Developer/SRE Coding Challenge

The purpose of this challenge is to see what agile engineering looks like to you. We want clean, maintainable, production-quality code. Because we are focused on multiple dimensions of your code, the functional dimension is narrow and focused.

We are not looking for fluency with any particular frameworks. You are neither limited in what frameworks or libraries you may use, nor are you penalized for using them. The language we want you to use depends on the role and the project. Likely options are ES6/TypeScript, Java, or Python. Your technical recruiter will let you know which we prefer in your case.

After we receive your code, we will schedule a half-hour discussion with you to talk about what you did, and why you did the things you did.

The Problem:

Our users are science teachers who are as comfortable using the command line as they are using a browser. In their “Heat and Temperature” science unit, they want to assign students temperature unit-conversion problems on paper worksheets. After students turn in their completed worksheet, the teachers want to be able to enter the questions and student responses into a computer to be graded.

Their students will do problems to convert between: **Kelvin, Celsius, Fahrenheit,** and **Rankine**.

Requirements:

1. The teacher must be able to provide an input temperature.
2. The teacher must be able to provide a target unit of measure.
3. The teacher must be able to provide a student’s numeric response.
4. The student’s response must match an authoritative answer after *both* the student’s response and authoritative answer are rounded to the ones place. The system indicates that the response is **`correct`**, **`incorrect`**, or **`invalid`**.
5. Implement a basic continuous integration/continuous deploy pipeline for your code using your solution of choice (cloud solutions are acceptable), or describe how you would implement a CI/CD pipeline. What you provide should support a peer code review process and seamless app deployment when a commit is merged to trunk. The process should provide the organization with confidence that the changes deployed will not break existing app functionality.

Example scenarios (not exhaustive):

Input Temperature	**Target Units**	**Student Response**	**Output**
84.2 Fahrenheit	Rankine	543.5	correct
-45.14 Celsius	Kelvin	227.51	correct
317.33 Kelvin	Fahrenheit	110.5	incorrect

444.5 Rankine	Celsius	-30.9	incorrect
6.5 Fahrenheit	Rankine	dog	incorrect
dog	Celsius	45.32	invalid

Submitting your response:

1. Create a public GitHub repo that will contain your code. (If you prefer to make it private, that’s okay, just give the GitHub user “`FlexionCodeReview`” permission to read your repo.)
2. Include a `README.md` that explains how to install (or deploy) and run (or access) your program.
3. Notify our technical recruiter that you are done, and provide your repo URL.
4. No more changes can be committed after the deadline given to you by the technical recruiter.