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

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

CACHE CODE MATH

Repeated Addition & Multiplication

Fall 2023

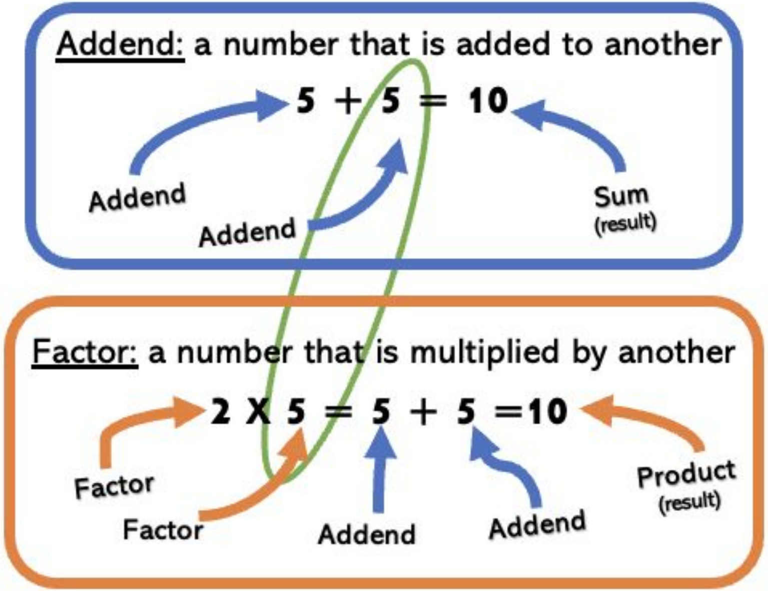
CACHE COUNTY SCHOOL DISTRICT

UtahState University

Stanford University

CACHE CODE MATH

Math Vocabulary



Addend: a number that is added to another

$5 + 5 = 10$

Addend

Addend

Sum (result)

Factor: a number that is multiplied by another

$2 \times 5 = 5 + 5 = 10$

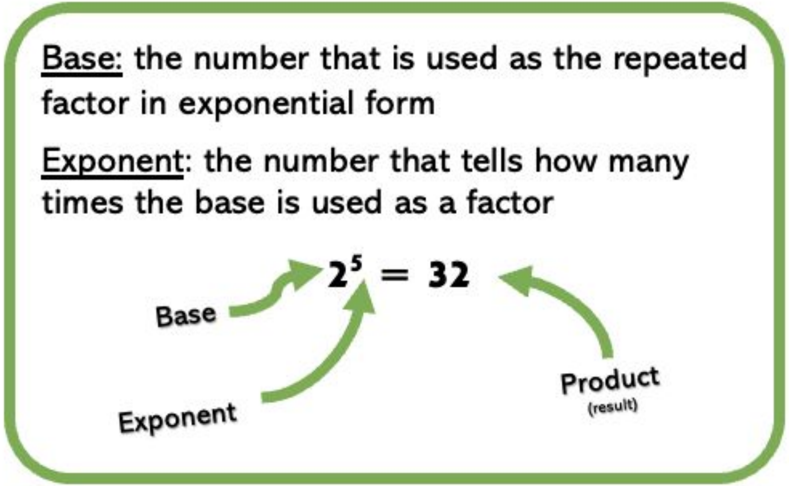
Factor

Factor

Addend

Addend

Product (result)



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

$2^5 = 32$

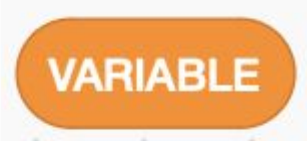
Base

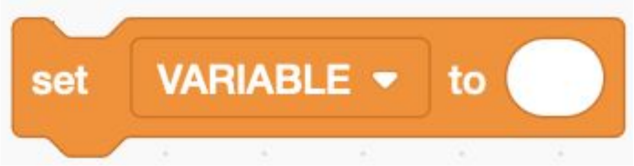
Exponent

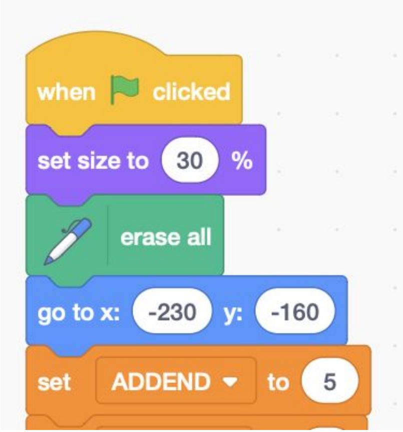
Product (result)

CACHE CODE MATH

Variable Block





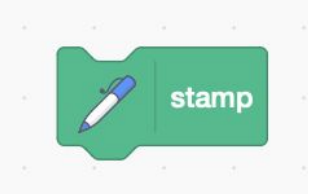


```
when ⚑ clicked
set size to 30 %
erase all
go to x: -230 y: -160
set ADDEND to 5
```

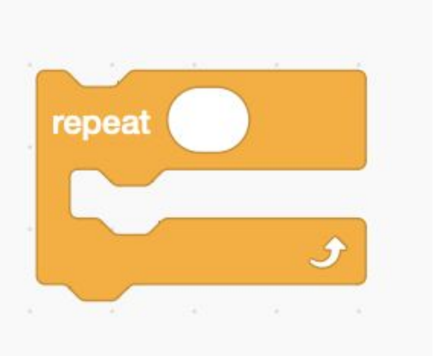
Sets the specified variable to the amount entered

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

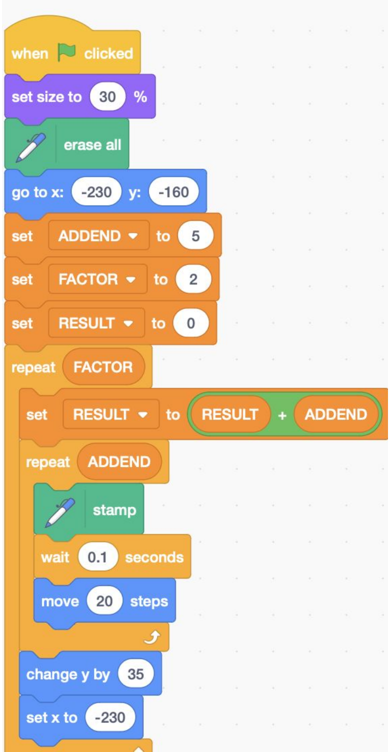
Stamp and Repeat Block



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.



```
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
```

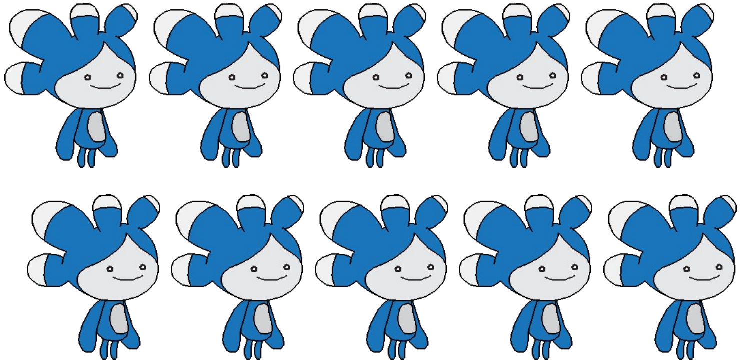
How many times will the 2nd repeat block loop?
How many stamps will there be?

CACHE CODE MATH

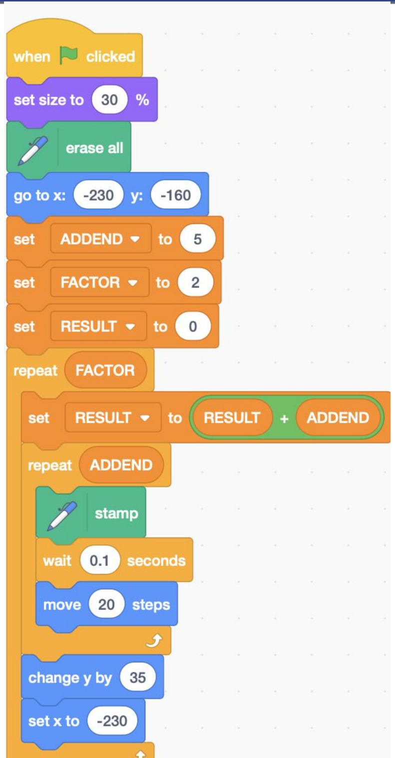
Repeated Addition

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

5 + 5 = 10
5 x 2 = 10



Read & Predict



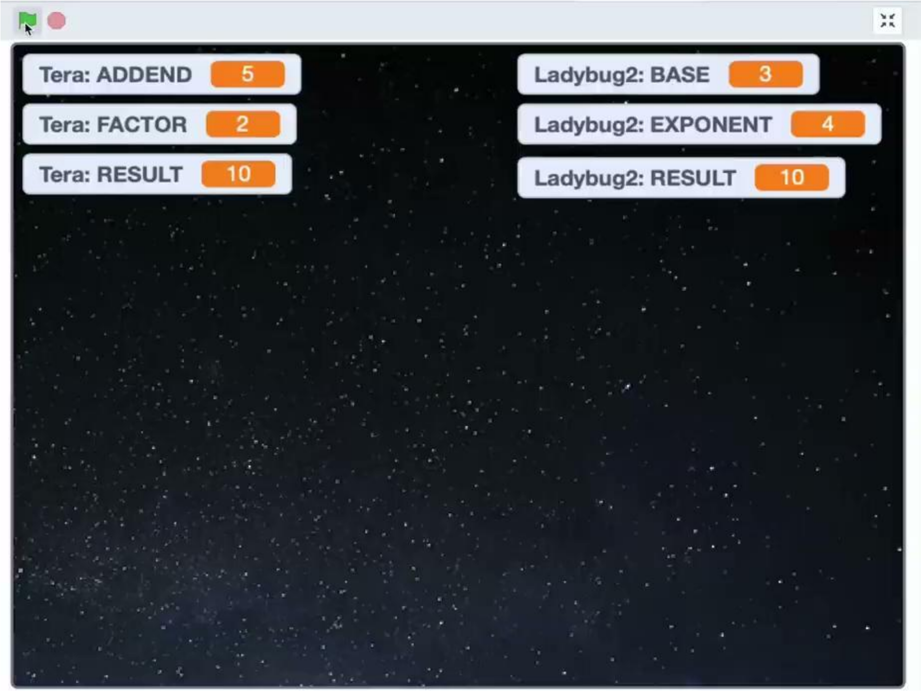
```
when 🏴 clicked
set size to 30 %
erase all
go to x: -230 y: -160
set ADDEND to 5
set FACTOR to 2
set RESULT to 0
repeat FACTOR
    set RESULT to RESULT + ADDEND
    repeat ADDEND
        stamp
        wait 0.1 seconds
        move 20 steps
    ⤴
    change y by 35
    set x to -230
```

What do you think this code will do?

What will be the final value of the variable RESULT?

CACHE CODE MATH

Watch the Code



Tera: ADDEND 5

Tera: FACTOR 2

Tera: RESULT 10

Ladybug2: BASE 3

Ladybug2: EXPONENT 4

Ladybug2: RESULT 10

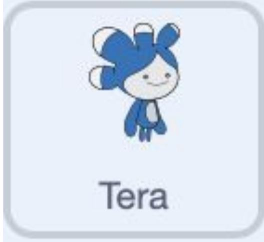
See the Code

Go the the following URL & click “See Inside”

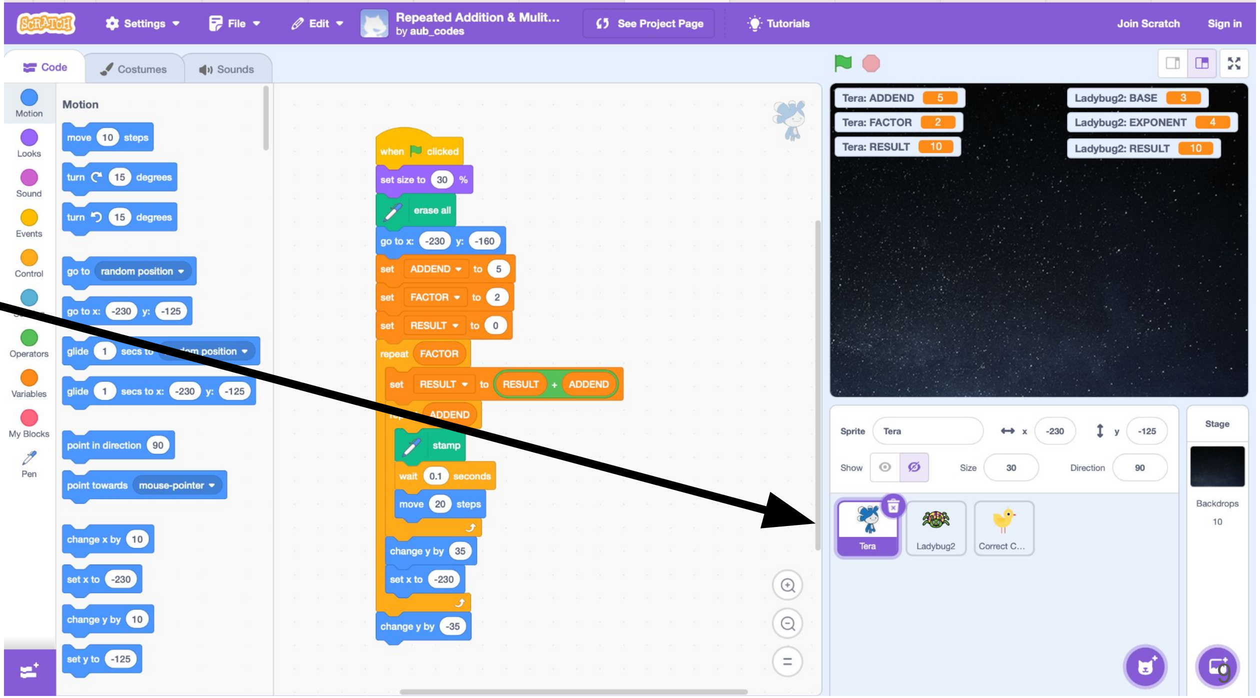
https://scratch.mit.edu/projects/873533865

CACHE CODE MATH

Select Tera



Click on the Tera sprite.



Motion

```
move 10 steps
turn ↻ 15 degrees
turn ↺ 15 degrees
go to random position
go to x: -230 y: -125
glide 1 secs to random position
glide 1 secs to x: -230 y: -125
point in direction 90
point towards mouse-pointer
change x by 10
set x to -230
change y by 10
set y to -125
```

```
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
change y by -35
```

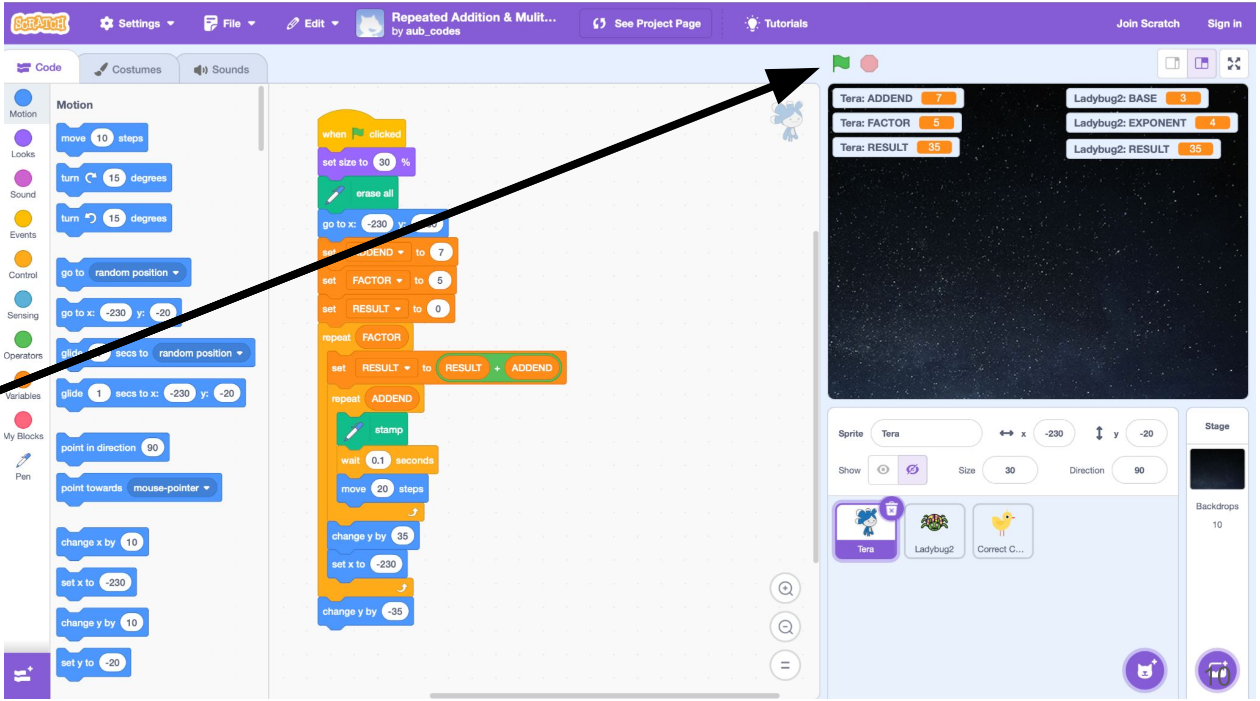
Tera: ADDEND	5
Tera: FACTOR	2
Tera: RESULT	10
Ladybug2: BASE	3
Ladybug2: EXPONENT	4
Ladybug2: RESULT	10

Sprite: Tera   x: -230   y: -125
Show   Size: 30   Direction: 90

CACHE
CODE
MATH

Run the Code

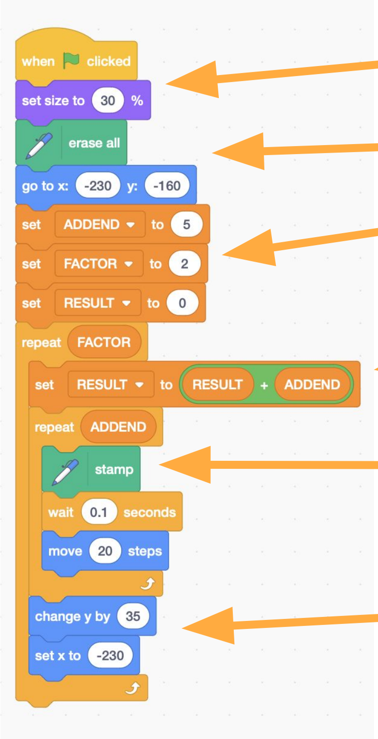
Click the green flag to run the code!



```
when 🏴 clicked
set size to 30 %
erase all
go to x: -230 y: -20
set ADDEND to 7
set FACTOR to 5
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
change y by -35
```

Tera: ADDEND	7
Tera: FACTOR	5
Tera: RESULT	35
Ladybug2: BASE	3
Ladybug2: EXPONENT	4
Ladybug2: RESULT	35

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.

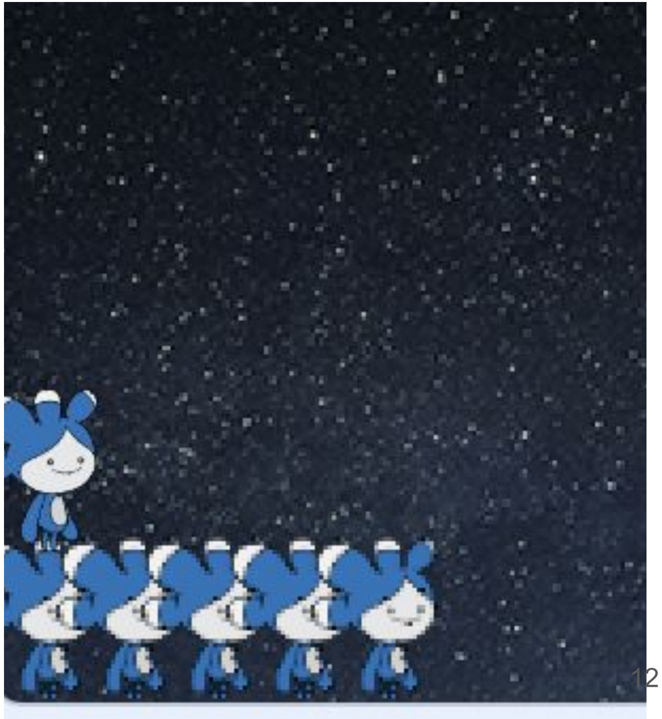
Tera moves up to the start of a new row.

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.

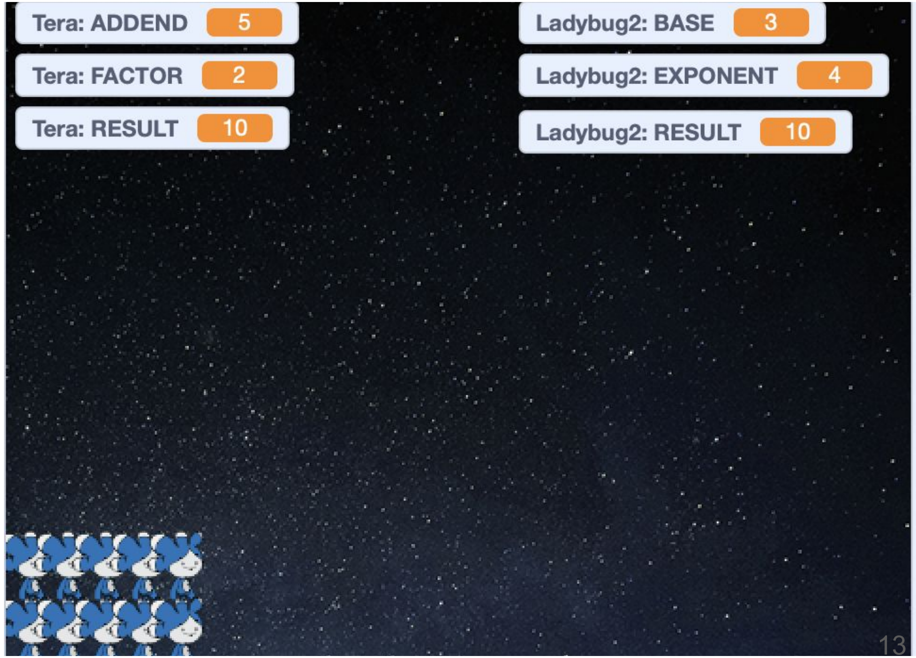


Understand the Code

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

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

5 x 2 = 10
5 + 5 = 10



Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10

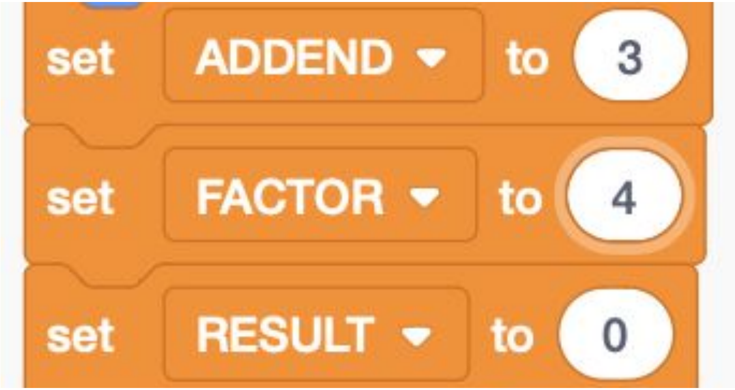
Ladybug2: BASE 3
Ladybug2: EXPONENT 4
Ladybug2: RESULT 10

Your Turn!

Change Tera's code to model

3 x 4 = 12

3+3+3+3 = 12



```
set ADDEND ▾ to 3
set FACTOR ▾ to 4
set RESULT ▾ to 0
```

CACHE CODE MATH

Your Turn!

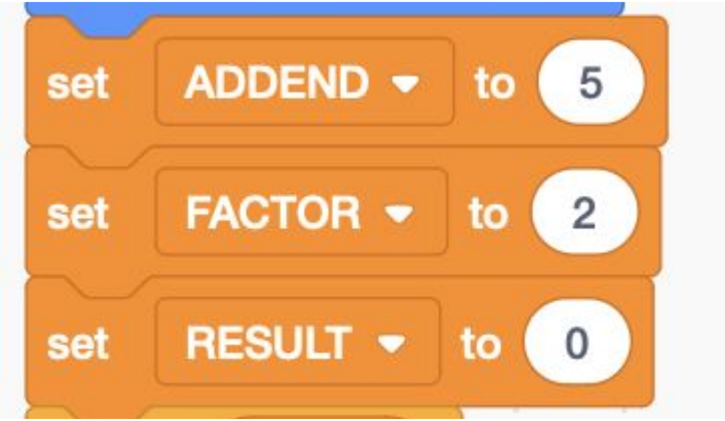
Change the code to model a multiplication problem of your choice!

Share with your neighbor.

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.



```
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.

$2^5 = 32$

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

Modifying Tera

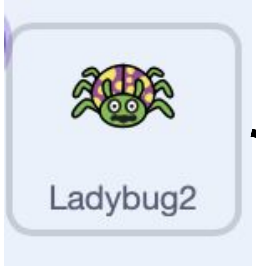
We will modify the repeated addition code to program repeated multiplication.

2^5

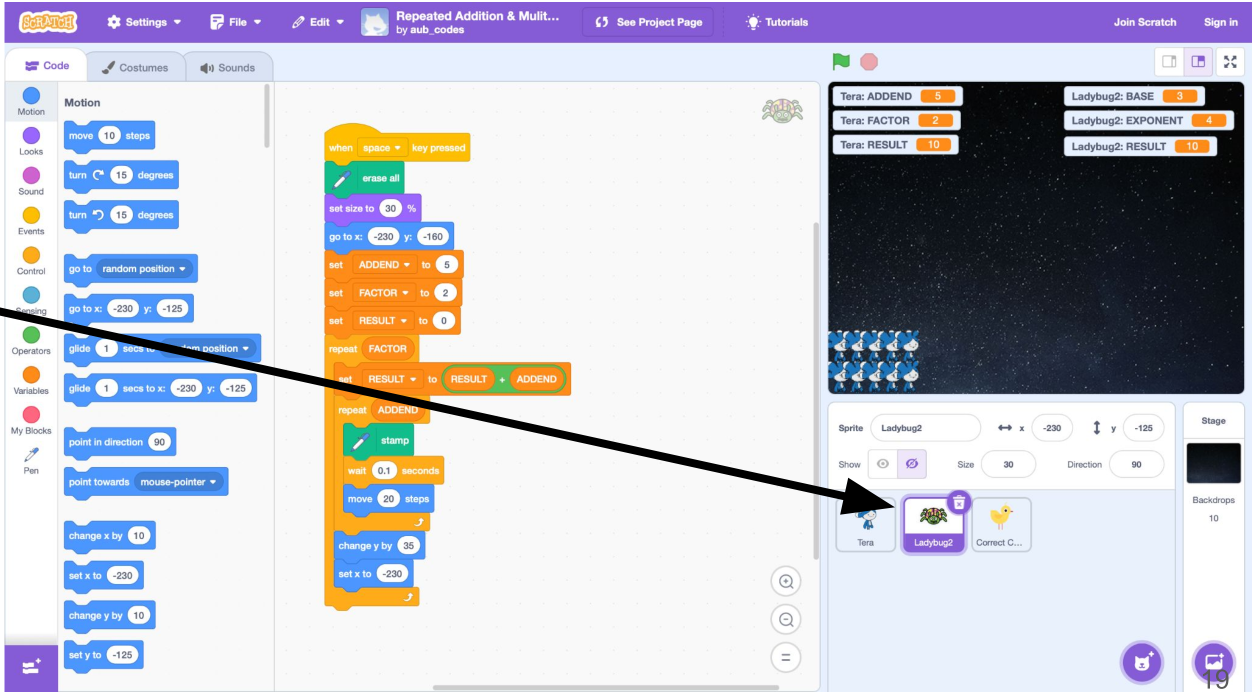
2 x 2 x 2 x 2 x 2= 32

CACHE CODE MATH

Select Ladybug



Click on the Ladybug2 sprite.



```
when space key pressed
erase all
set size to 30 %
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
```

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

Sprite Ladybug2 x -230 y -125

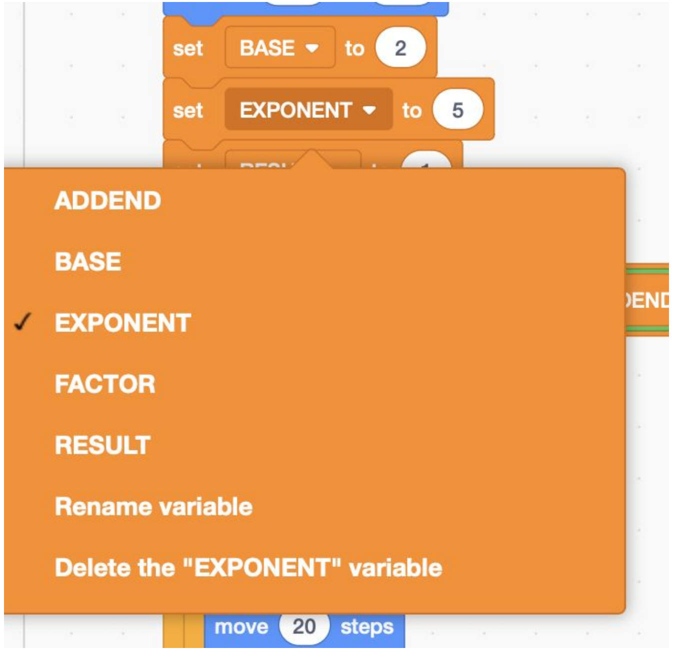
Show Size 30 Direction 90

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.**



```
set BASE ▾ to 2
set EXPONENT ▾ to 5
```

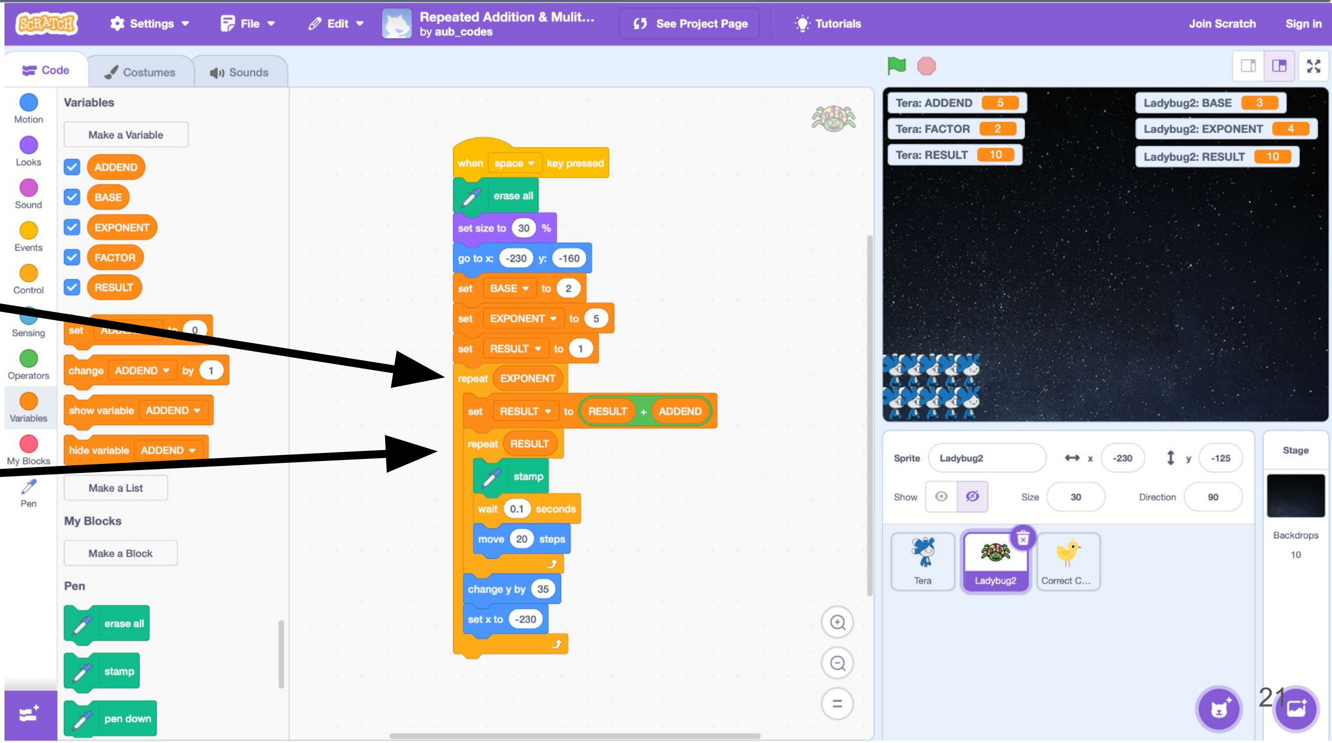
Chicks sprite has the correct code for exponents

Repeat Blocks

In repeat blocks:

Change FACTOR to EXPONENT

Change ADDEND to RESULT



☑ ADDEND
☑ BASE
☑ EXPONENT
☑ FACTOR
☑ RESULT

```
set ADDEND to 0
change ADDEND by 1
show variable ADDEND
hide variable ADDEND
erase all
stamp
pen down
```

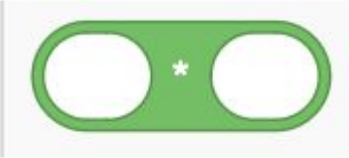
```
when space key pressed
erase all
set size to 30 %
go to x: -230 y: -160
set BASE to 2
set EXPONENT to 5
set RESULT to 1
repeat EXPONENT
    set RESULT to RESULT + ADDEND
    repeat RESULT
        stamp
        wait 0.1 seconds
        move 20 steps
    change y by 35
    set x to -230
```

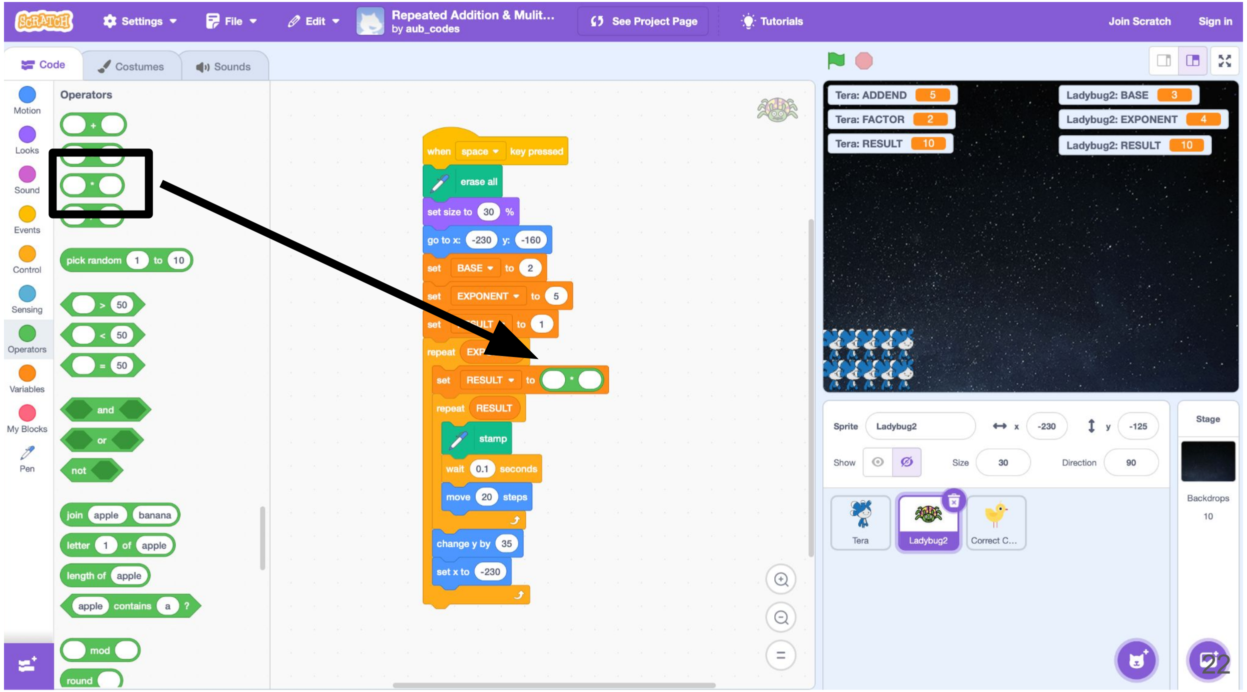
Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10
Ladybug2: BASE 3
Ladybug2: EXPONENT 4
Ladybug2: RESULT 10

Sprite Ladybug2 x -230 y -125
Show Size 30 Direction 90

Operator Block

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





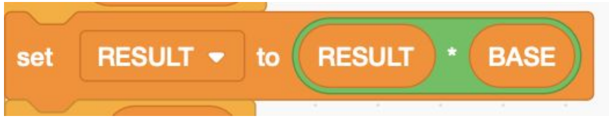
Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10
Ladybug2: BASE 3
Ladybug2: EXPONENT 4
Ladybug2: RESULT 10

```
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 EXP
    set RESULT to ( * )
    repeat RESULT
        stamp
        wait 0.1 seconds
        move 20 steps
    change y by 35
    set x to -230
```

CACHE CODE MATH

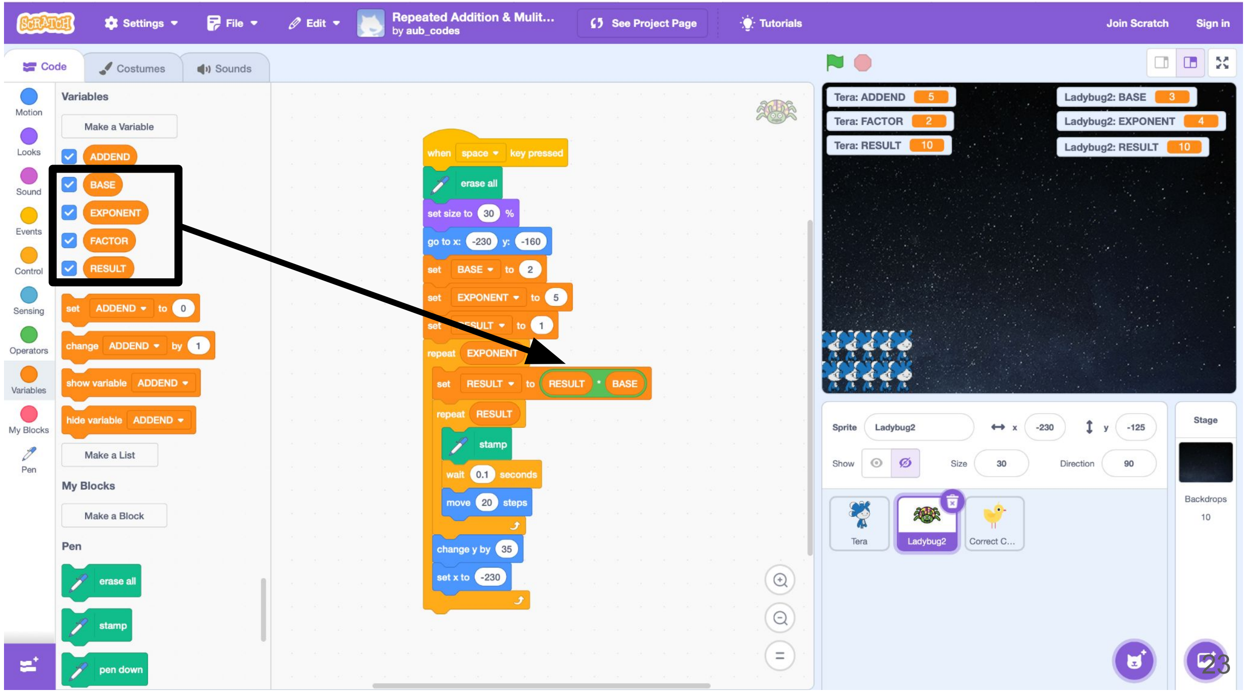
Assign Operators

Add the RESULT and BASE variables to the multiplication operator



```
set RESULT to (RESULT * BASE)
```

Chicks sprite has the correct code for exponents



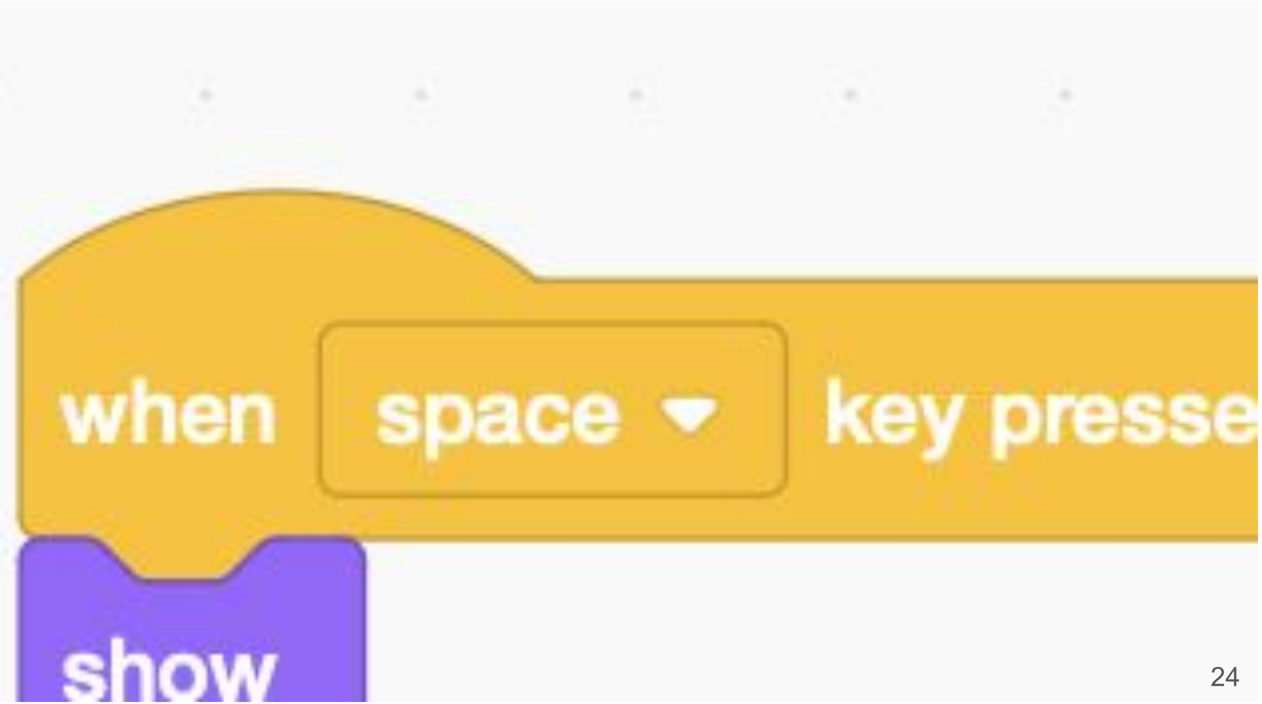
```
when space key pressed
erase all
set size to 30 %
go to x: -230 y: -160
set BASE to 2
set EXPONENT to 5
set RESULT to 1
repeat EXPONENT
    set RESULT to (RESULT * BASE)
    repeat RESULT
        stamp
        wait 0.1 seconds
        move 20 steps
    change y by 35
    set x to -230
```

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

Sprite Ladybug2 x -230 y -125
Show Size 30 Direction 90

Run the Code

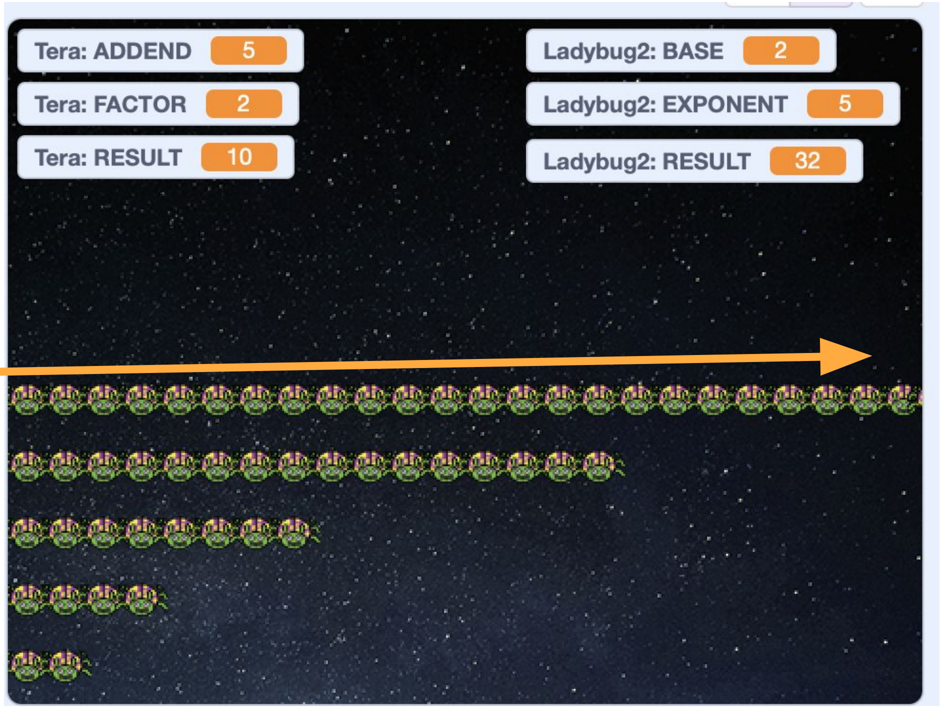
Press **spacebar** to run the code!



```
when space ▾ key presse
show
```

Uh Oh!

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



Tera: ADDEND 5

Tera: FACTOR 2

Tera: RESULT 10

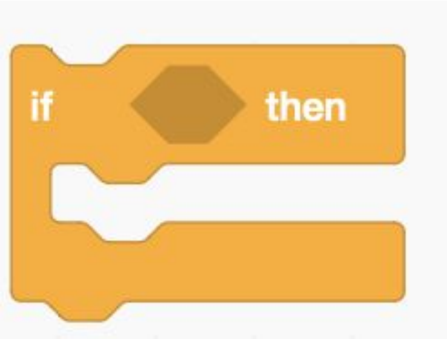
Ladybug2: BASE 2

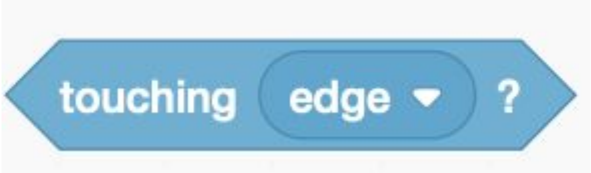
Ladybug2: EXPONENT 5

Ladybug2: RESULT 32

Fix the Code

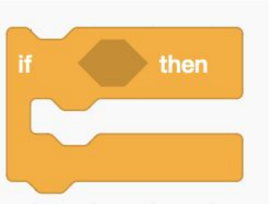
We can use an “**if, then**” control block and a sensing block to help the duplicating ladybugs stay within the screen.



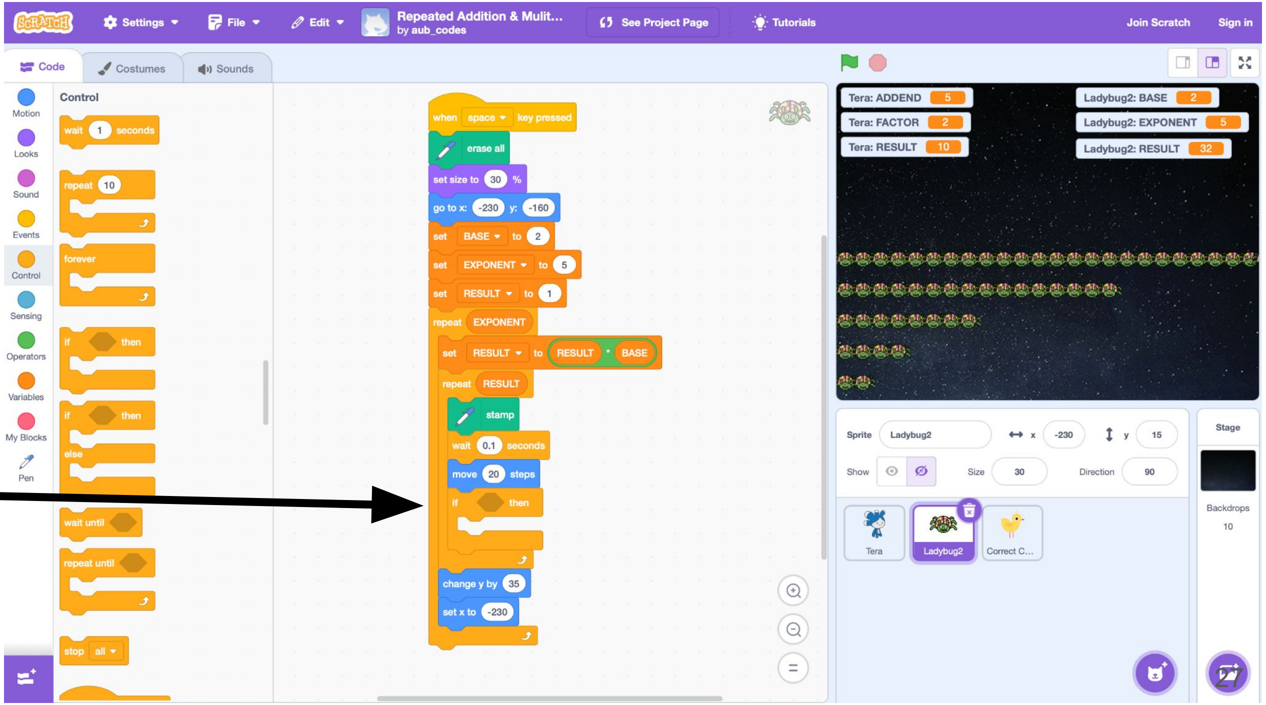


CACHE CODE MATH

Control Block



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

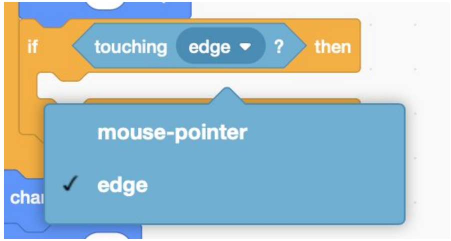


```
when space key pressed
erase all
set size to 30 %
go to x: -230 y: -160
set BASE to 2
set EXPONENT to 5
set RESULT to 1
repeat EXPONENT
    set RESULT to RESULT * BASE
    repeat RESULT
        stamp
        wait 0.1 seconds
        move 20 steps
        if then
    change y by 35
    set x to -230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

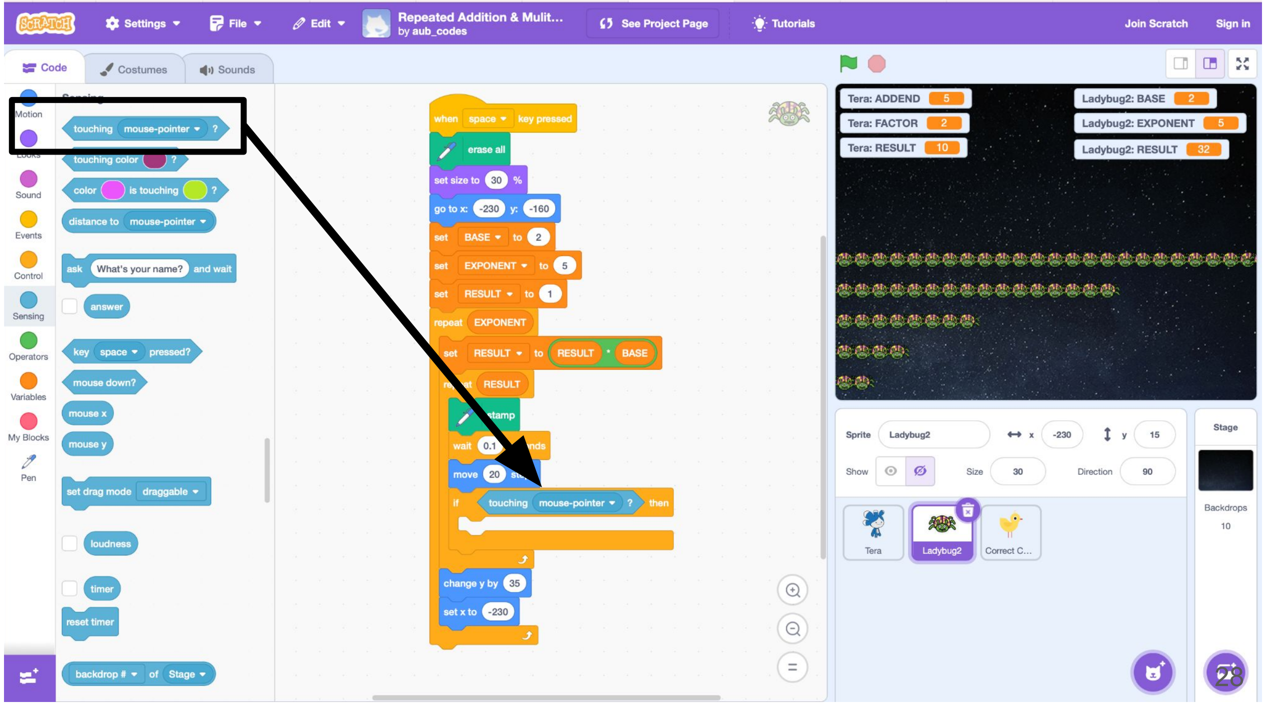
Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10
Ladybug2: BASE 2
Ladybug2: EXPONENT 5
Ladybug2: RESULT 32

Sprite Ladybug2 x -230 y 15
Show Size 30 Direction 90

Sensing Block

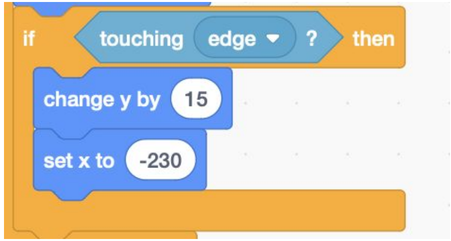


Add touching block to the **“if, then”** control block and set to “edge.”



CACHE CODE MATH

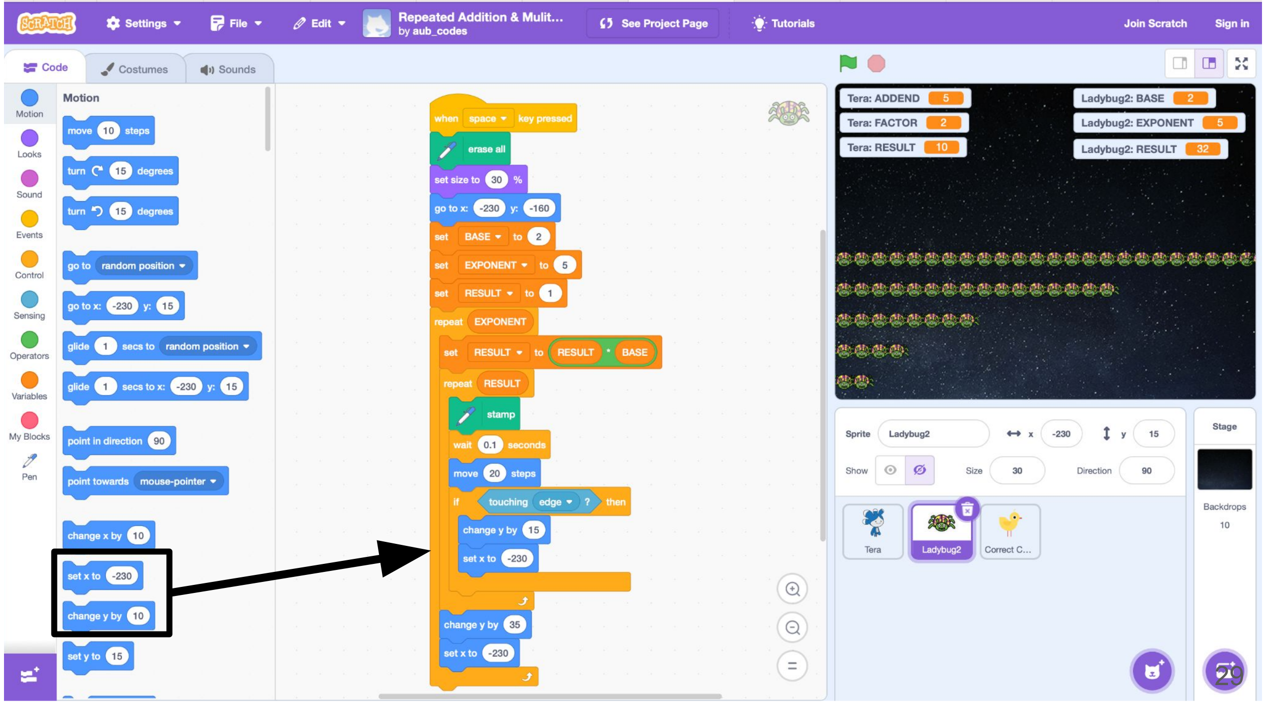
Motion Block



```
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.

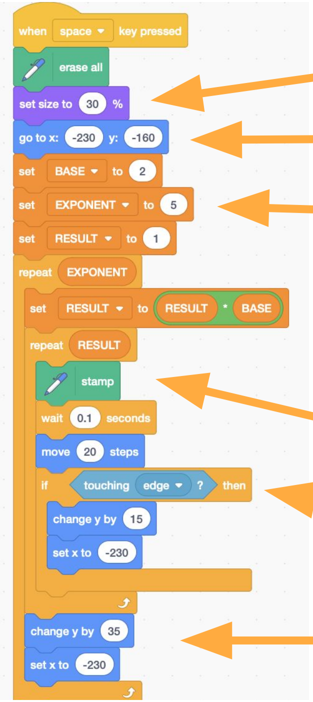


```
when space key pressed
erase all
set size to 30 %
go to x: -230 y: -160
set BASE to 2
set EXPONENT to 5
set RESULT to 1
repeat EXPONENT
    set RESULT to (RESULT * BASE)
    repeat RESULT
        stamp
        wait 0.1 seconds
        move 20 steps
        if touching edge ? then
            change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

Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10
Ladybug2: BASE 2
Ladybug2: EXPONENT 5
Ladybug2: RESULT 32

Sprite Ladybug2 x -230 y 15
Show Size 30 Direction 90

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.

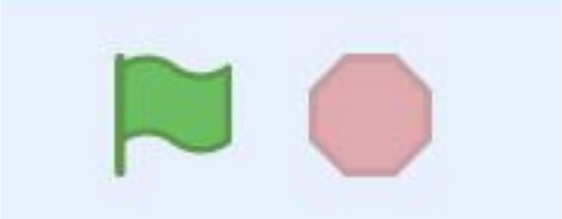
The answer is calculated by multiplying BASE by previous value of the RESULT variable.

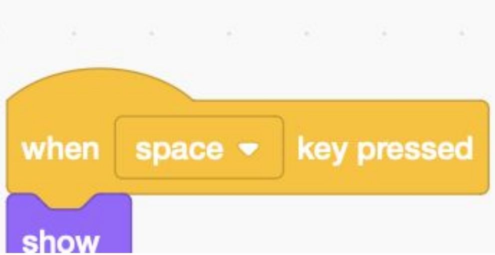
An image of the sprite is "stamped" (copied) RESULT number of times.

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

Ladybug moves up to the start of a new row.

Run the Code





```
when space ▾ key pressed
show
```

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

What differences do you notice?

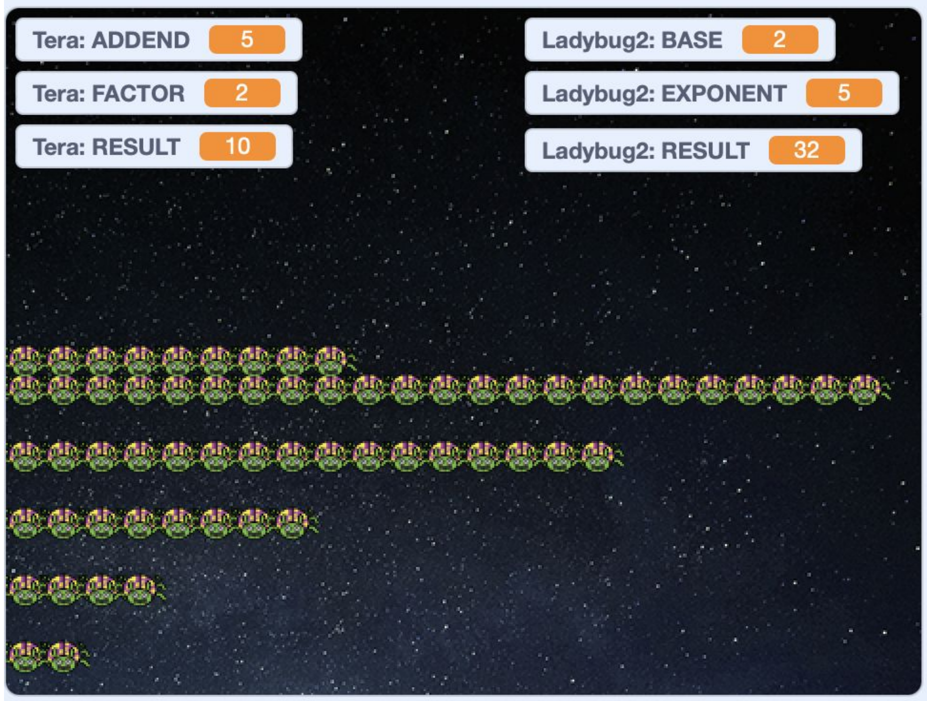
CACHE CODE MATH

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:
2x2x2x2x2= 32



Tera: ADDEND 5
Tera: FACTOR 2
Tera: RESULT 10

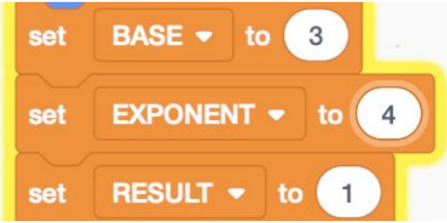
Ladybug2: BASE 2
Ladybug2: EXPONENT 5
Ladybug2: RESULT 32

Your Turn!

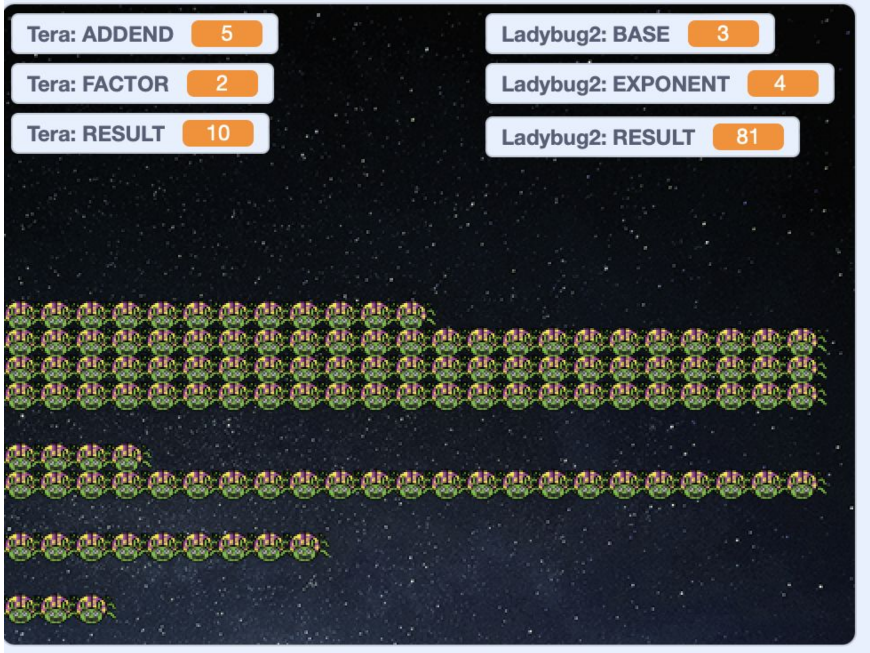
Change the code to model

$3^4 = 81$

3x3x3x3 = 81



```
set BASE to 3
set EXPONENT to 4
set RESULT to 1
```



Tera: ADDEND	5
Tera: FACTOR	2
Tera: RESULT	10
Ladybug2: BASE	3
Ladybug2: EXPONENT	4
Ladybug2: RESULT	81

CACHE CODE MATH

Your Turn!

Change the code to model an exponent of your choice!

Share with your neighbor.

Complete Exit Ticket

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