

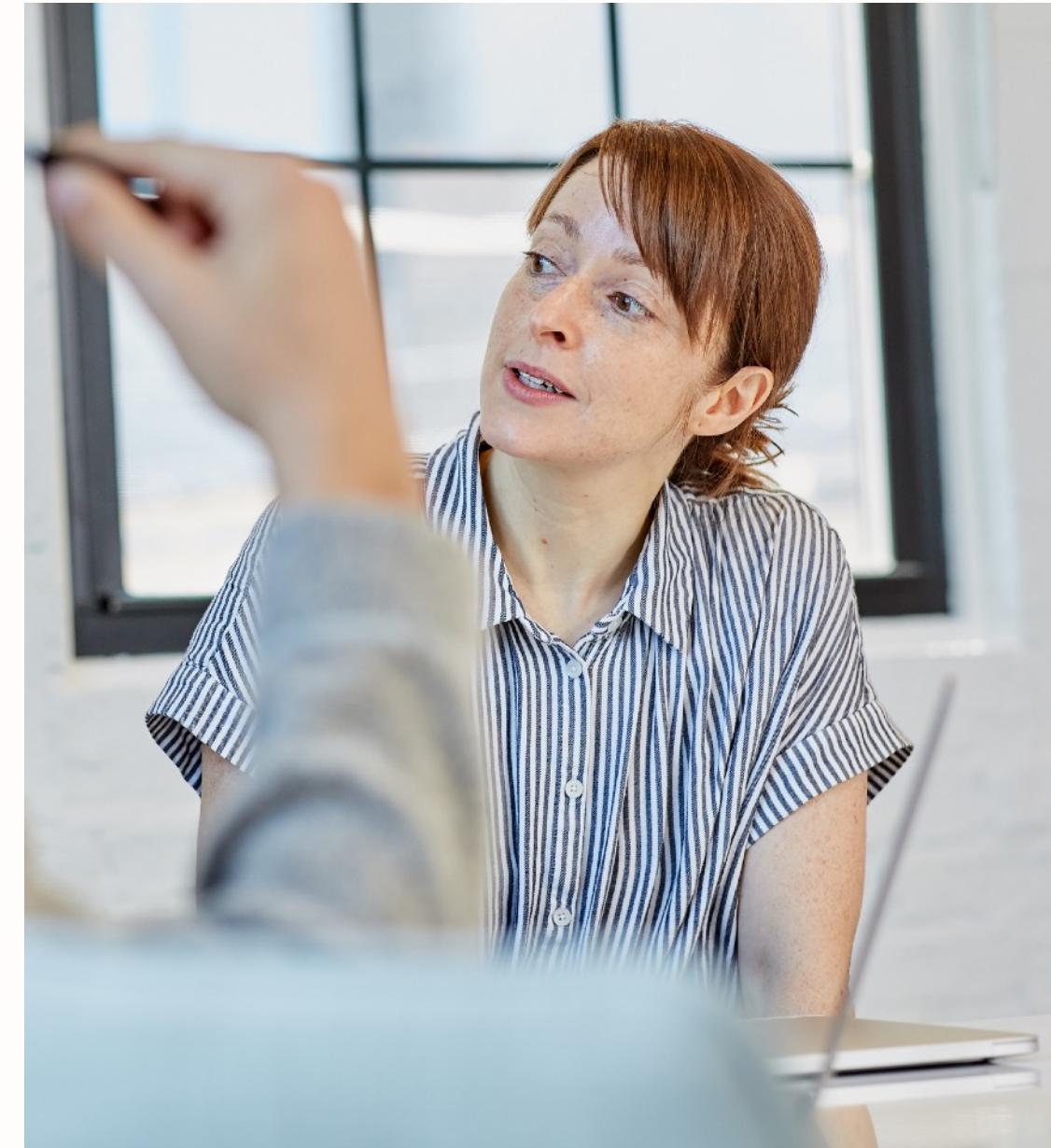
Tandem Code Challenge

Apprentice Software Engineer | December 2019

Welcome!

We're so excited you're considering joining the Tandem team!

In this document, you will find our 2019 Apprentice Software Engineer code challenge.





About our challenge

There is no strict objective measure for pass/fail. While there are defined acceptance criteria for the challenge, it is up to you to fill in the blanks for any requirement that is undefined. Requirements may seem unclear at times, but we are unable to provide additional information while your work is in progress. In the event something is ambiguous, make your own decision and provide the rationale.

Your submission should showcase your capabilities. We encourage you to be creative and play to your strengths. Use any language and platform you're comfortable with and expound on any area which interests you — so long as the core requirements are implemented.

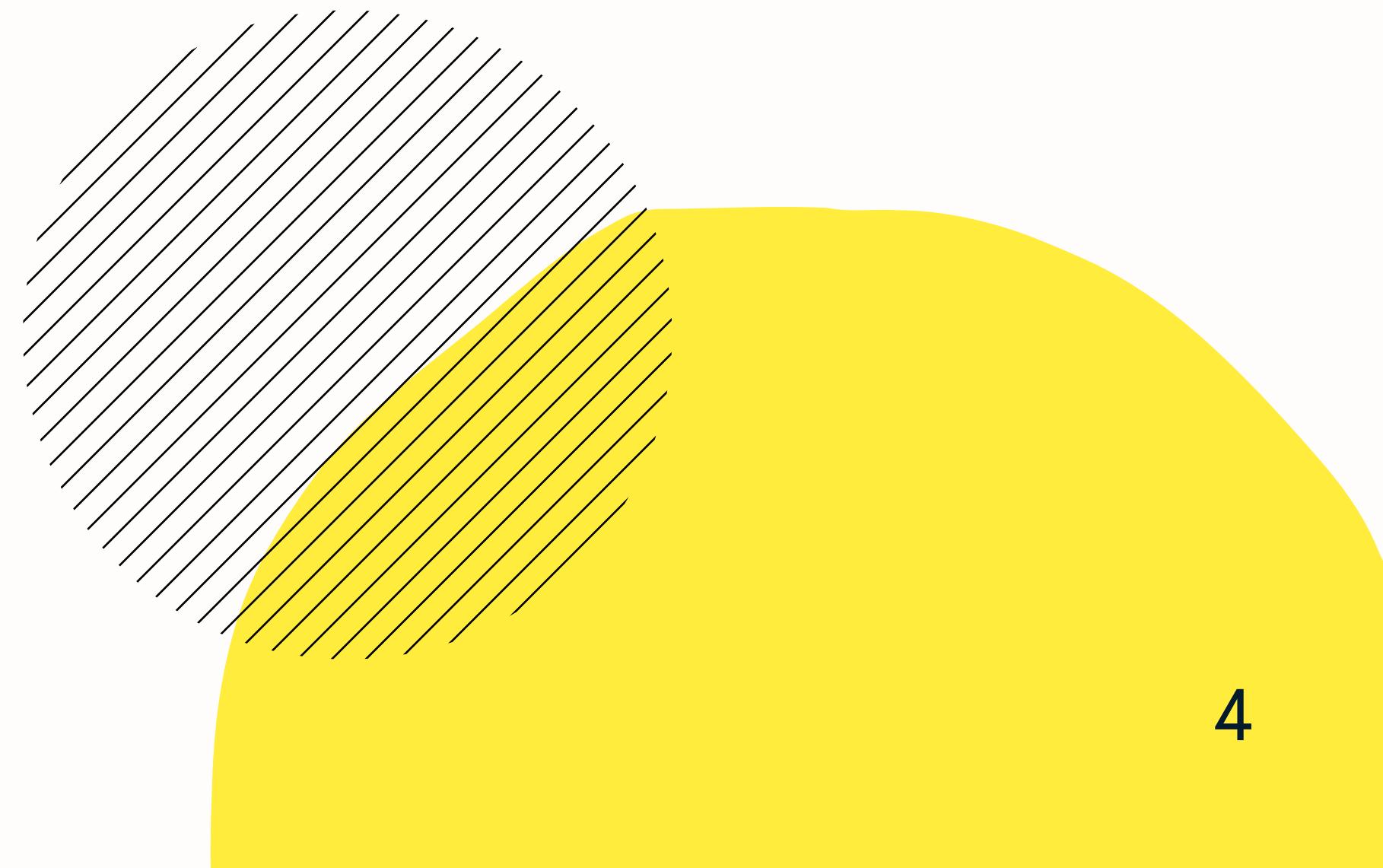
How to submit

The application window for our 2020 apprenticeship will close Monday, December 16, 2019. Please submit your challenge by 11:59PM CST on Monday, December 16, 2019. Unfortunately, submissions received after this time cannot be considered for our apprenticeship.

Please include a link to your submission in your application (we do have a question for this). You are welcome to put your code challenge submission in Github, Bitbucket, or a public source control service of your choosing. Or you can use Google Drive, Dropbox, or any other file sharing service you like. Just be sure the permissions to the file or folder allow us to access it.

Please exclude any binaries or dependencies that can be built or resolved via a package manager (remember to `.gitignore` those `node_modules!`). It's best to use as few external dependencies as possible to keep the setup simple.

Your submission must include instructions for how to run your code. You must also list any system dependencies (e.g. Ruby 2.3, Erlang runtime, JDK8, etc). Your README is a great place to put this information.



Challenge submission review

We will review all challenge submissions and schedule a follow-up video call with the strongest. In our call, we will discuss your choices and learn more about your thought process. Due to a high volume of submissions, we anticipate that we will be unable to meet personally with each candidate.

We encourage you to use this challenge in your online portfolio and include the requirements and acceptance criteria found in this document in your README. We can't hire everyone as much as we'd love to, but we want to leave you with assets to use as you continue your job search.

We also know that code reviews are wonderful learning tools. There are local communities that provide this support, if you're interested. One of our engineering managers runs [Dev Together](#), a community that hosts Code Review & Pairing events every other month. You share your code and are matched with a mentor who will complete a code review and pair program with you at the event.



Prompt: We grow in Tandem!

At Tandem, we are serious about everyone's growth, including that of our plants. Whether you are walking around upstairs or downstairs at the office, you will find our green friends happily growing like the rest of us. There are quite a few Tandelorians who will walk around the office and water the plants to ensure they are healthy.

However, we have quite a big plant family, and it is hard to manually keep track of when to water any particular plant. We don't want our plants to ever miss out on a watering!

Goal

Your goal is to create an application that generates a watering schedule for the next 12 weeks for all of our plants.

Use creative license in terms of how you want us to see this schedule. At minimum, the plant caretaker looking at the schedule should be able to easily identify which plants to water on a particular date. It could be a user interface (UI), command-line tool, written to file, etc.

We would also like to see a README which includes any information about how to run the code, any known issues or complexity we should look out for, and any additional features you would like to have added to make your scheduler even more awesome.

Before you begin, familiarity with the following concepts will be helpful:

- Arrays and loops
- Date manipulation
- Parsing JSON

Assumptions

- We do not water our plants on a weekend. Work-life balance is important, and we shouldn't be in the office on a weekend.
- Our plants are reasonably tolerant and will still be happy if they are watered a day before or after the day they should be watered.
- Watering an individual plant takes no time at all so you don't have to worry about how many plants can be watered in a particular day.
- The scheduling should start from next Monday and last for 12 weeks.
- All plants will be watered on the first day of the schedule (next Monday).
- We recognize that when to water a plant is heavily dependent on other factors such as soil, weather, humidity, etc. You can assume that we know exactly when to water these specific plants.
- You have been provided a JSON file which contains data for our plants.

Acceptance Criteria

- The user can view which plant(s) to water on which date(s).
- The schedule covers the next 12 weeks starting next Monday.
- No plants are watered on Saturdays or Sundays.
- Each plant is watered on its desired schedule or as close as possible, taking into account weekends.

Additional considerations

Tandem's first core value is "Ship Quality Work." We love unit tests and automated test coverage in our projects. Well-tested code provides documentation for other developers and prevents future regressions (bugs introduced after a change) among other things. If you would like a challenge, consider adding unit tests to display your commitment to quality code.



A group of approximately ten people are seated around a long, light-colored wooden table in a casual dining environment. They are all engaged in eating, with various plates of food, napkins, and utensils visible on the table. The individuals are dressed in casual attire, including shirts, hoodies, and a plaid shirt. The background features a plain white wall with a closed door and some green plants hanging from above. The overall atmosphere is relaxed and social.

tandem