

Overview:

Your task is to create an application that generates a watering schedule for the next 12 weeks for all our plants. We encourage you to be creative and present the schedule in a way that is easy for a plant caretaker to understand. The schedule should clearly show which plants need watering on which dates.

You are free to use any language and platform you're comfortable with. The solution can be a User Interface (UI), command-line tool, or even a schedule written to a file—whatever suits your style. Remember to include a README with instructions on how to run your code, as well as any known issues or complexities. Additionally, feel free to describe any extra features you would have liked to add.

Requirements:

- The schedule should cover the next 12 weeks, starting from **next Monday**.
- **Plants should not be watered on Saturdays or Sundays** (work-life balance!).
- Each plant has a watering frequency, and the application should generate a schedule based on that.
- Every plant should be watered on the first day of the schedule (next Monday), after which its specific schedule should be followed as closely as possible.
- You've been provided with a **JSON file** containing data about the plants, including their watering frequency (in days).

Flowchart Requirement:

Along with your code submission, please include a simple **flowchart** that demonstrates your approach to solving the watering schedule task. We expect you to use standard flowchart symbols to maintain consistency and clarity.

Please follow this basic flowchart notation:

- **Oval (Terminator):** Start and end points of your process.
- **Rectangle (Process):** Represents a process or action step (e.g., "Check if it's a weekend," "Assign watering date").
- **Diamond (Decision):** Represents a decision point (e.g., "Is it a weekend?").
- **Arrow (Connector):** Represents the flow of the process from one step to another.
- **Parallelogram (Input/Output):** Represents input or output (e.g., "Read plant data from JSON").

Reference for flowchart symbols:

<https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained>

Acceptance Criteria:

- The user can easily view which plants to water on which date(s).
- The schedule starts on **next Monday** and covers the following **12 weeks**.
- **No watering takes place on weekends (Saturdays and Sundays).**
- Each plant is watered based on its desired schedule, considering weekends.

Bonus Task: Watering Limit

As a **bonus challenge**, you can add the following rule to your watering schedule:

- The **plant caretaker can only water a maximum of three plants per day.**

This will require you to:

1. Prioritize which plants should be watered on a given day when more than three plants are due.
2. Distribute the remaining plants to the nearest available days while still adhering to the weekend rule (no watering on Saturdays or Sundays).

Feel free to describe your approach in the README or flowchart.

Assumptions:

- **No watering on weekends:** Plants can tolerate being watered a day before or after if their schedule falls on a weekend.
- **Work-life balance:** Assume we don't come to the office on weekends.
- **No time constraints:** You don't need to worry about how many plants can be watered in a day; watering an individual plant takes no time at all (except for the bonus challenge).
- **Start from next Monday:** All plants will be watered on the first day (next Monday), and then on their regular schedule.
- **Plant-specific schedules:** We assume that all the information you need (like the watering frequency) is already provided in the JSON file.

How to Submit:

- **Submission Format:** You can upload your solution to a public source control service like GitHub or Bitbucket, or share it via Google Drive, Dropbox, or another file-sharing service. Ensure that the folder permissions allow us to access it.

- **Packaging:** Please exclude any binaries or dependencies that can be resolved via a package manager (e.g., .gitignore node_modules or other similar folders).
- **Alternatively:** You can send a ZIP file of your code with a document or README explaining how to run your application.

Challenge Submission Review:

- We will review all submissions and schedule a follow-up video call with the strongest candidates. During the call, we'll discuss your choices, your thought process, and any challenges you encountered.
- Use this opportunity to showcase your skills and approach, even in areas where the requirements are not well-defined. Feel free to add creative features, as long as the core requirements are met.
- We encourage you to add this challenge to your portfolio and share it publicly. The README should include the challenge requirements and acceptance criteria from this document.

Sample Data

```
[
  {
    "plant_id": 1,
    "name": "Aloe Vera",
    "watering_frequency": 7
  },
  {
    "plant_id": 2,
    "name": "Peace Lily",
    "watering_frequency": 3
  },
  {
    "plant_id": 3,
    "name": "Spider Plant",
    "watering_frequency": 5
  },
  {
    "plant_id": 4,
    "name": "Snake Plant",
    "watering_frequency": 14
  },
  {
    "plant_id": 5,
    "name": "Fern",
    "watering_frequency": 2
  },
  {
    "plant_id": 6,
    "name": "Cactus",
    "watering_frequency": 10
  },
  {
```

```
    "plant_id": 7,  
    "name": "Orchid",  
    "watering_frequency": 7  
  },  
  {  
    "plant_id": 8,  
    "name": "Bamboo",  
    "watering_frequency": 4  
  },  
  {  
    "plant_id": 9,  
    "name": "Money Plant",  
    "watering_frequency": 6  
  },  
  {  
    "plant_id": 10,  
    "name": "Lavender",  
    "watering_frequency": 8  
  }  
]  
]
```