Let's Go Bowling

Overview

Create a 10-pin bowling scoring application that uses traditional scoring rules to score each frame.

Put your project in Github and send us the repo when you are done.

Please note, that while the overall assessment is time boxed to approximately 48 hours, the work required should only take a couple of hours. If you find that you are bumping up against that time, you should look to try to simplify your approach.

Scoring rules

A good overview of how to score the bowling game can be found here: https://www.thoughtco.com/bowling-scoring-420895

Example scenarios:

Scenario 1:

Given a score of a strike or spare, and it is the first frame,

When I provide "X" or "7,/", respectively, via the command line,

Then the recently completed frame number is printed to the command line along with the current score, which is empty.

Scenario 2:

Given an open frame score, with the first attempting being 2 and the second being 3,

When I provide "2,3", via the command line,

Then the recently completed frame number is printed to the command line along with the current score that is printed out which is the accumulative score of the previous frames and the newly entered score.

Scenario 3:

Given the current frame is the 10th frame, and 3 attempts are awarded due to at least 1 strike or spare in the frame,

When I provide "3,/,X", via the command line,

Then the recently completed frame number is printed to the command line along with the final score after which the application exits.

Requirements

- Scores should be provided to the application through STDIN and the current recorded score should be printed to STDOUT after accepting the new score.
- The recently completed frame number and current score must be printed to STDOUT after each score submission.
- The application should exit once the score for the 10th frame is submitted and the final score is printed to STDOUT.
- The application must support at least 1 user. If you support more than 1 user, you must include user identifying information in the scoring output summary.
- You must include tests with your submission.
- Provide instructions for how to run your application. Preferably in a README.md at the root of your project.

Additional Notes

- You can use whatever language you prefer.

BONUS

Performing the following optional tasks will result in extra points being awarded:

- Using Clojure
- Dockerizing your application

Be prepared to discuss your code and add to it later in the interview process.