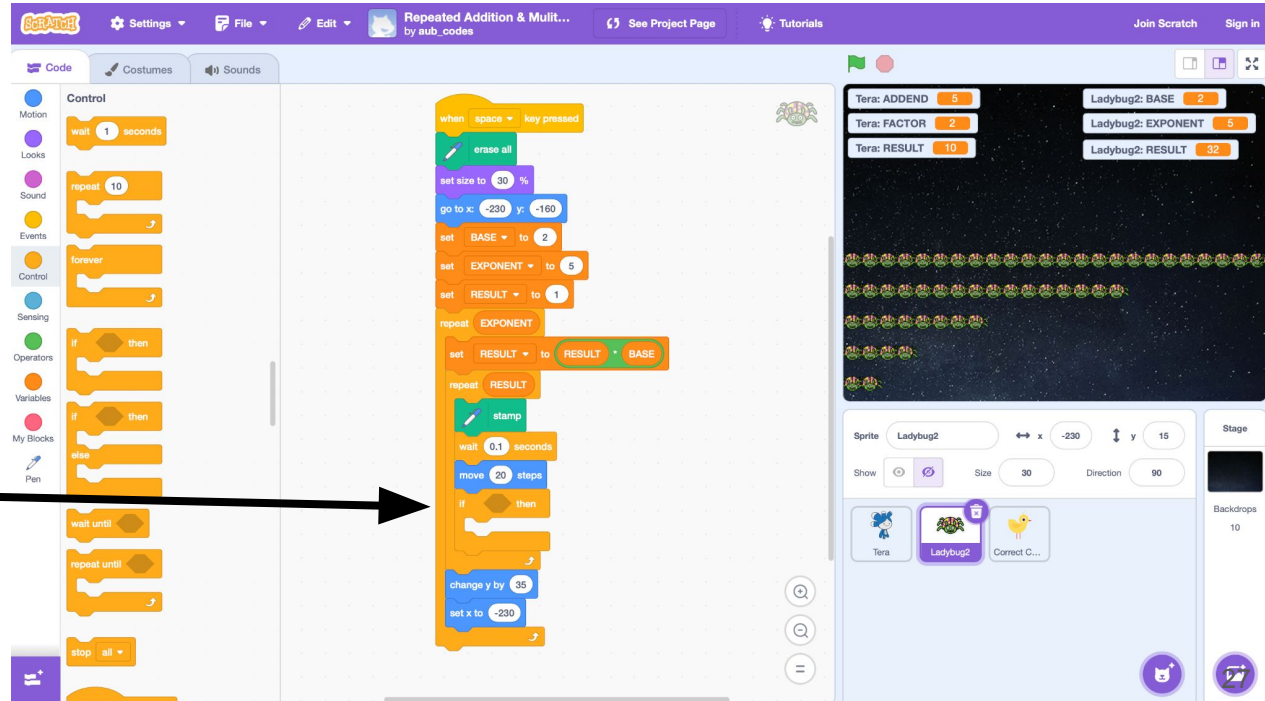
CODE

MATH

Control Block



Add **"If, then"** control block in the repeat RESULT block **under** move block.



The screenshot shows the Scratch code editor for a project titled "Repeated Addition & Mult... by aub_codes". The code is as follows:

```
when space key pressed
  erase all
  set size to 30 %
  go to x: -230 y: -160
  set BASE to 2
  set EXPONENT to 5
  set RESULT to 1
  repeat EXPONENT
    set RESULT to RESULT * BASE
    repeat RESULT
      stamp
      wait 0.1 seconds
      move 20 steps
      if then
      change y by 35
    set x to -230
```

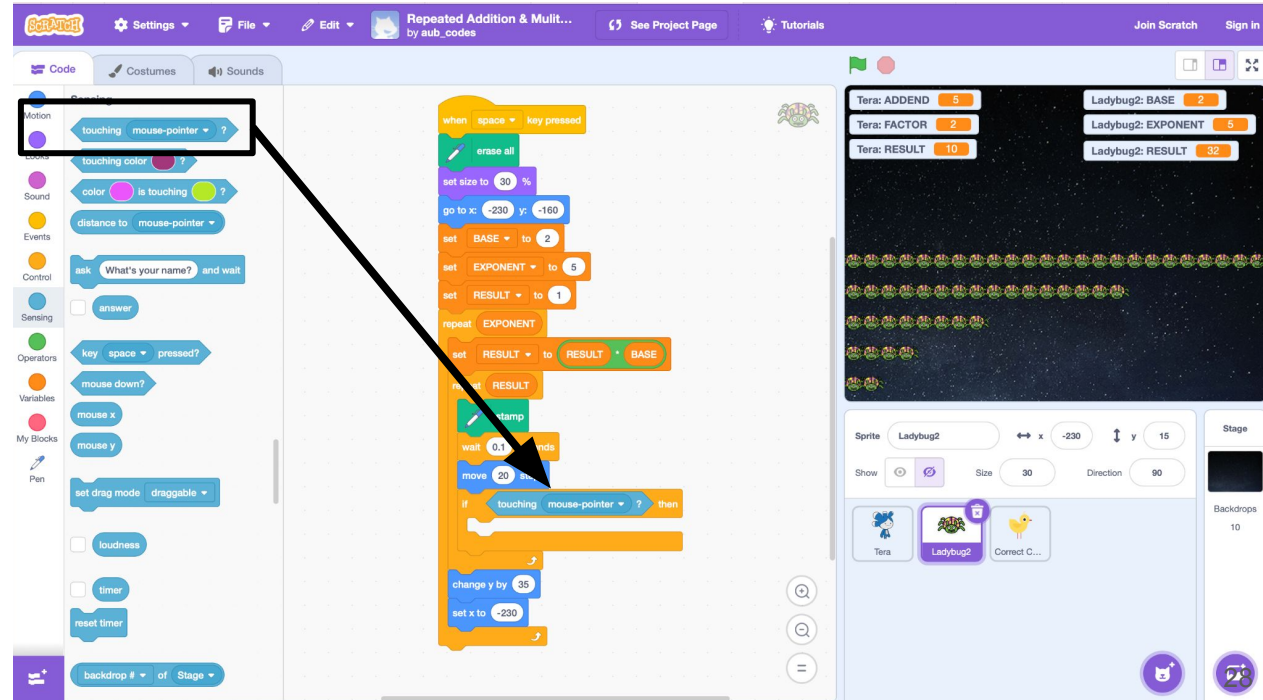
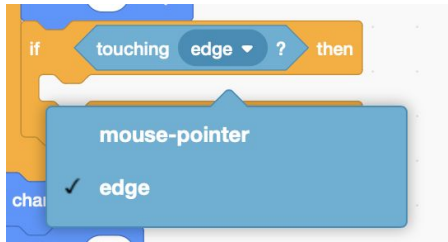
The right side of the editor shows the stage with a ladybug sprite named "Ladybug2" at x: -230, y: 15. The stage also displays variables for "Tera" (ADDEND: 5, FACTOR: 2, RESULT: 10) and "Ladybug2" (BASE: 2, EXPONENT: 5, RESULT: 32).

CACHE

CODE

MATH

Sensing Block



Add touching block to the “if, then” control block and set to “edge.”

CACHE

CODE

MATH

Motion Block

```
if touching edge ? then
  change y by 15
  set x to -230
```

Add **“change y by”** motion block; set to 15.

Add **“set x to”** motion block; set to -230.

The screenshot shows the Scratch IDE interface. The top bar displays the project title "Repeated Addition & Mult..." and the user "by aub_codes". The left sidebar shows the "Motion" category selected, with various motion blocks listed. A black box highlights the "set x to -230" and "change y by 10" blocks in the sidebar, with an arrow pointing to the "set x to -230" block in the script. The script starts with a "when space key pressed" event, followed by "erase all", "set size to 30 %", "go to x: -230 y: -180", "set BASE to 2", "set EXPONENT to 5", "set RESULT to 1", a "repeat EXPONENT" loop containing "set RESULT to RESULT * BASE", a "stamp" block, "wait 0.1 seconds", "move 20 steps", an "if touching edge ? then" block containing "change y by 15" and "set x to -230", and "change y by 35" and "set x to -230" blocks. The right sidebar shows the "Ladybug2" sprite selected, with its position set to x: -230, y: 15, size 30, and direction 90. The stage shows a dark background with a row of Ladybug2 sprites.

CACHE

CODE

MATH

Understand the Code

```
when space key pressed
  erase all
  set size to 30 %
  go to x: -230 y: -160
  set BASE to 2
  set EXPONENT to 5
  set RESULT to 1
  repeat EXPONENT
    set RESULT to RESULT * BASE
  repeat RESULT
    stamp
    wait 0.1 seconds
    move 20 steps
    if touching edge ? then
      change y by 15
      set x to -230
    change y by 35
    set x to -230
```

The sprite's size is set to 30% of its original size. It needs to be small because we are going to "repeat" many of them.

Ladybug (sprite) goes to the bottom left corner of the screen.

The rows of Ladybug are repeated EXPONENT number of times (5) and the total is stored in the RESULT variable.

The answer is calculated by multiplying BASE by previous value of the RESULT variable.

An image of the sprite is "stamped" (copied) RESULT number of times.

Ladybug does not move off the stage and instead starts a new row.

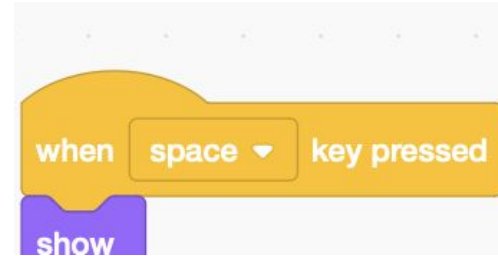
Ladybug moves up to the start of a new row.

CACHE

CODE

MATH

Run the Code



Click the green flag to run Tera.
Press the spacebar to run LadyBug2.

What differences do you notice?

CACHE

CODE

MATH

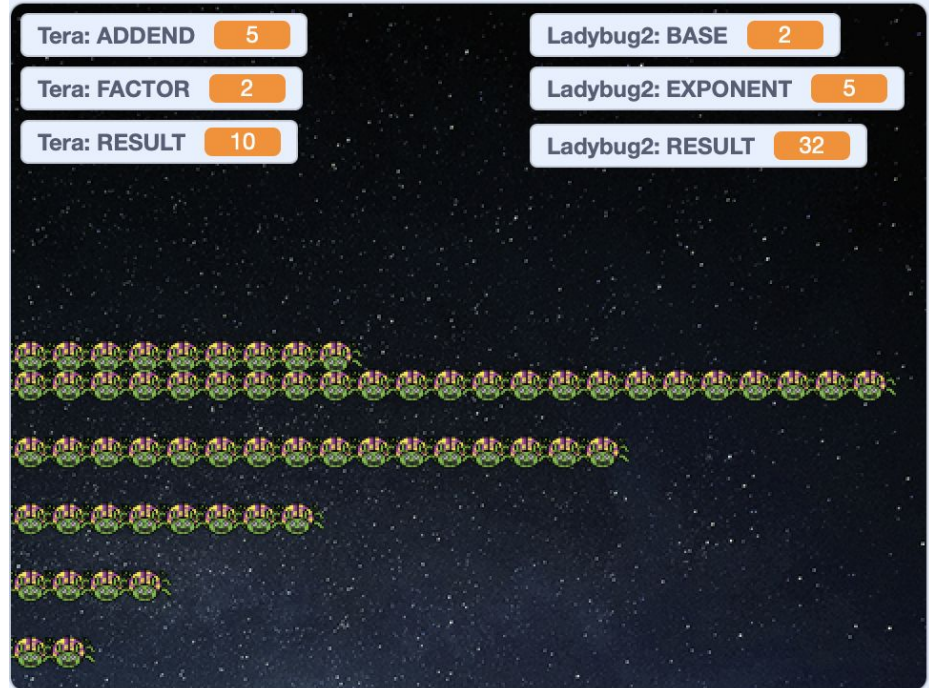
Understand the Programs

There are 2^5 ladybugs.

We get this answer by **repeated multiplication** of BASE and RESULT, EXPONENT number of times.

In a math equation, this is:

$$2 \times 2 \times 2 \times 2 \times 2 = 32$$



CACHE

CODE

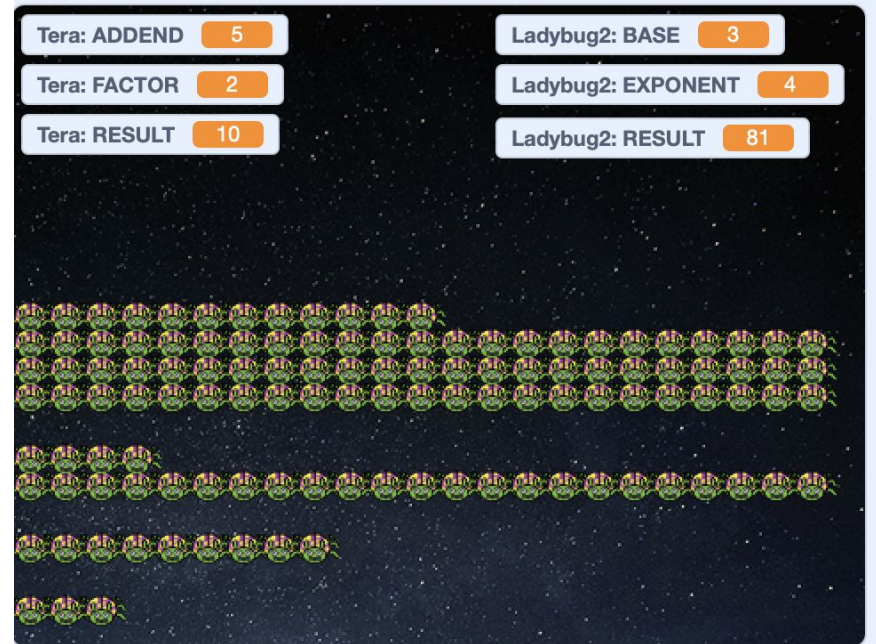
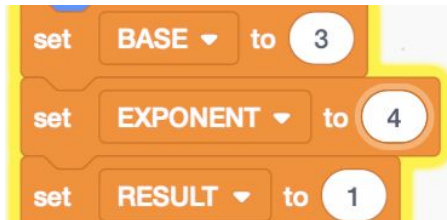
MATH

Your Turn!

Change the code to model

$$3^4 = 81$$

$$3 \times 3 \times 3 \times 3 = 81$$





CACHE

CODE

MATH

Your Turn!

Change the code to model an exponent of your choice!

Share with your neighbor.

CACHE

CODE

MATH

Complete Exit Ticket

Click:

https://usu.co1.qualtrics.com/jfe/form/SV_cT14yQ9tTxbQbRk