

# Project Justification

After you complete the graphic organizer below, use this project justification document to explain how you used computational thinking in your project.

**Problem Identification.** For each iteration of your problem, please explain how you arrived at your identified problem.

**Decomposition.** For each iteration where you decomposed an identified problem, please explain how this decomposition helped you solve your identified problem.

**Pattern Recognition.** For each iteration where you recognized patterns in data, please explain how these patterns helped you solve your identified problem.

**Abstraction.** For each iteration where you abstracted information, please explain how abstraction allowed you to solve your identified problem.

# Iteration 1

Problem Identification

To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 2

Problem Identification

To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

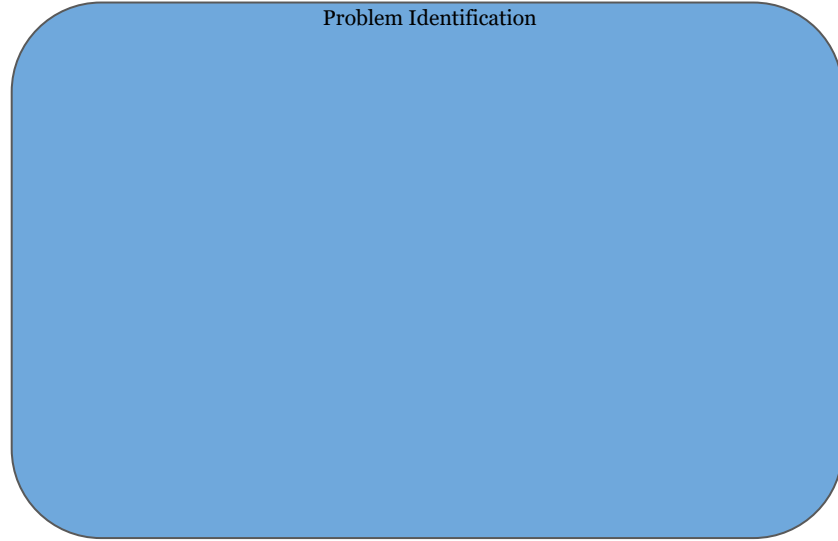
Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

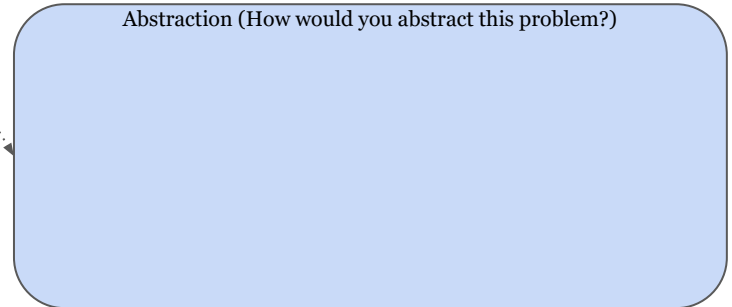
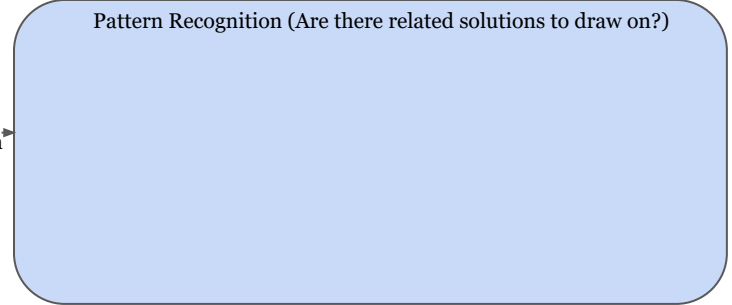
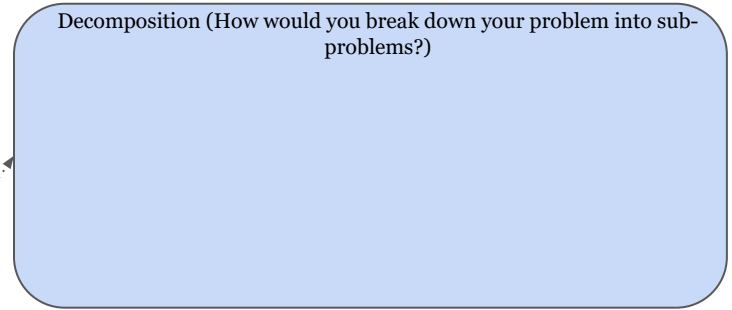
## Graphic Organizer



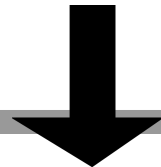
# Iteration 3



To set up your  
identified problem



## Graphic Organizer



# Iteration 4

Problem Identification

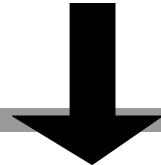
To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

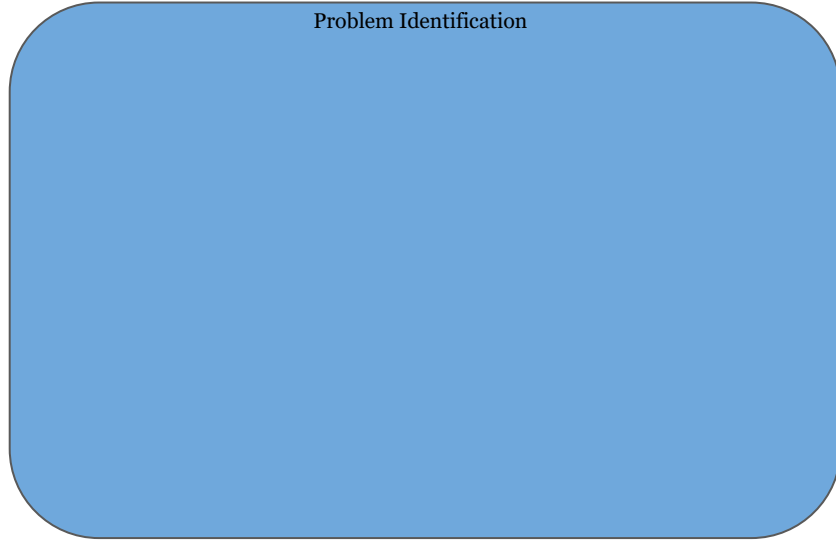
Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 5



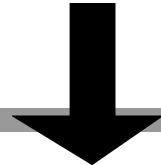
To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 6

Problem Identification

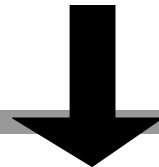
To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

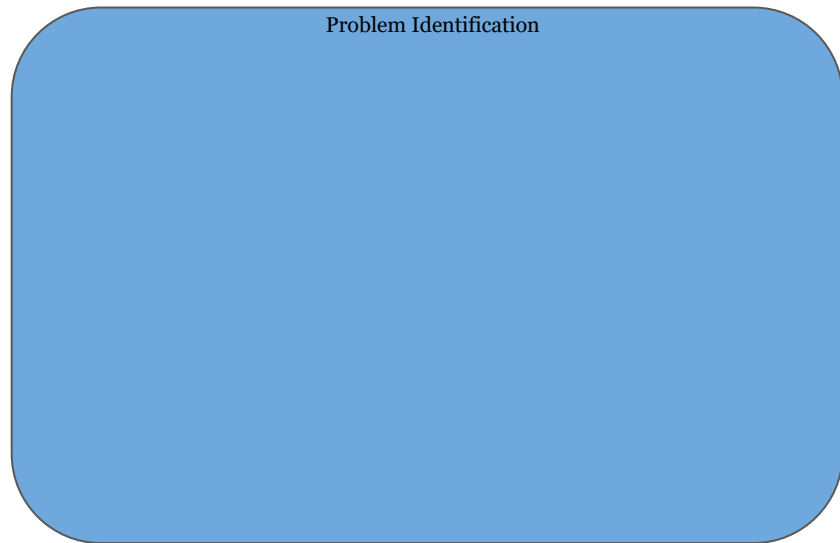
Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 7



To set up your  
identified problem

Two dotted arrows originate from the right side of the 'Problem Identification' box. One arrow points to the 'Decomposition' box, and the other points to the 'Abstraction' box. A third dotted arrow points from the 'Pattern Recognition' box back to the 'Problem Identification' box.

Decomposition (How would you break down your problem into sub-problems?)

A light blue rounded rectangle with a thin black border, intended for decomposing the problem into sub-problems.

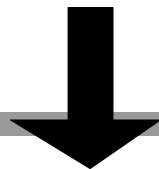
Pattern Recognition (Are there related solutions to draw on?)

A light blue rounded rectangle with a thin black border, intended for recognizing patterns or related solutions.

Abstraction (How would you abstract this problem?)

A light blue rounded rectangle with a thin black border, intended for abstracting the problem.

## Graphic Organizer





# Iteration 8

Problem Identification

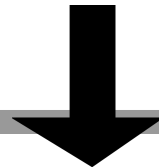
To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

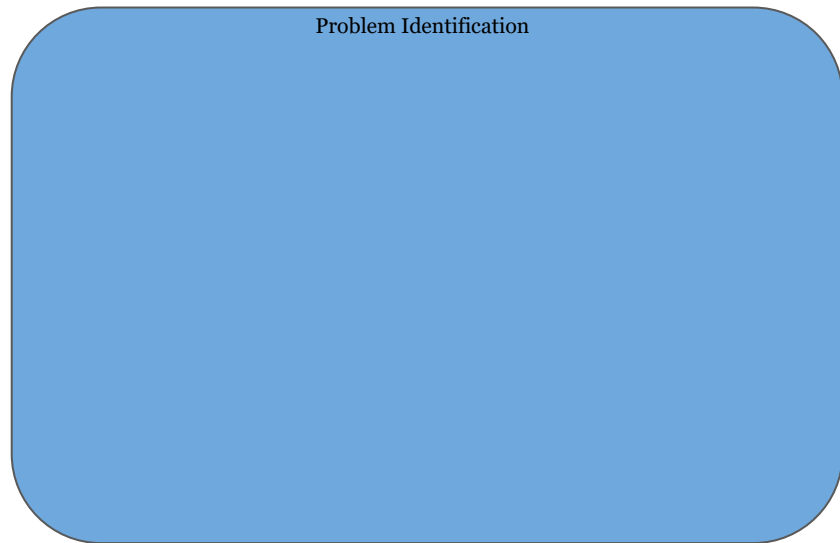
Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 9



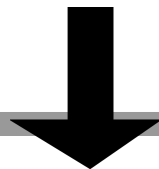
To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer



# Iteration 10



To set up your  
identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

## Graphic Organizer