Coding Algebraic Expressions to
 Generate Linear Growing Patterns

**Algebra**

**Unit 3 Line Master 2a**

You will alter existing code to generate various linear growing patterns that are described using algebraic expressions.
To do this, you will alter the rate of change (multiplier) and
initial values in the code.

Begin with the following application:

<https://scratch.mit.edu/projects/795444171/editor/>

 If you have a Scratch login, save the project
 in your Scratch account by selecting **Remix**
 at the top of the screen.

 A login is not required to work with the code,
 but you will not be able to save your changes
 without it.

**Note:** Like the first code you created, this code uses both variables and lists.
In this code, **termNumber** and **termValue** are variables, while **termValueList**and **termNumberList** are lists.



1. Click on the **green flag** to execute the code.
 You will see that two lists are generated and
 displayed on the stage—one list called
 **termNumberList**that contains the term
 numbers and one list called **termValueList**that contains the term values.

 Coding Algebraic Expressions to
 Generate Linear Growing Patterns (cont’d)

**Algebra**

**Unit 3 Line Master 2b**

 For term number 0, our term value is 0.

 For term number 1, our term value is 3.

 For term number 2, our term value is 6.

 For term number 3, our term value is 9, and so on.

* What operation do we perform on the term number to get

 the term value?

 Did you answer “multiply by 3?” That’s correct!

 If you aren’t certain how we got this answer, look at the
 relationship between the two sets of data:

 By what do you have to multiply 1 to get 3?

 By what do you have to multiply 2 to get 6?

 By what do you have to multiply 3 to get 9?

2. Let’s take a closer look at the algebraic expression that is
 used to generate the term values in the code:

 

 The **termValue** variable is set to: **3 \* termNumber+ 0**

 In coding, an “\*” indicates *multiplication*, so this can be thought
 of as 3*x* + 0, where *x* is the termNumber.

 a) In the code, alter the expression to the following:

 

 Before executing the code, predict what will be output
 in the two lists.

 b) Click on the green flag to execute the code.
 Was your prediction correct?

 Coding Algebraic Expressions to
 Generate Linear Growing Patterns (cont’d)

**Algebra**

**Unit 3 Line Master 2c**

3. Alter the expression in the code to generate each list of numbers.
 For each set of values, write down the expression you use
 to generate the term values. The first expression has been
 included for you.

|  |  |  |
| --- | --- | --- |
|  a) 5 \* termNumber + 0 Graphical user interface, chart  Description automatically generated | b) \_\_\_\_\_\_\_\_\_\_\_\_\_ Graphical user interface, chart  Description automatically generated | c) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Graphical user interface, chart  Description automatically generated |

4. Now, let’s alter the expression in the code to use an initial value
 of 2, so, that at termNumber 0, the termValue will start at 2.

 The new expression is: 3 \* termNumber + 2

 a) Alter the Scratch code now to reflect this change:



 Before executing the code, predict what will be outputted
 in the lists.

 b) Click on the green flag to execute the code.
 Was your prediction correct?

 Coding Algebraic Expressions to
 Generate Linear Growing Patterns (cont’d)

**Algebra**

**Unit 3 Line Master 2d**

5. Alter the expression in the code to generate the lists of numbers
 below. For each set of values, write down the expression you
 use to generate the term numbers on the blank line.

**Hint:** The initial value is the value of term 0 and is listed as **termNumber** 0. The constant rate, or multiplier, is the amount by which the values in the **termValueList** are changing.

 

|  |  |  |
| --- | --- | --- |
|  a) \_\_\_\_\_\_\_\_\_\_\_\_\_  | b) \_\_\_\_\_\_\_\_\_\_\_\_\_  | c) \_\_\_\_\_\_\_\_\_\_\_\_\_\_  |

6. Change the expression back to: 3 \* termNumber + 2
 Now, you’ll alter the code so that you output the term numbers
 and values all the way to term number 10.

 To do this, change the *defined count* in the repeat to
 **termNumber*>*10**. This will ensure the term values will be
 output up to and including term number 10.

 **defined count:** In coding, the number
 of times instructions are repeated**,**
 based on a predefined value or until
 a condition has been met.

 Coding Algebraic Expressions to
 Generate Linear Growing Patterns (cont’d)

**Algebra**

**Unit 3 Line Master 2e**

7. Alter the expression and the defined count in the code to generate
 the following lists of numbers. For each set of values, write down
 the expression you use to generate the term values on the blank
 line. Write the number you use for the defined count as well.

|  |  |  |
| --- | --- | --- |
|  a) \_\_\_\_\_\_\_\_\_\_\_\_\_ defined count: \_\_\_\_\_  | b) \_\_\_\_\_\_\_\_\_\_\_\_\_ defined count: \_\_\_\_\_  | c) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ defined count: \_\_\_\_\_  |

8. Finally, let’s alter the code so that the termNumber variable
 increases by an amount other than 1. That is, we will alter the
 change termNumber by 1block.

 Let’s start with our original expression, **3 \* termNumber + 2**,
 but output up to term number100, increasing our **termNumber**
 by **10** each time. The altered code and output are shown
 on the next page.

 Coding Algebraic Expressions to
 Generate Linear Growing Patterns (cont’d)

**Algebra**

**Unit 3 Line Master 2f**

 

Change this number to 100.

Change this number from 1 to 10.

9. Alter the code to output the following lists of numbers.
 For each set of values, write down the expression you use
 to generate the term values on the blank line.
 What expression did you use in the repeat until block?
 By what did you increase the termNumber variable each time?

|  |  |
| --- | --- |
|  a) \_\_\_\_\_\_\_\_\_\_\_\_\_ defined count: \_\_\_\_\_  | b) \_\_\_\_\_\_\_\_\_\_\_\_\_ defined count: \_\_\_\_\_  |