

Name(s) _____ Period _____ Date _____

Activity Guide - Writing Algorithms



Algorithms and the Problem-Solving Process

An **algorithm** is a finite set of instructions that accomplish a task.

The **Problem Solving Process** is useful when planning and writing algorithms. This process will help you clarify and break down a problem into manageable steps so you can easily identify the code you need to write for each step.

Define

- Read the instructions carefully to make sure you understand the goals.
- Rephrase the problem in your own words.
- Identify any new skills you are being asked to apply.
- If there is starter code, read it to understand what it does.

Prepare

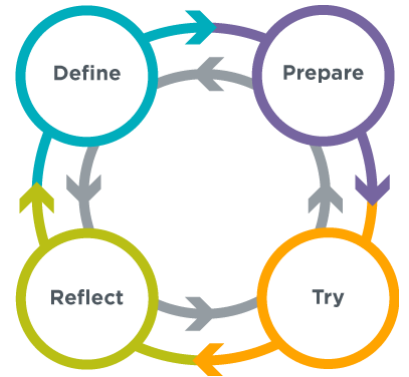
- Write out or draw the steps you need to take to solve the problem.
- List what you already know how to do and what you don't yet.
- Explain your algorithm to a classmate.
- Review similar programs that you've written in the past.

Try

- Write one small piece at a time.
- Test your program often.
- Use comments to document what your code does.
- Go back to a previous step if you get stuck or don't know whether you've solved the problem.

Reflect

- Compare your program to the defined problem to make sure you've solved all aspects of the problem.
- Ask a classmate to try your program and note places where they struggle or show confusion.
- Ask a classmate to read your code to make sure that your documentation is clear and accurate.
- Try to "break" your program to find types of interaction or input that you could handle better.
- Identify changes or improvements you can make next to your program.



Algorithm Planning

For your chosen problem, apply the Problem Solving Process to make sure you understand its requirements, identify similar problems, and plan and refine your algorithm.

Define

Read the instructions carefully to ensure you understand the goals. Rephrase the problem in your own words.

Prepare

Write out the steps you need to take to solve the problem.

Questions to Consider:

- Which steps do you already know how to do? Which don't you know how to do yet?
- Explain your steps to a classmate. Was there anything your classmate found confusing? Did they have suggestions that could help improve your algorithm?
- Look at similar programs that you've written in the past. What did you do before that would be helpful in solving this problem?

Try

Write one small piece at a time.

Test your program often!

Use comments to document what your code does.

Go back to the previous steps if you get stuck or aren't sure whether you've solved the problem.

Reflect

Compare your finished program to the defined problem to make sure you've solved all aspects of the problem.

Questions to Consider:

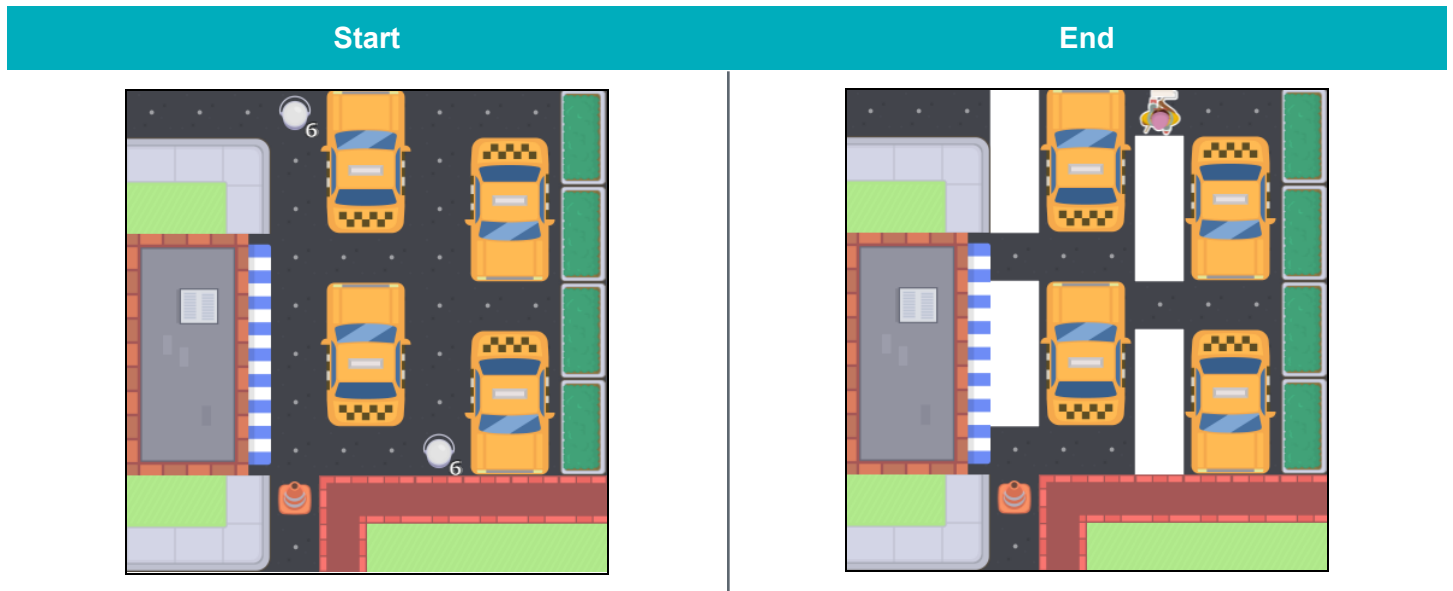
- Ask a classmate to try your program. Are there any parts of your code where they were confused?
- Ask a classmate to read your code to make sure that your documentation is clear and accurate. Do they understand what your code is supposed to do?
- What are some changes or improvements you could make to your program?

What have you accomplished?

What do you need to do next?

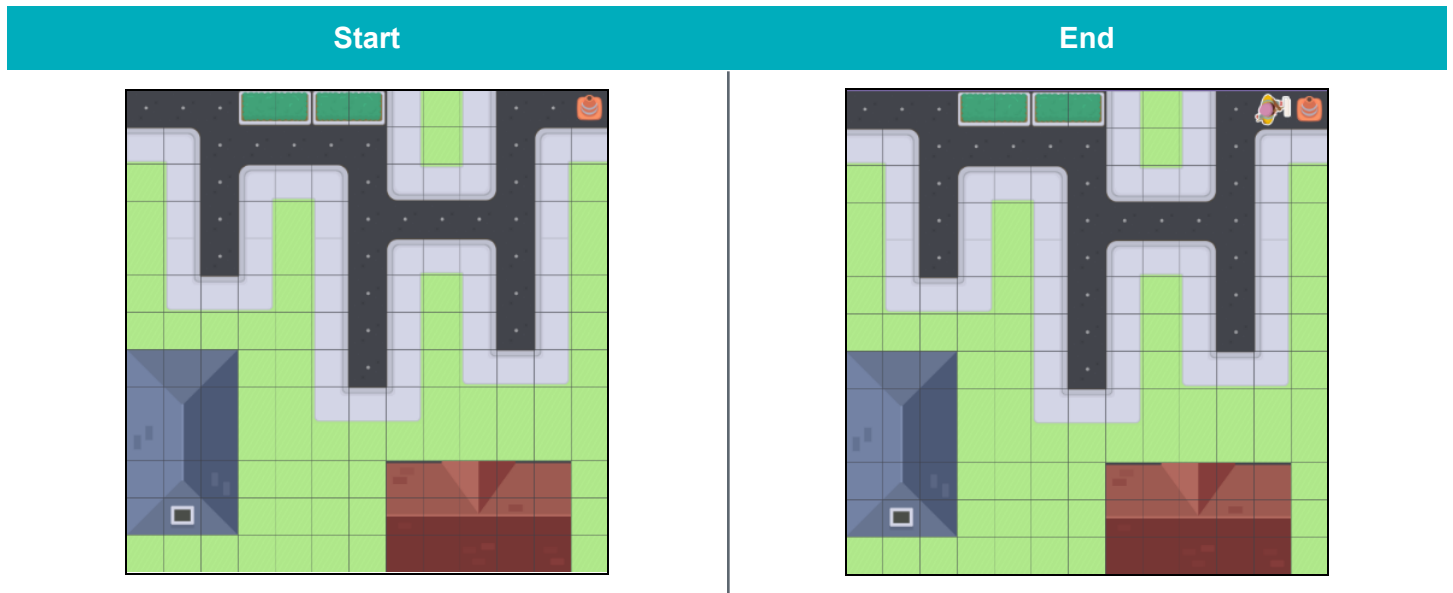
Choice A: TrafficPainter

The Painter has a new job, and they need help painting lines on each side of the taxis to help manage the traffic in The Neighborhood.



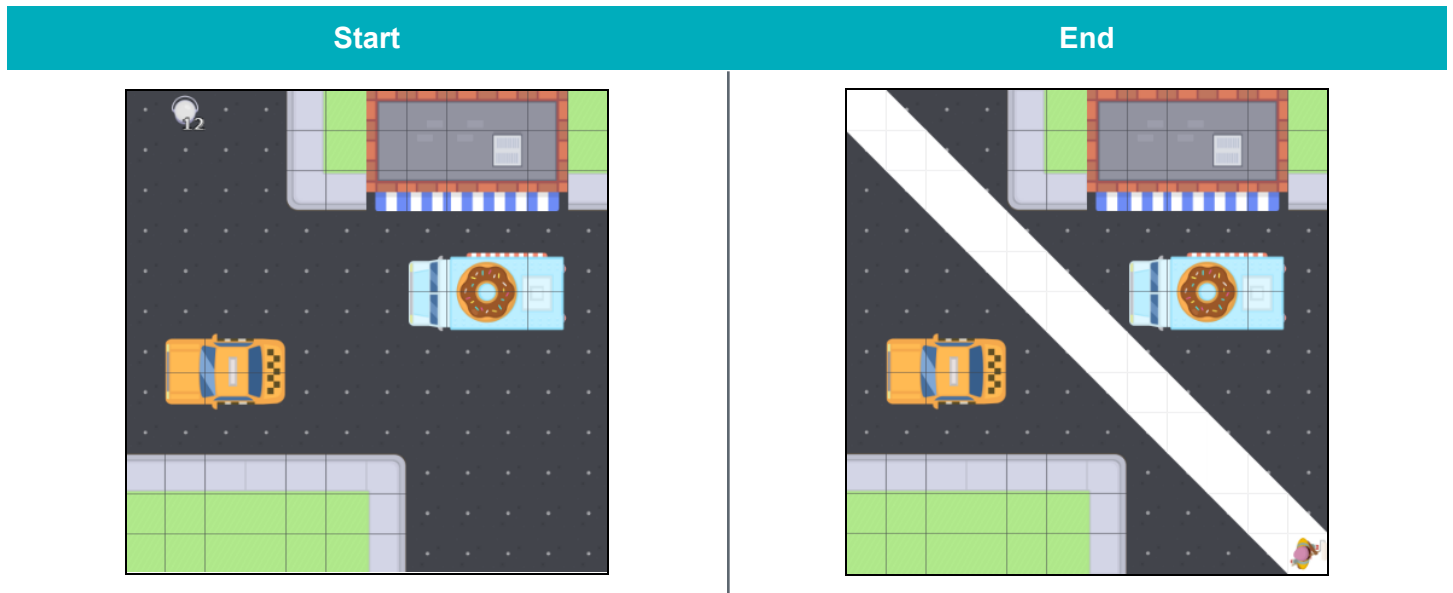
Choice B: MarathonPainter

The Painter is participating in a race, and they need help navigating around The Neighborhood to reach the finish line. **Note:** The Painter must go down each part of the street.



Choice C: MuralPainter

The Painter is planning a mural to paint in The Neighborhood, and they need help painting a diagonal line to get started.



Choice D: EventPainter

The Painter is volunteering at an event, and they need help decorating the areas in The Neighborhood.

