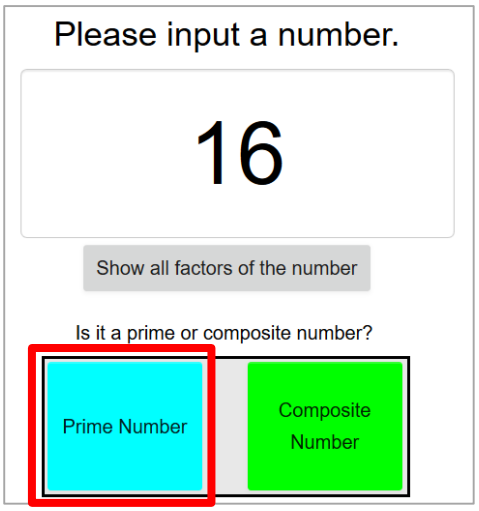
FACTOR APP



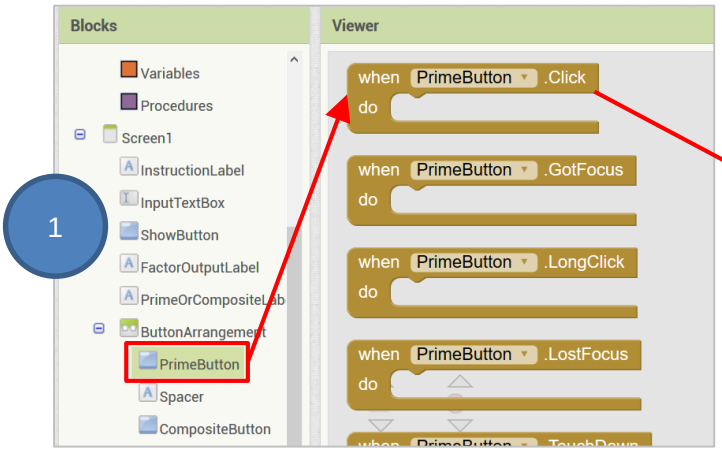
START HERE

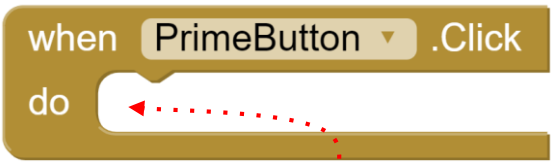
❑ Open your Factor App project.

PRIME BUTTON

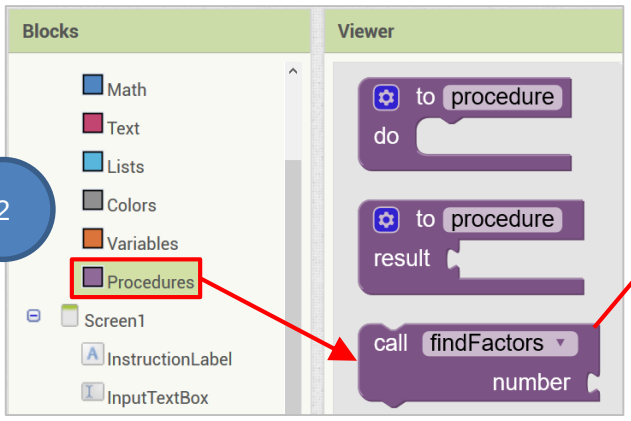
If the user clicks on the **PrimeButton**, you will need to check if the number of factors is equal to 2.

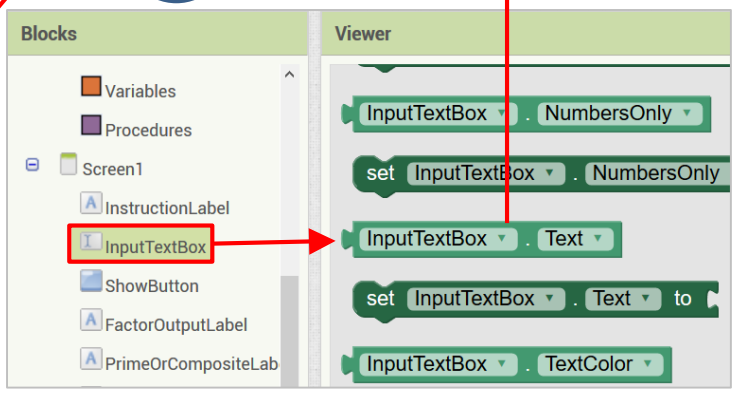
❑ Pull out **PrimeButton.Click** block from the PrimeButton drawer.





❑ Pull out the **call findFactors** block from the Procedures drawer and snap the **InputTextBox.Text** block to this block.

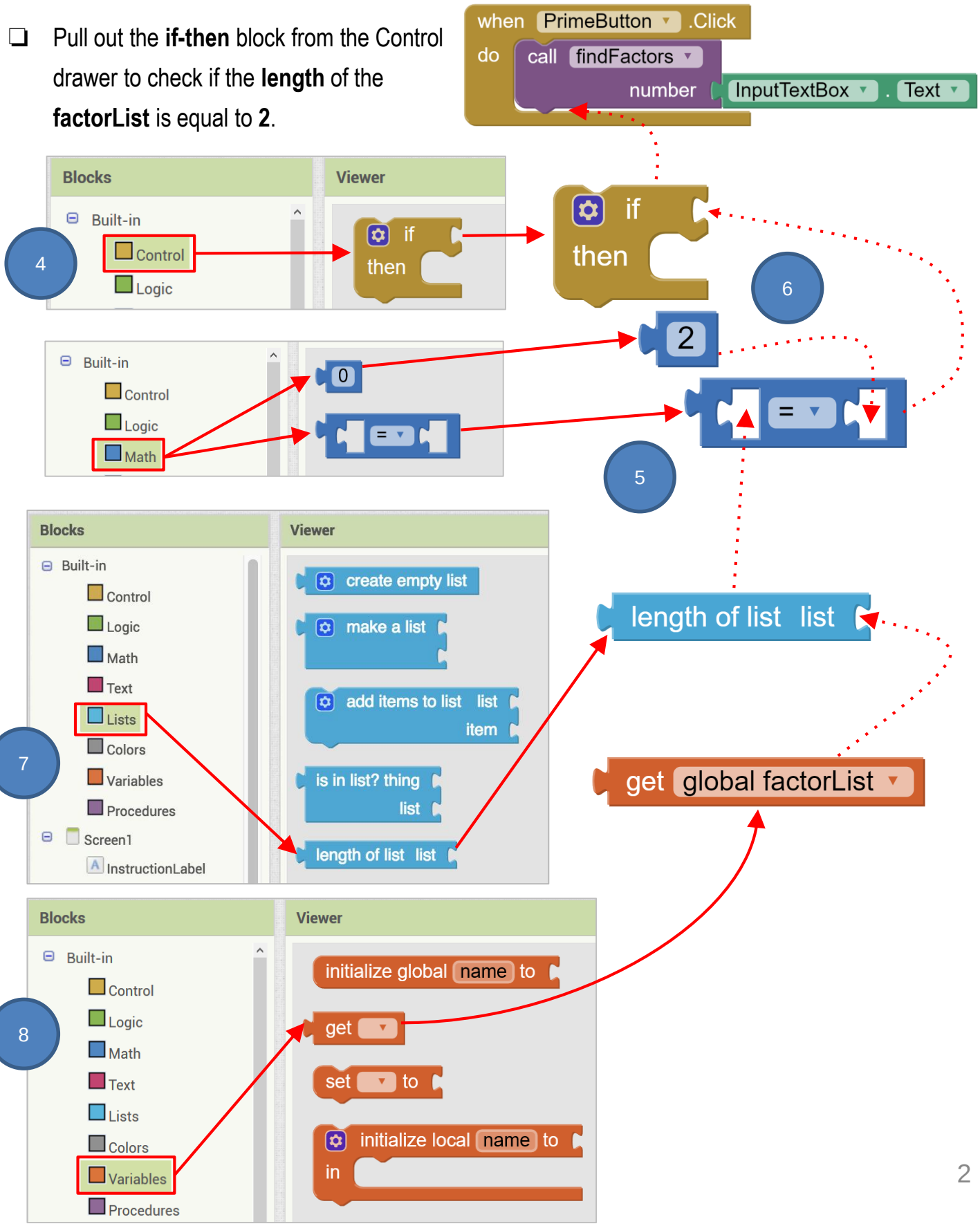




FACTOR APP

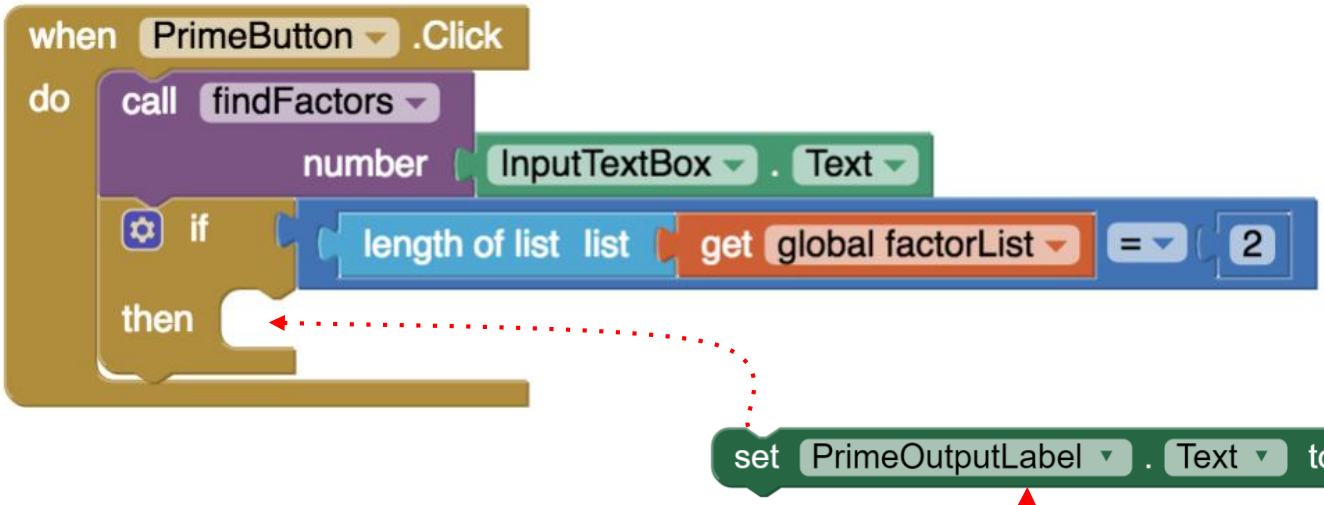
PRIME BUTTON (CONT'D)

- ❑ Pull out the **if-then** block from the Control drawer to check if the **length** of the **factorList** is equal to **2**.

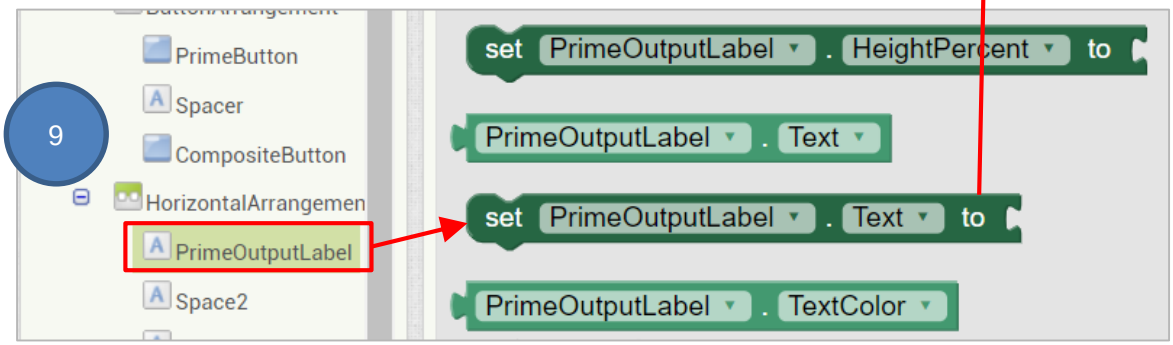


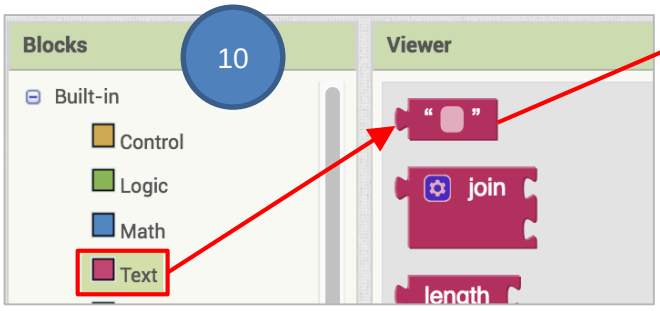
FACTOR APP

PRIME BUTTON (CONT'D)



❑ Snap a text block with "**This is a prime number.**" to the **set PrimeOutputLabel.Text** block.





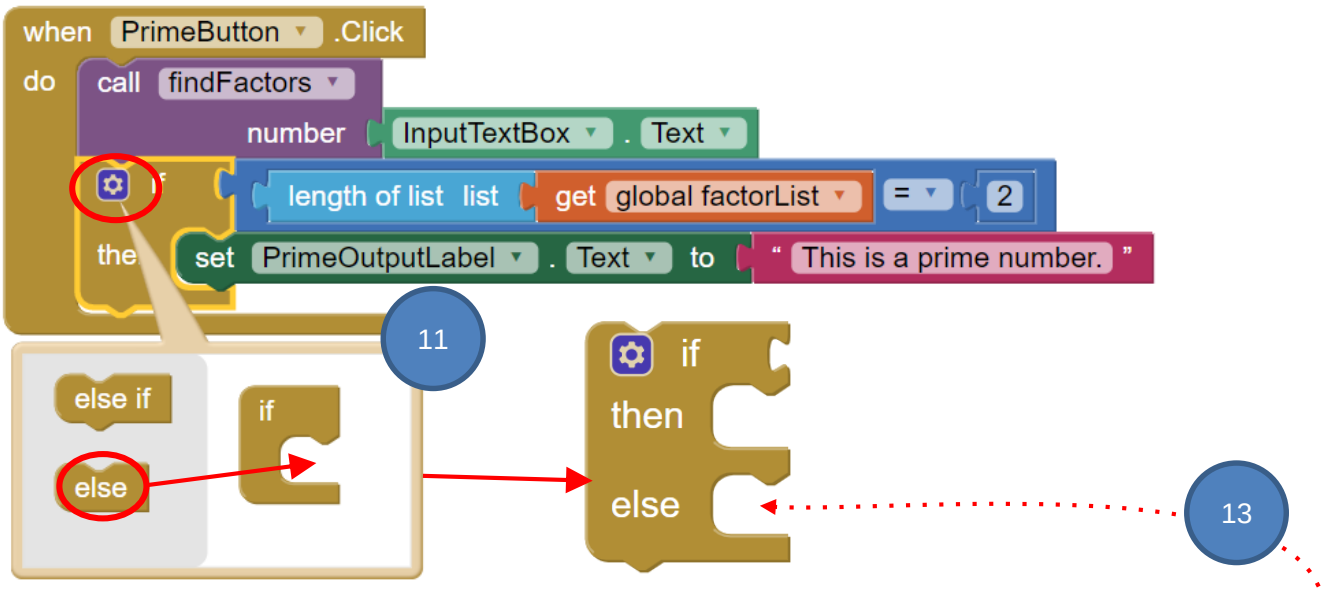
CT Tips

Use **conditionals** to determine if a number is prime or composite.

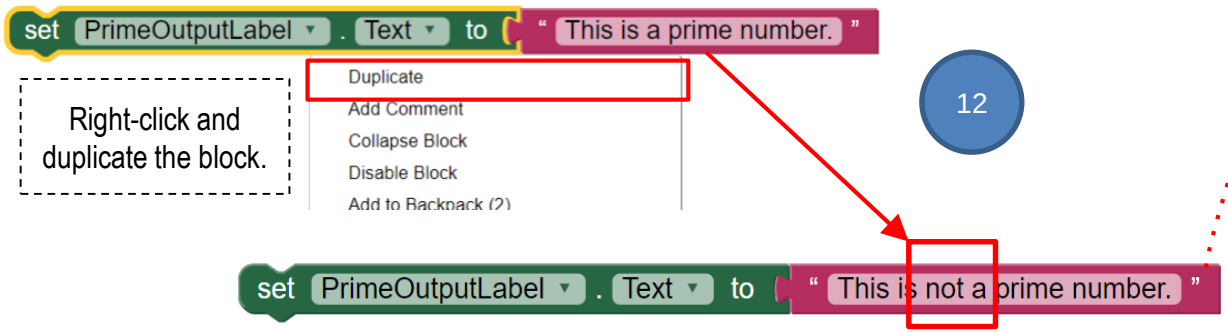
FACTOR APP

PRIME BUTTON (CONT'D)

❑ Click the blue gear of **if-then**, and drag the **else** block into the **if** to create an **if-then-else** block.



❑ Duplicate the **setPrimeOutputLabel.Text** block, update the text to "**This is not a prime number.**" and snap it into the "else" slot.

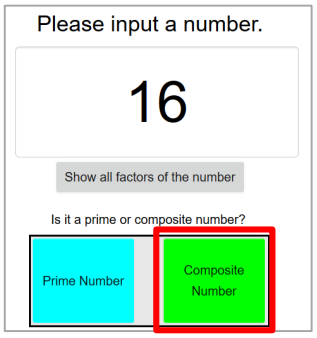


❑ Well done! We just completed the coding for the **PrimeButton**.

❑ Now, do you know how to code the **CompositeButton**?

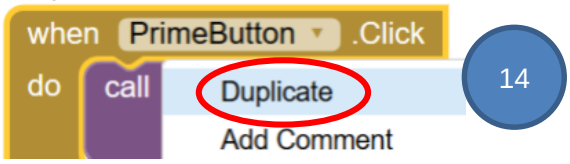
FACTOR APP

COMPOSITE BUTTON

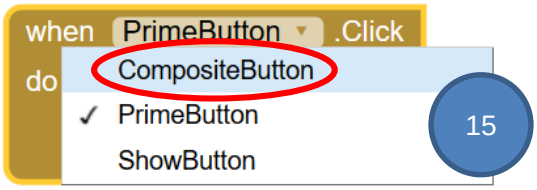


If the user clicks on the **CompositeButton**, you will need to check if the number of factors is greater than 2.

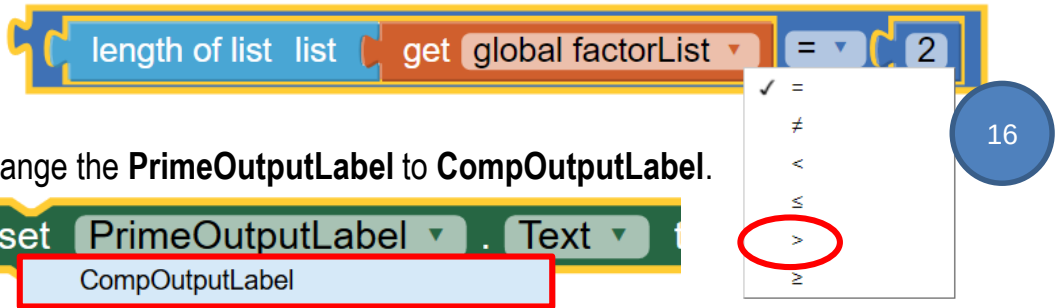
- ❑ Right-click on **PrimeButton.Click** and duplicate the whole block.



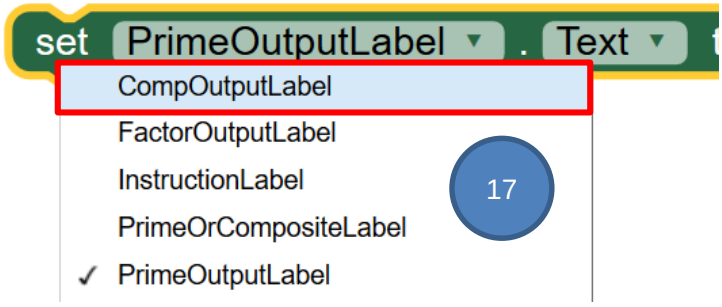
- ❑ Change **PrimeButton** to **CompositeButton**.



- ❑ Change the **equals** (=) operator to **greater than** (>) operation in the **if** block.



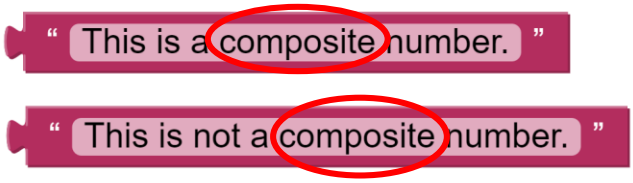
- ❑ Change the **PrimeOutputLabel** to **CompOutputLabel**.



Update "prime" to "composite".

18

- ❑ Update the text block in the **if-then-else** block to "**This is a composite number**" and "**This is not a composite number**" respectively.



TESTING & DEBUGGING

19

- ❑ Test using the MIT AI2 Companion.

  Type in numbers you know are prime, like 7 and 13. Do you get the correct response when you press the Prime and Composite buttons?

  Try some composite numbers. Are your results correct?

  What happens when you type 1 and 0?

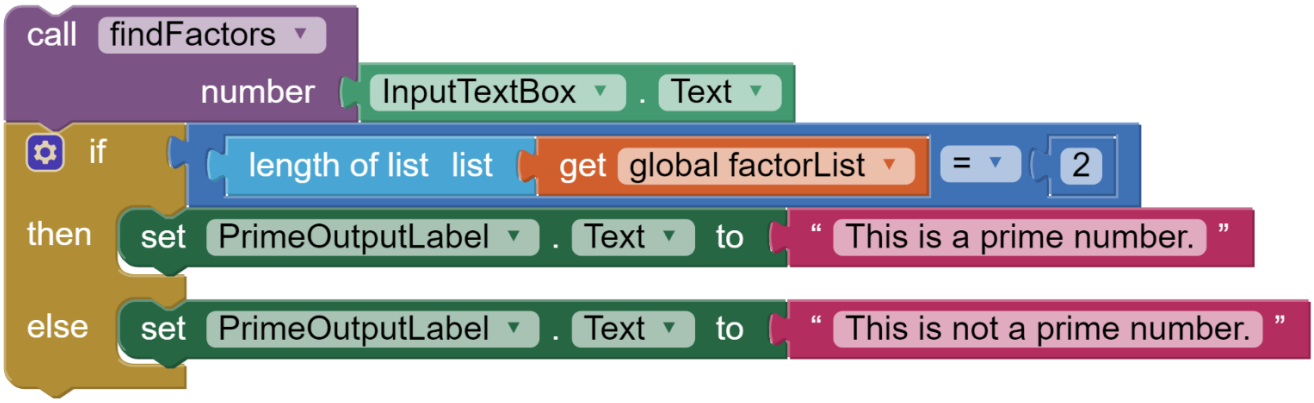
FACTOR APP

COMPUTATIONAL THINKING CONCEPTS

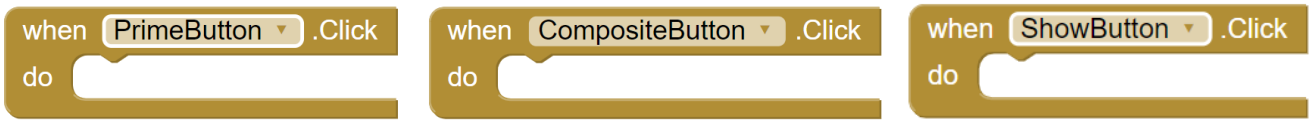
The following are the Computational Thinking Concepts learnt in Lesson 4.

L3U8.4-8.5 Factor App

1. Sequences



2. Events



3. Conditionals

4. Operators

FACTOR APP

COMPUTATIONAL THINKING PRACTICES

The following are the Computational Thinking Practices used in this Unit.

L3U8.4-8.5 Factor App
1. Reusing and remixing: a) Students duplicate and modify the PrimeButton.Click event block to complete the CompositeButton.Click block.
2. Being incremental and iterative: a) Students create the app in stages.
3. Abstracting and modularizing: a) Students write findFactors procedure to find the factors of a number.
4. Testing and debugging: a) Students test the app to make sure the app works.
5. Algorithmic thinking: a) Students learn the concepts of factor, prime number and composite number. b) Students apply an algorithm to find factors of a number in their code.