

Name(s) _____ Period _____ Date _____

Activity Guide - Writing Algorithms with 2D Arrays



Algorithms and the Problem-Solving Process

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

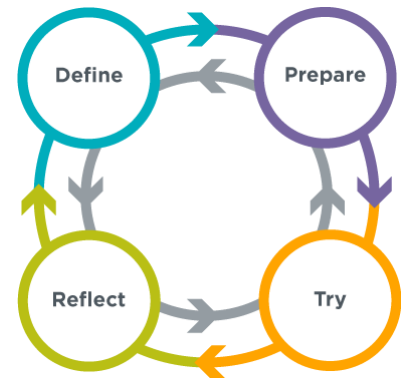
- Write out or draw the steps you need to take to solve the problem.
- List what you already know how to do and what you don't yet.
- Explain your algorithm to a classmate.
- Review similar programs that you've written in the past.

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 and identify the **precondition(s)** and **postcondition(s)** of the problem.

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: Movie Ratings

A movie streaming platform collects the number of user ratings received for different movies over a week. The number of ratings received are stored in a 2D array, where each row represents a movie, and each column represents a day of the week.

In the `Streaming` class, write the `calcTotalRatings()` method to calculate and return the sum of all movie ratings received during the month.

Precondition(s) the 2D array of movie ratings is not null and contains valid values

Postcondition(s) the 2D array of movie ratings is not modified

```
total = 0
loop from 0 to ratings.length - 1
  loop from 0 to ratings[0].length - 1
    total += rating
return total
```

Choice B: Sales Report

A store manager has the sales data for different departments stored in a 2D array, where each row represents a department and each column represents a day of the week.

In the `Store` class, write the `calcTotalSales()` method to calculate and return the total sales by all departments.

Precondition(s) the 2D array of sales data is not null and contains valid values

Postcondition(s) the 2D array of sales data is not modified

```
total = 0

loop from 0 to sales.length - 1
  loop from 0 to sales[0].length - 1
    total += value

return total
```

Choice C: Sports Scores

A sports analyst has collected data on the scores of various teams in a league over a month in a 2D array, where each row represents a team and each column represents a day of the week.

In the `League` class, write the `calcTotalScores()` method to calculate and return the total points scored by all teams.

Precondition(s) the 2D array of scores is not null and contains valid values

Postcondition(s) the 2D array of scores is not modified

```
total = 0
loop from 0 to scores.length - 1
  loop from 0 to scores[0].length - 1
    total += points
return total
```

Choice D: Traffic Analysis

A city planner has collected data on the number of vehicles passing through various intersections in a 2D array, where each row represents an intersection and each column represents a day of the week.

In the `Traffic` class, write the `calcTotalVehicles()` method to calculate and return the total number of vehicles that passed through all intersections.

Precondition(s) the 2D array of vehicles is not null and contains valid values

Postcondition(s) the 2D array of vehicles is not modified

```
total = 0
loop from 0 to vehicles.length - 1
  loop from 0 to vehicles[0].length - 1
    total += numVehicles
return total
```