Please read these instructions through to the end carefully before you begin coding. Two exercises are provided, and it’s important to remember that we don’t expect you to work out an exhaustive set of test cases for either. Rather, the point is to have a common set of experiences that we can all use to frame our follow-up conversation. The point is also to have fun, so both exercises involve automating testing for games.

The first exercise calls for automating a set of UI steps, and the second for automating a set of API steps. Start with the one you’re most comfortable with and, if time allows, move to the other. Use whatever tool you’d like for each. We’re much less interested in the tools you use and how far you get than we are in your approach to automation tasks and how you respond to the inevitable issues that come up along the way. If you choose a tool that turns out not to be up to this task at all, and you’re able to explain its shortcomings in this context, we’ll be impressed. If the directions aren’t clear and unambiguous, and you point out where they can be made better, we’ll be impressed. If the app’s architecture isn’t automation friendly, and you have suggestions for improving it, we’ll be impressed. *We want to be impressed, and we don’t want to hamper your ability to play to your strengths*: come to our follow-up prepared with your impressions and opinions for a colleague-to-colleague exchange of ideas.

We urge you to timebox these exercises. Again, the point is to have a common set of experiences we can refer to, and these exercises allow us this, even if you and we have never worked on automating the same commercial app as part of our day jobs. Your spending no more than two hours, total, would provide us with enough to talk about during our follow-up.

**I – The Checkers Game**

For this exercise, implement the following steps:

1. Navigate to https://www.gamesforthebrain.com/game/checkers/
2. Confirm that the site is up
3. Make five legal moves as orange:
   1. Include taking a blue piece
   2. Use “Make a move” as confirmation that you can take the next step
   3. Restart the game after five moves
   4. Confirm that the restarting had been successful

You’ll find that the game engine is predictable enough to allow for you to win a game reliably. So, you can try for this if you’re inclined and have a little time left over.

**II – The Card Game**

For this exercise:

1. Navigate to https://deckofcardsapi.com/
2. Confirm the site is up
3. Get a new deck
4. Shuffle it
5. Deal three cards to each of two players
6. Check whether either has blackjack
7. If either has, write out which one does

Keep in mind that, for these exercises, quality wins over quantity. Showcase your coding craftsmanship and standards.

If, for some reason, implementing the steps in code becomes onerous or you don’t have time to complete all seven steps for each exercise, write a quick outline of your strategy and the points where you’d need to think about implementation details. If you’d need help to complete this out during a sprint, let us know what questions you’d ask of whom.

If you have any questions about how to proceed or if any part of these instructions isn’t clear, don’t hesitate to reach out to the recruiter who contacted you. That person will play the role of scrum master in supplying any needed information and removing any obstacle you might encounter.

Once you’ve gotten as far as you can, please zip the files containing your work into two packages: one for the checkers game and one for the card game. Send them as attachments to an email addressed to the recruiter who contacted you. Feel free to note anything you’d like us to know as we look at your work. Remember that a clear summary of what happened when is a valuable asset throughout the testing process.