Creating Algorithms

Use this resource to reflect on your practice or identify opportunities to integrate algorithms in your classroom. This resource was developed to support middle school science teachers to integrate computational thinking into NGSS-aligned lessons. The content can be adapted for other content areas, grade bands and contexts.

What is an algorithm?

A repeatable process that delivers an expected result.

Look fors:

Students may be creating algorithms when they are:

- ❑ Decomposing problems or tasks
- ❑ Identifying essential steps
- ❑ Testing and debugging
- ❑ Considering efficiency

Prompting questions:

Ask students to reflect on their process or progress with these prompting questions:

- ❑ What do you need to know to be able to solve this problem/do this task?
- ❑ What should be the result of the problem or task?
- ❑ Why is each step required to solve the problem/task?
- ❑ Would you include additional words or details to explain this process to a partner?
- ❑ Does each step have the result you intend it to?
- ❑ Does a partner testing your algorithm get the same results as you?
- ❑ Are there certain inputs where you do not get the intended result?
- ❑ Can your algorithm have the same result with less steps?
- ❑ Do you notice any patterns in your procedure?

Activities & Tools

Activities that may integrate algorithms into science	You might use this when...	Template Student resources to use or modify to integrate algorithms in your classroom	Example An example of a student resource applied to a science topic
Sort and categorize relationships between objects or processes	Students are identifying distinguishing qualities about items in order to classify them	Template: Creating algorithms to identify, sort, and categorize	Example: Creating algorithms to identify plants
Develop protocols to troubleshoot a question or design problem	Students have knowledge about a concept, process or tool and want to help others troubleshoot on their own	Template: Creating algorithms to troubleshoot a design problem	Example: Creating algorithms to troubleshoot a circuit
Code a computational tool to illustrate a scientific phenomenon	Students are learning about something that is difficult to understand because of time, visibility, or size	Template: Creating algorithms to illustrate scientific phenomenon	Example: Creating algorithms to illustrate a food web in Scratch

Creating Algorithms: Identify, Sort & Categorize Objects

An algorithm is a repeatable process that delivers an expected result. One way to use an algorithm is to categorize relationships and characteristics of objects in order to identify them. In the example below, a student collected plants at a park over a year. They created an algorithm to help others identify plants they find in the same area by considering what they look like and when and where they were collected.

1 Part 1: Identify Characteristics

Collect the object you want to identify. What do you know about them? Create a data table, list, or chart to help you and similarities and differences.

Checker-bloom
Pink with dark green leaves
Grows March-May
Found in grasslands

California Poppy
Orange
Grows March-November
Found in grasslands

Crimson Columbine
Red with yellow center
Grows March-May
Found in rocky north facing slopes

Western Hound's Tongue
Blue with a white center
Grows January-March
Found in rocky north facing slopes

Blue Eyed Grass
Violet with a yellow center
Grows March-May
Found in grasslands

Bush Moneyflower
Orange
Grows March-November
Found in scrub

Wild Buckwheat
Pink
Grows June-November
Found in rocky north facing slopes

How would you help someone filter through all of the objects to identify a single one?

Use the following steps to get started:

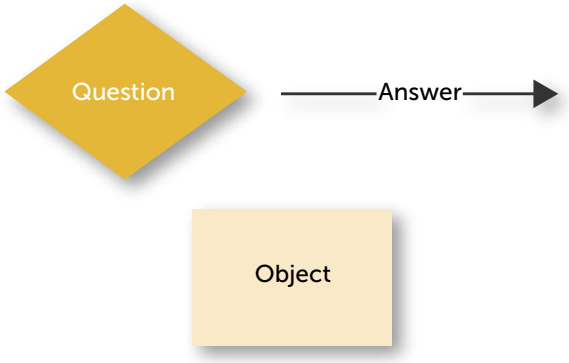
1. Identify a single characteristic that sorts all of the objects into categories of approximately equal size.
2. In each category, identify an another characteristic that sorts the objects into additional subgroups.
3. Repeat until each object is in a single category.

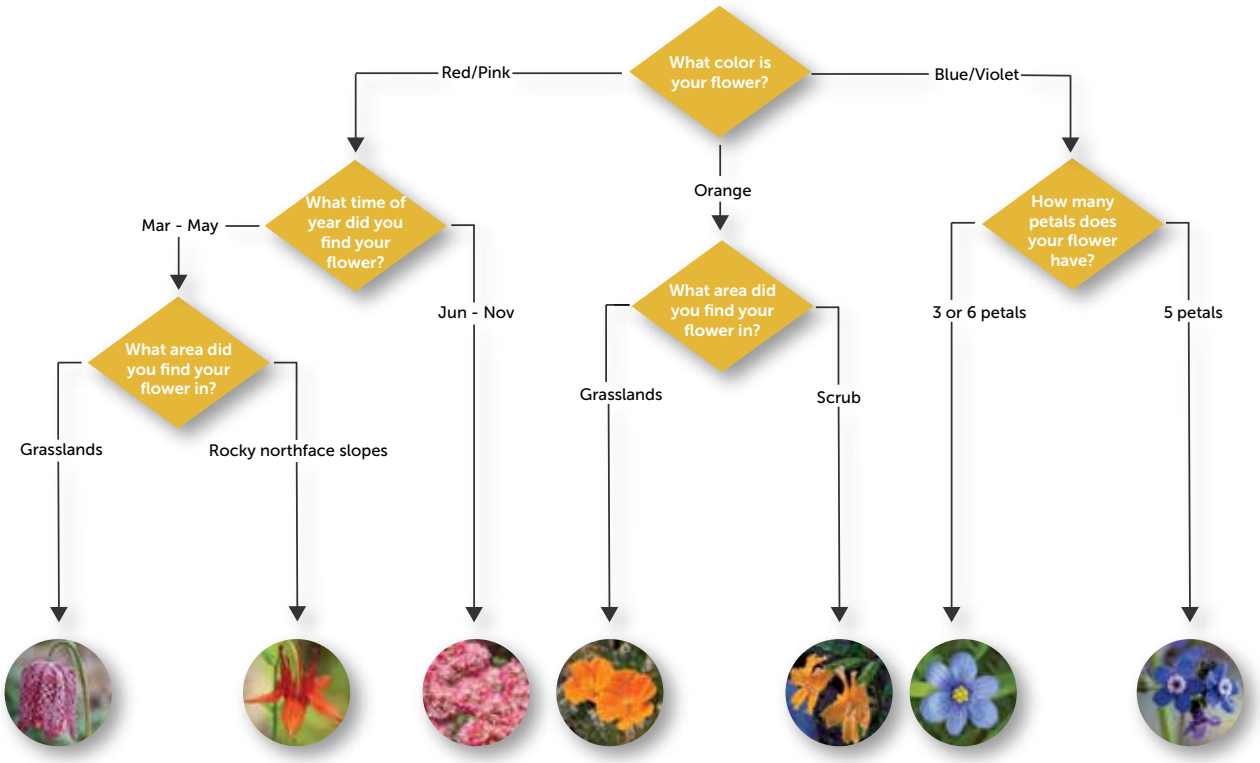
List the characteristics you used to sort each group below.

1. Color
2. Time of year
3. Location
4. Number of petals

2 Part 2: Develop Your Algorithm

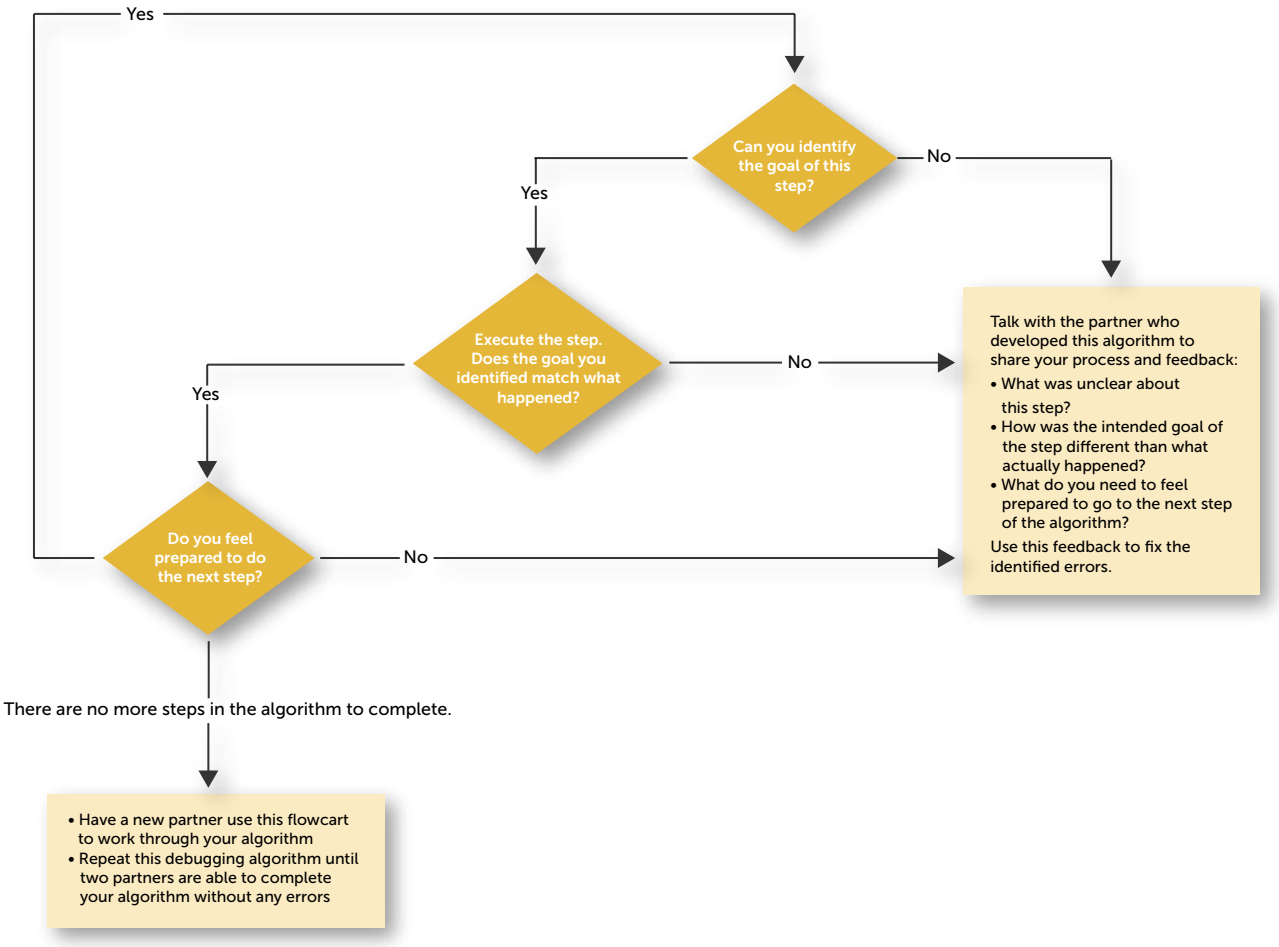
Create an algorithm to help someone filter through all of the objects to identify a single one. Use the key below to use the same shapes as the example in your algorithm and create new shapes to show other types of steps. You can also create your own shapes to draft your algorithm on a computer using a tool such as LucidChart, Smartdraw, or Draw.io.





3 Part 3: Pair Debugging Algorithm

While completing your algorithm, work with a partner to debug -- which is to find and fix errors -- and improve it:



References:

Wild Flowers of Golden Gate
California Plant Finder
Calscape

Creating Algorithms: Identify, Sort & Categorize Objects

An algorithm is a repeatable process that delivers an expected result. One way to use an algorithm is to categorize relationships and characteristics of objects in order to identify them.

1 Part 1: Identify Characteristics

Collect the object you want to identify. What do you know about them? Create a data table, list, or chart to help you find similarities and differences. How would you help someone filter through all of the objects to identify a single one?
First, identify a single characteristic that sorts all of the objects into categories of approximately equal size. Then, identify a characteristic in each subcategory that sorts the objects into additional subgroups. Repeat until each object is in a single category. List the characteristics you used to sort each group below.

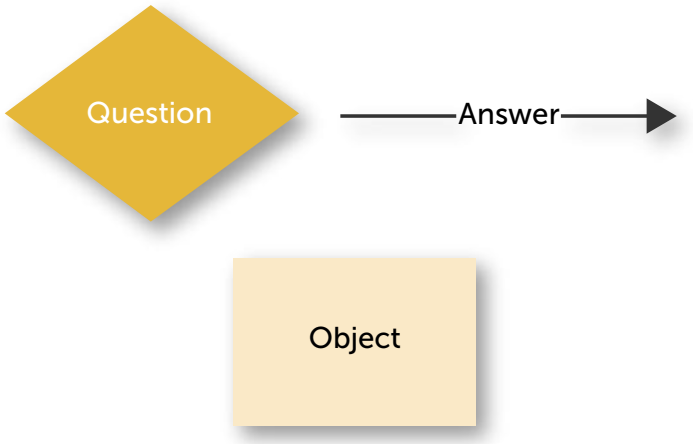
1. _____

2. _____

3. _____

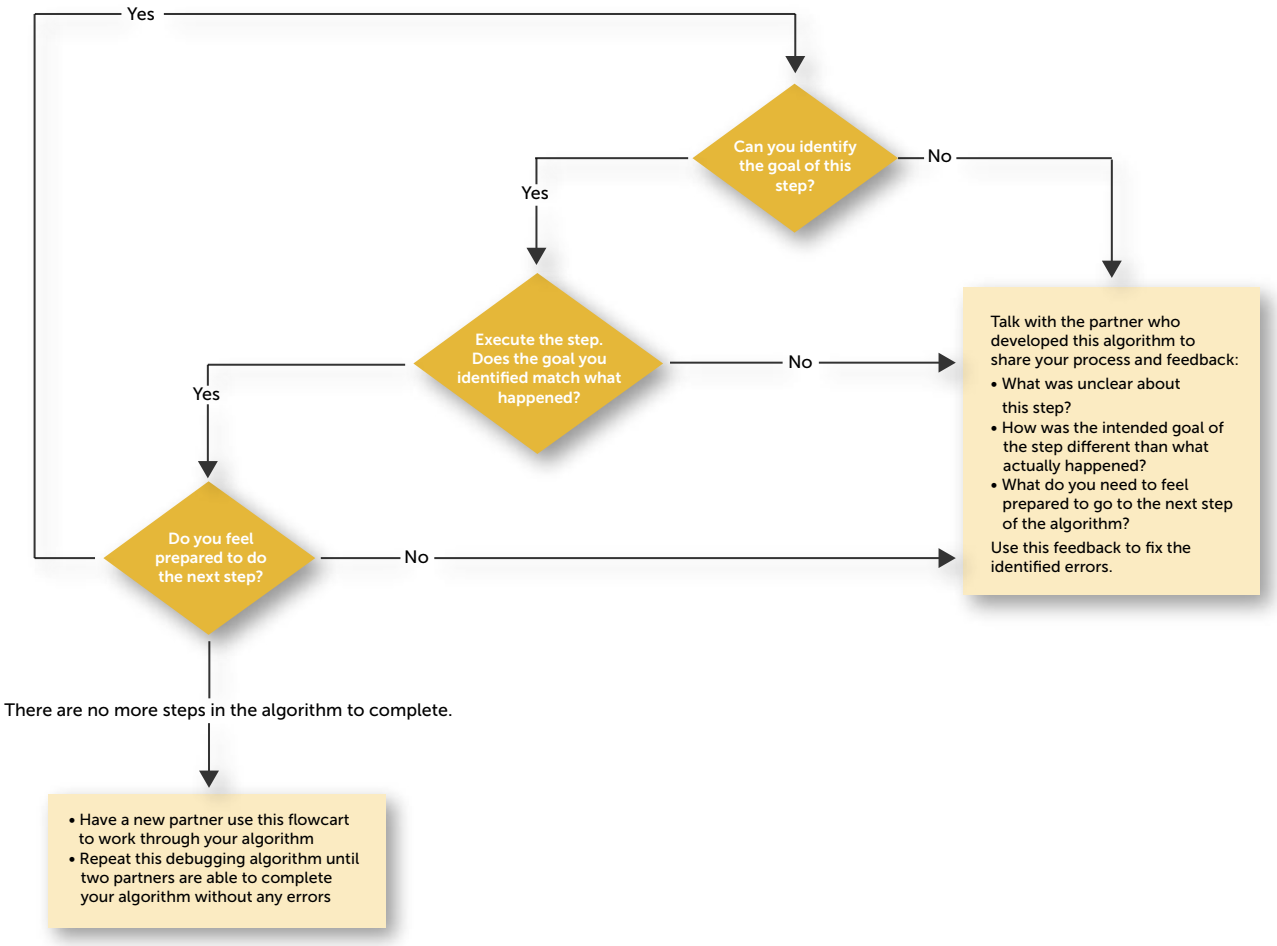
2 Part 2: Develop Your Algorithm

Create an algorithm to help someone filter through all of the objects to identify a single one. Use the key below to use the same shapes as the example in your algorithm and create new shapes to show other types of steps. You can also create your own shapes to draft your algorithm on a computer using a tool such as LucidChart, Smartdraw, or Draw.io.



3 Part 3: Pair Debugging Algorithm

While completing your algorithm, work with a partner to debug -- which is to find and fix errors -- and improve it:

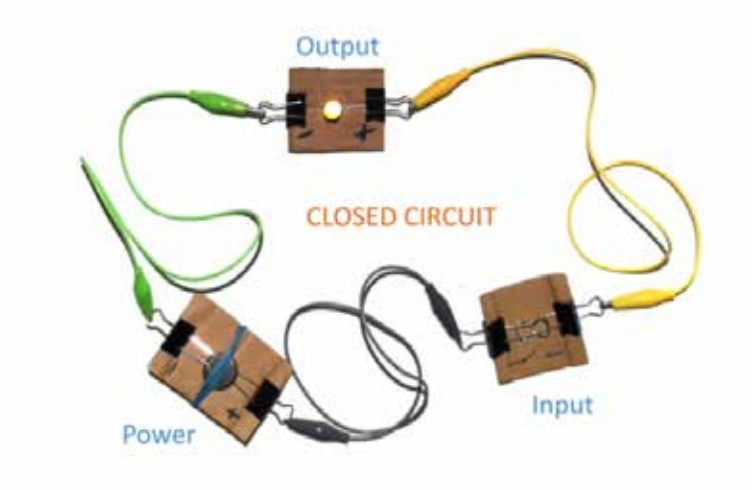


Creating Algorithms: Troubleshoot a Design Problem

An algorithm is a repeatable process that delivers an expected result. An engineer might use an algorithm to help others troubleshoot common design problems. In the example below, students completed the circuit arcade activity (https://makerpromise.org/circuit-arcade/). Then, they created an algorithm to help others troubleshoot why a lightbulb in a circuit may not be lighting up.

1 Part 1: Describe Your Design

Sketch and **label** your design here:

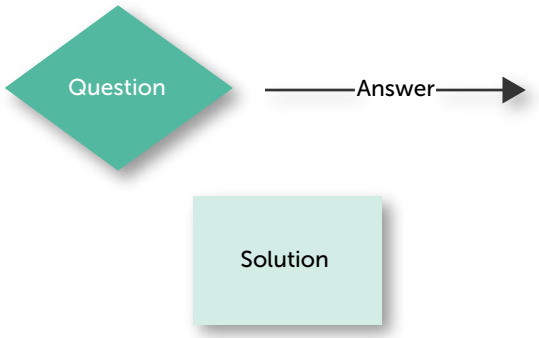


Think about the **parts** of your design and the **purpose** of each part (what it does and what makes it important to the overall function). Then, **troubleshoot** to identify why this part might not work.

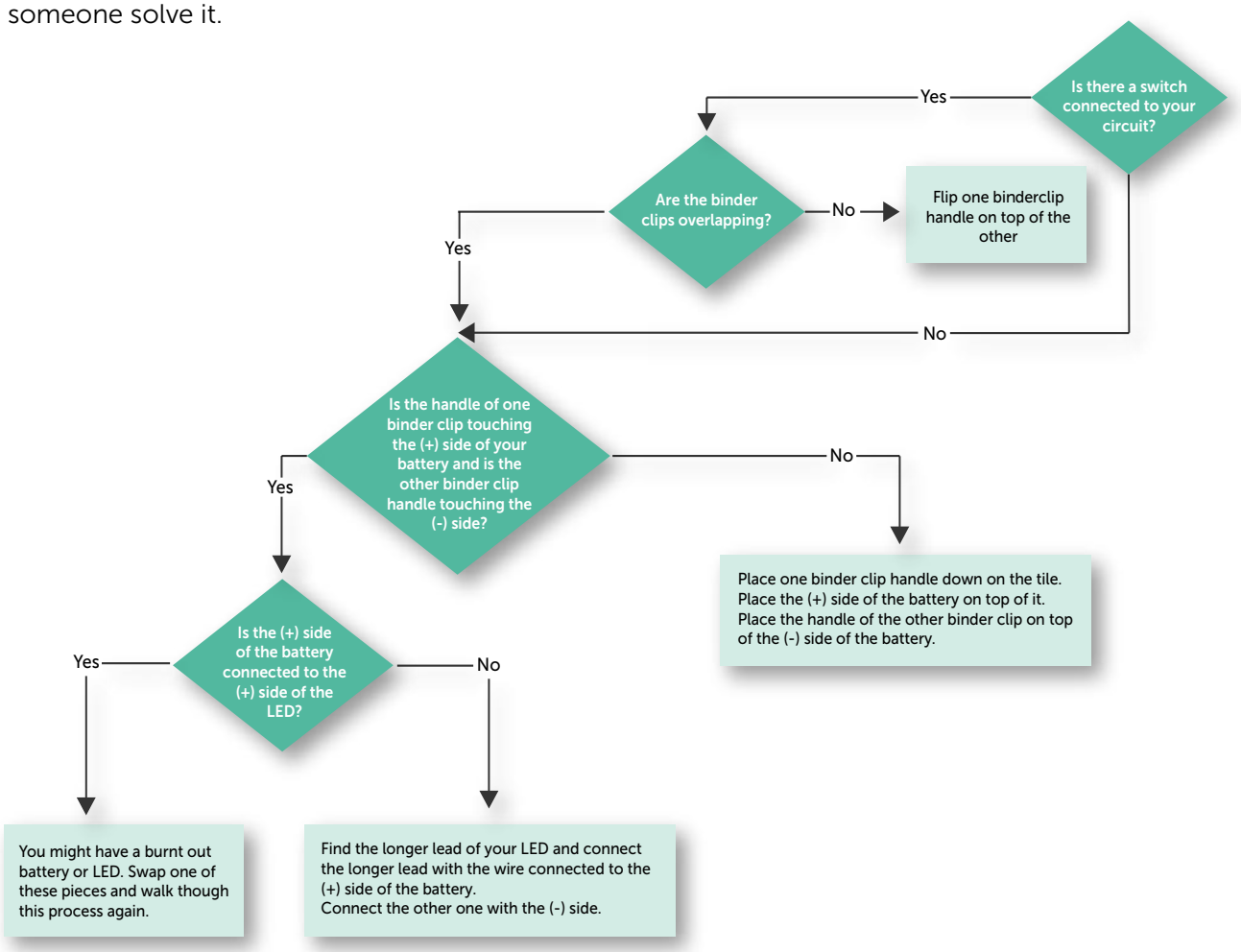
Part	Purpose	Troubleshoot
Output (LED)	The light (LED) turns on, which tells me that my circuit is closed and working.	The switch isn't closed. The LED isn't connected correctly (- to - and + to +). The LED burned out.
Power (Battery)	The battery provides the power needed to turn the light on.	The binder clips handles aren't touching both the (+) or (-) side of the battery. The battery burned out.
Input (Switch)	The switch allows my circuit to open and close.	The binder clip handles aren't overlapping each other.

2 Part 2: Draft Your Algorithm

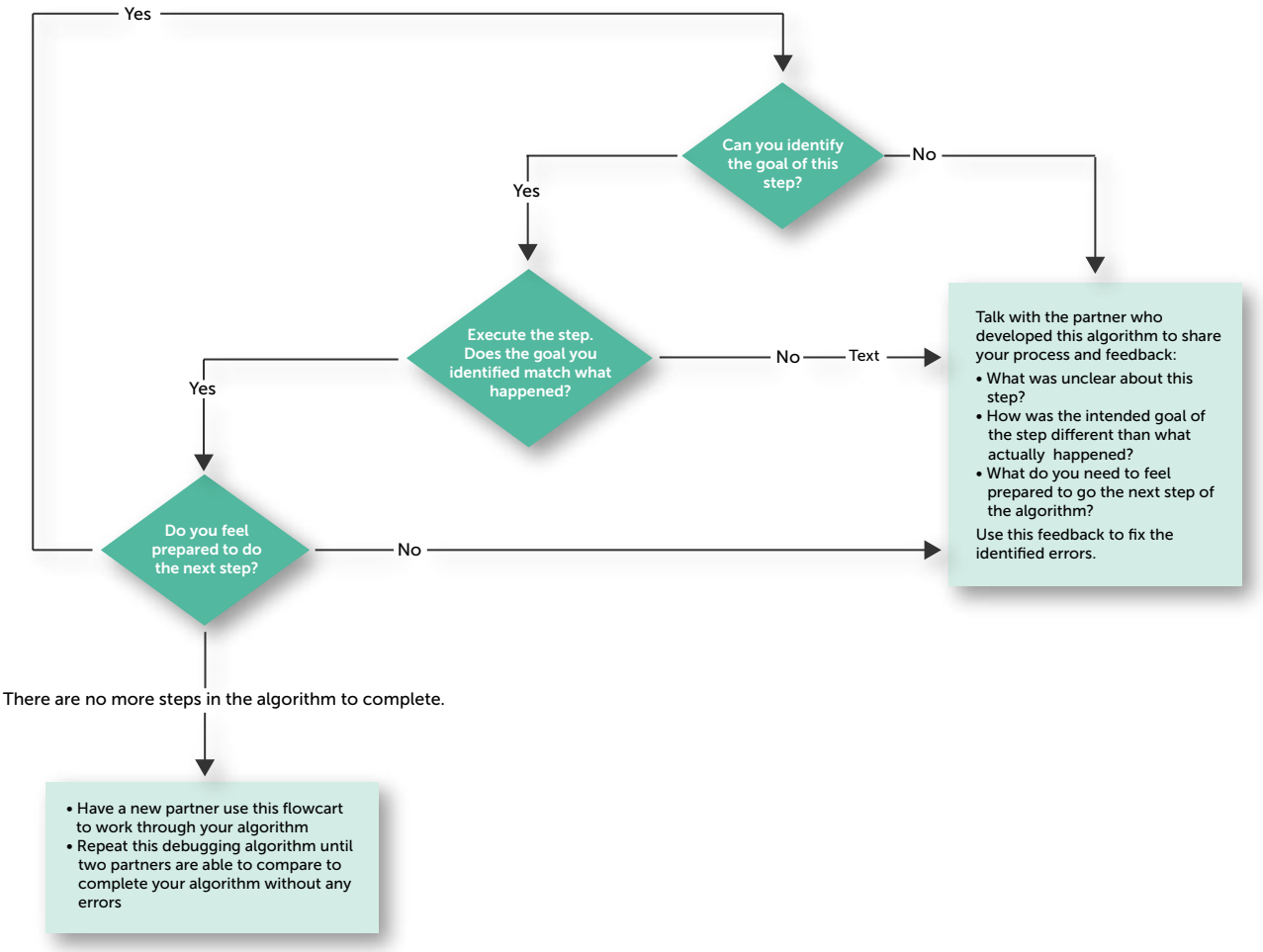
In this assignment, you are going to create an algorithm to solve a design problem. In the example, there are different shapes that represent different types of steps. Use the key below to use the same shapes in your algorithm and create new shapes to show other types of steps. You can also create your own shapes to draft your algorithm on a computer using a tool such as LucidChart, Smartdraw, or Draw.io.



Start by writing your problem below, and then add clarifying questions and solutions to help someone solve it.



3 Part 3: Pair Debugging Algorithm



Creating Algorithms: Troubleshoot a Design Problem

An algorithm is a repeatable process that delivers an expected result. An engineer might use an algorithm to help others troubleshoot common design problems.

1 Part 1: Describe Your Design

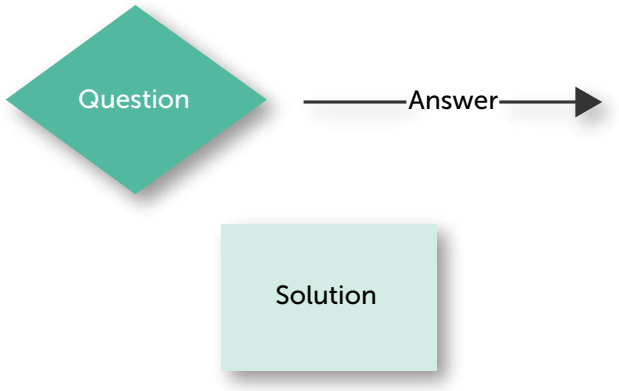
Sketch and **label** your design here:

Think about the **parts** of your design and the **purpose** of each part (what it does and what makes it important to the overall function). Then, **troubleshoot** to identify why this part might not work.

Part	Purpose	Troubleshoot

2 Part 2: Draft Your Algorithm

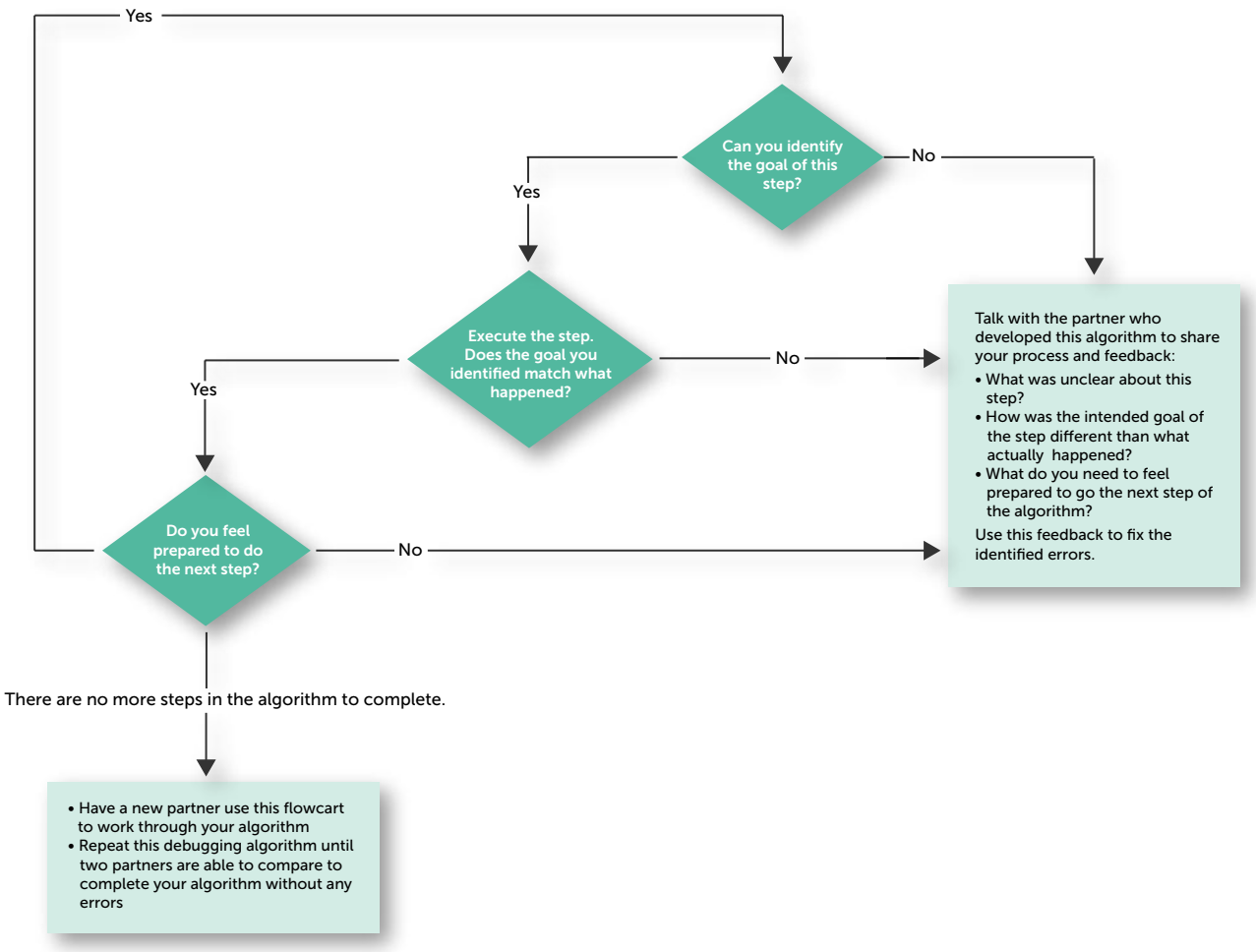
In this assignment, you are going to create an algorithm to solve a design problem. In the example, there are different shapes that represent different types of steps. Use the key below to use the same shapes in your algorithm and create new shapes to show other types of steps. You can also create your own shapes to draft your algorithm on a computer using a tool such as LucidChart, Smartdraw, or Draw.io.



Start by writing your problem below, and then add clarifying questions and solutions to help someone solve it.

3 Part 3: Pair Debugging Algorithm

After completing your algorithm, work with a partner to debug -- which is to find and fix errors -- and improve it:

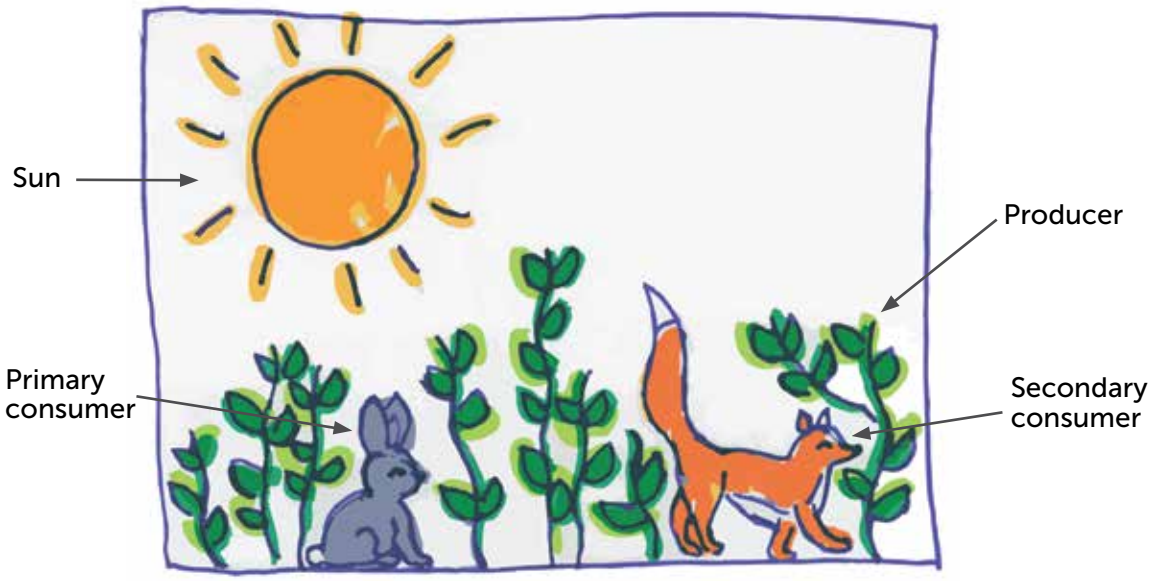


Creating Algorithms: Illustrate a Phenomenon

An algorithm is a repeatable process that delivers an expected result. In this assignment, you are going to create an algorithm to illustrate a scientific phenomenon. There are many predictable events that occur in the natural world, such as life cycles, chemical reactions, or Earth's processes. One way to illustrate these phenomena is by programming an algorithm. If your algorithm represents a mathematical relationship between parts of system, it may be a computational model. If there are no input/outputs or defined mathematical relationships, you are likely simply illustrating a scientific phenomenon. See resources for Creating Computational Models to learn more about how your illustration could be modified to become a computational model. In the example below, a student illustrated a simple food chain with a producer, herbivore and carnivore by programming in Scratch.

1 Part 1: Describe the Phenomenon

Sketch and **label** the phenomenon you will illustrate here:



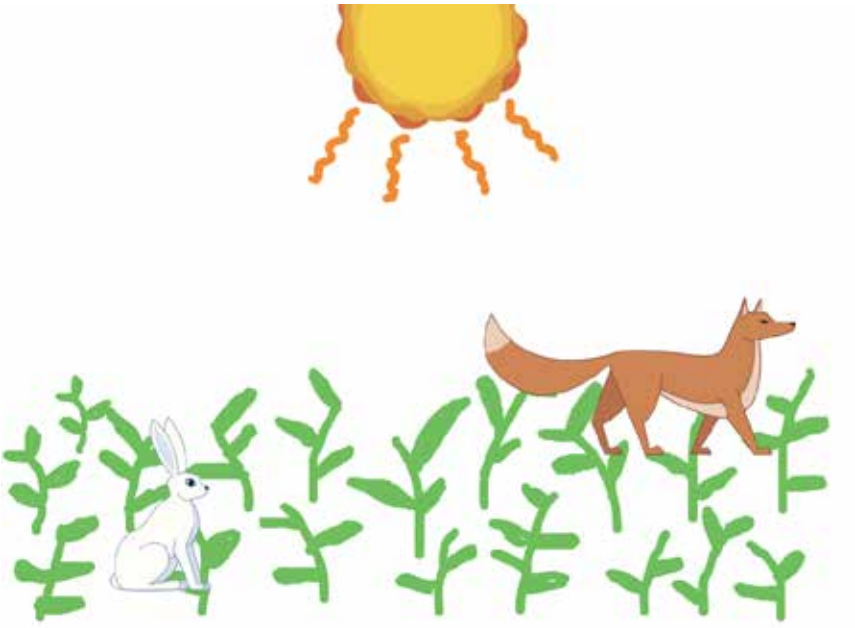
Think about the **parts** of the phenomenon you will illustrate, the **purpose** of each part, and if the part will perform an action in your program.

Part	Purpose	Action (yes/no). If yes, describe.
Sun	Provides energy	Appear and disappear on a timer to signal night and day
Producer	Converts sun's energy into food through photosynthesis	Grow/reproduce when exposed to the sun
Primary consumer (herbivore)	Eats producers	If herbivore is pushed, it eats the producer
Secondary consumer (carnivore)	Eats consumers	If carnivore is pushed, it eats the herbivore

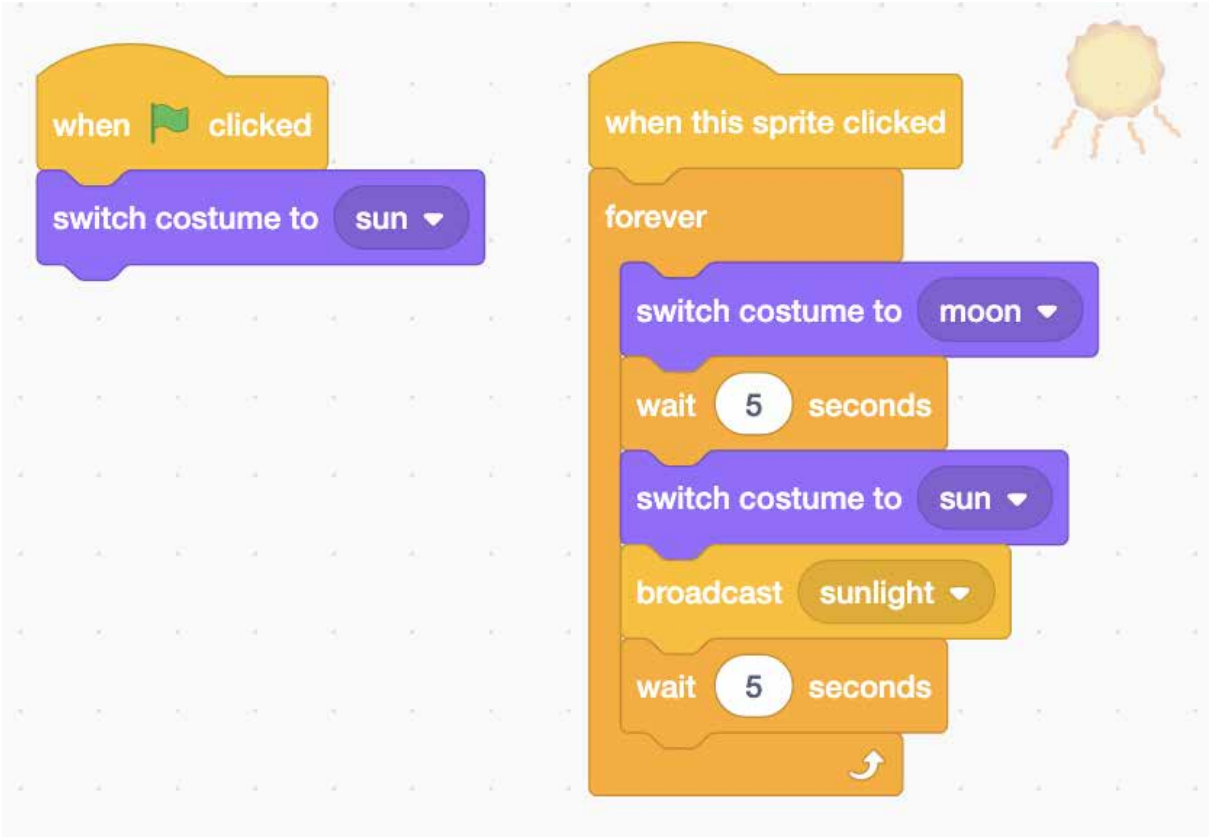
2 Part 2: Create Your Program

Now you will use a computational tool to create your algorithm. There are many tools available to program algorithms, such as coding platforms (e.g., Scratch, Snap, MakeCode) or computational making kits (e.g., Hummingbird Robots, Micro:bit, LegoWedo, Arduino, Raspberry Pi). Your teacher will tell you which tool(s) you may use for this assignment.

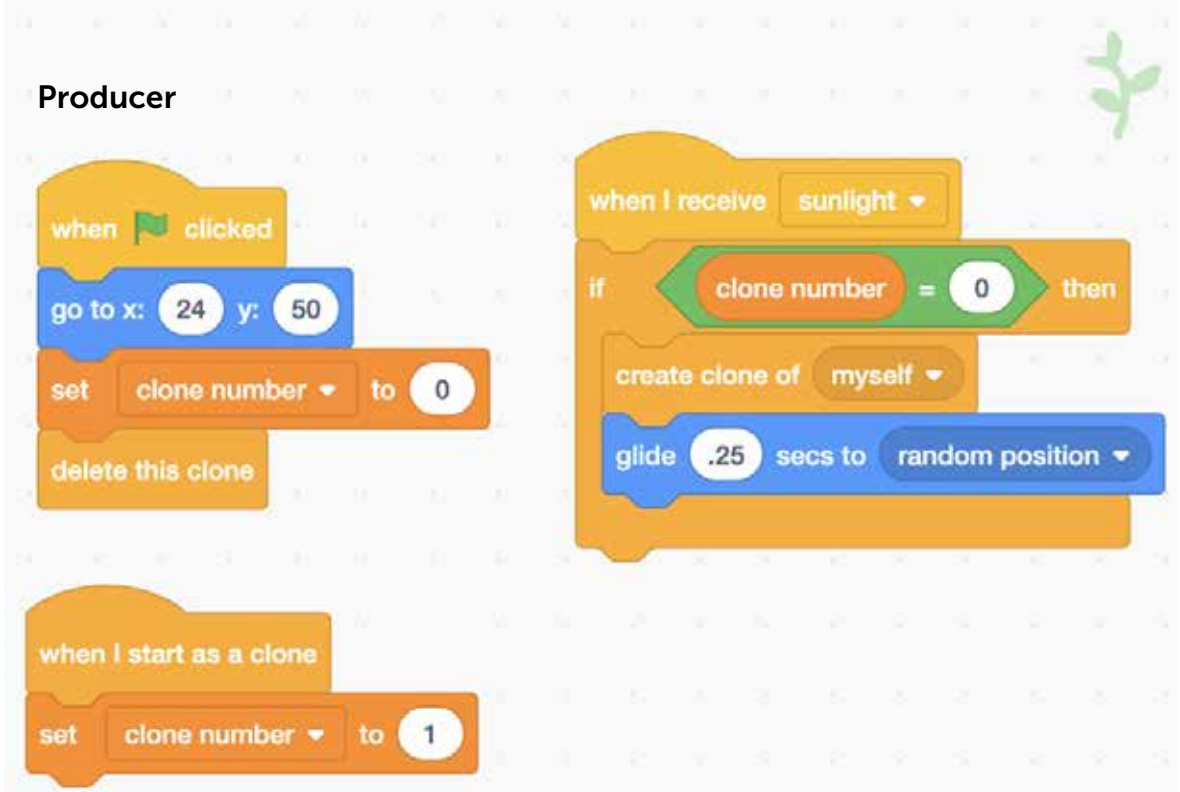
Link to Scratch project here



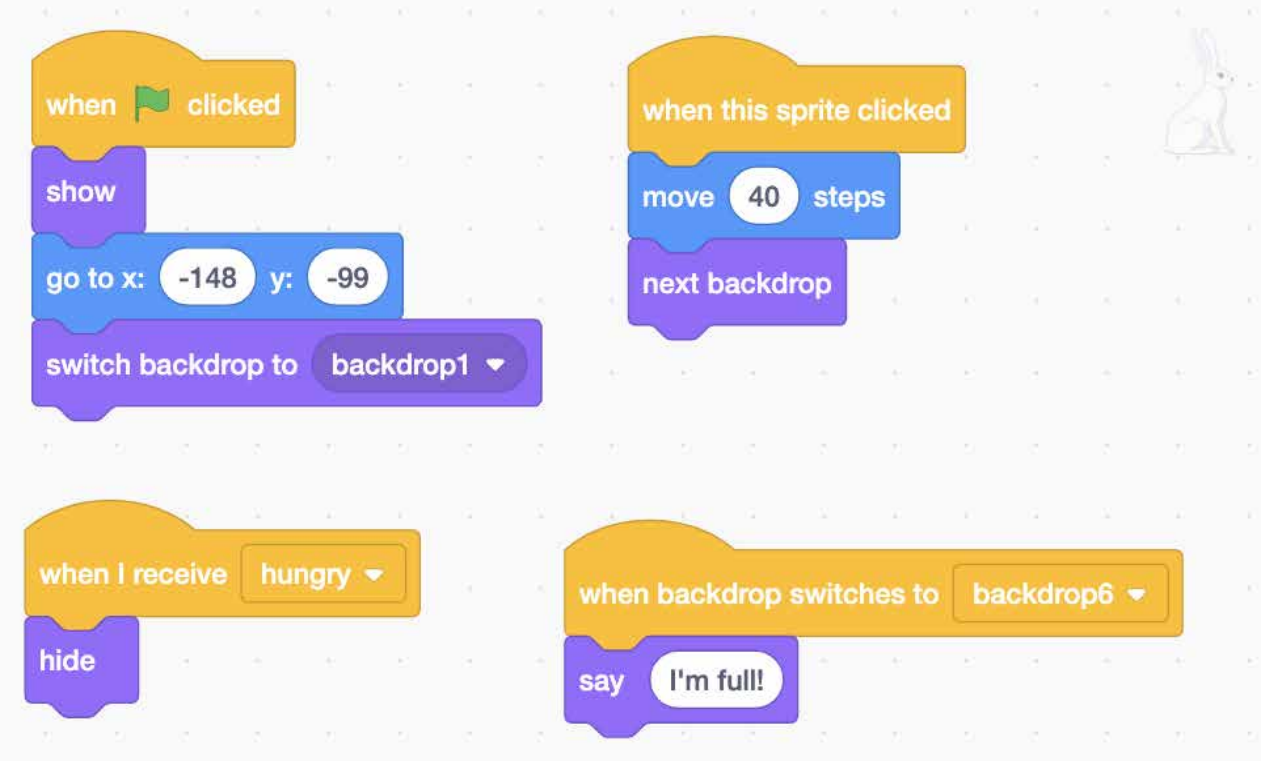
Sun



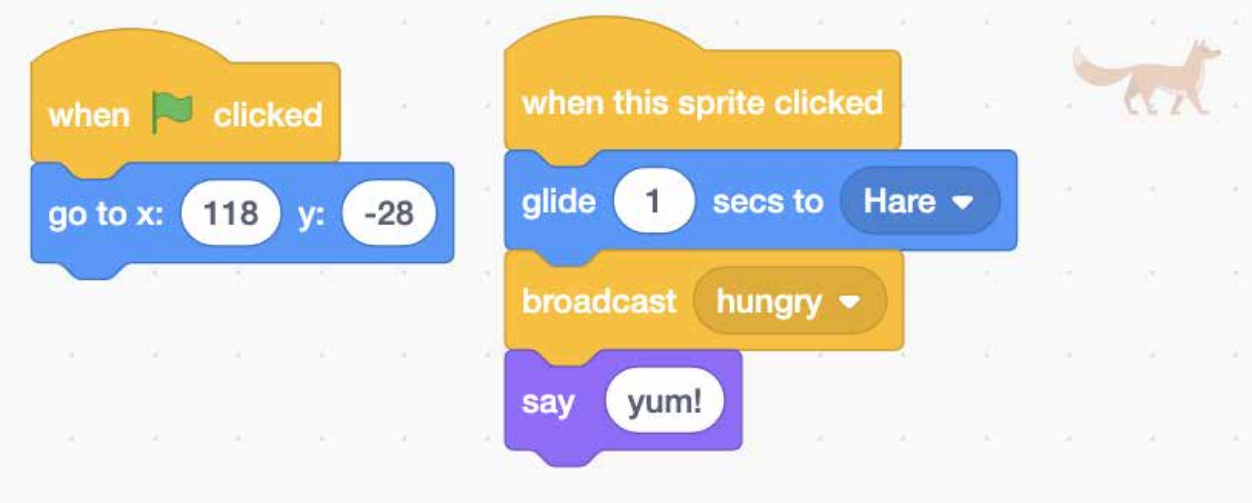
Producer



Primary consumer (herbivore)

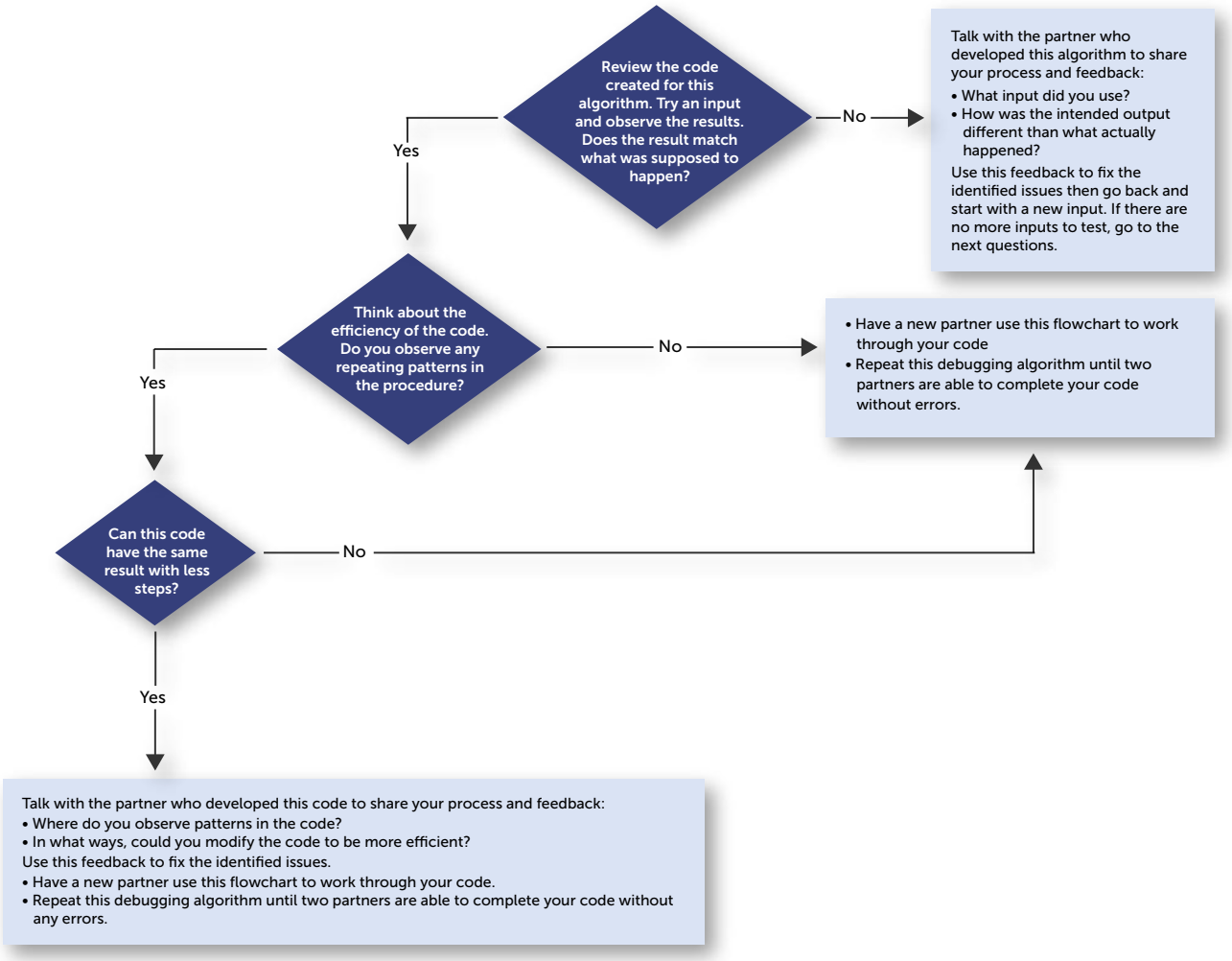


Secondary consumer (carnivore)



3 Part 3: Pair Debugging Algorithm

While completing your program, work with a partner to debug -- which is to find and fix errors -- and improve it:



Creating Algorithms: Illustrate a Phenomenon

An algorithm is a repeatable process that delivers an expected result. In this assignment, you are going to create an algorithm to illustrate a scientific phenomenon. There are many predictable events that occur in the natural world, such as life cycles, chemical reactions, or Earth's processes. One way to illustrate these phenomena is by programming an algorithm. If your algorithm represents a mathematical relationship between parts of system, it may be a computational model. If there are no input/outputs or defined mathematical relationships, you are likely simply illustrating a scientific phenomenon. See resources for Creating Computational Models to learn more about how your illustration could be modified to become a computational model.

1 Part 1: Describe the Phenomenon

Sketch and **label** the phenomenon you will illustrate here:

Think about the **parts** of the phenomenon you will illustrate, the **purpose** of each part, and if the part will perform an action in your program..

Part	Purpose	Action (yes/no). If yes, describe.

2 Part 2: Create Your Program

Now you will use a computational tool to create your algorithm. There are many tools available to program algorithms, such as coding platforms (e.g., Scratch, Snap, MakeCode) or computational making kits (e.g., Hummingbird Robots, Micro:bit, LegoWedo, Arduino, Raspberry Pi). Your teacher will tell you which tool(s) you may use for this assignment.

3 Part 3: Pair Debugging Algorithm

While completing your program, work with a partner to debug -- which is to find and fix errors -- and improve it: