

Open Book Coding Exercise

This exercise is intended to help you understand your ability to undertake research whilst delivering a solution under time pressure. This is an open-book test; you will not be penalised for using appropriate references. Use of eval in this exercise is **not allowed**.

YOU WILL BE ASSESSED ON TWO GROUNDS; **fulfilling the task at hand** and **elegance of the coded solution**. Good luck with the exercise, and we look forward to seeing your code!

Submission Guidelines

We are firm believers in version control system. We would also like to see how your solution evolves. As such, we would like you to use Git to manage your version history. Please ensure that you make regular commits of your code with a descriptive message to ensure that we can understand how your solution evolved as you tackled the requirements.¹

The top level of your code-base should contain a file `README.md`, which should explain how to run your code, and a file `ASSUMPTIONS.md`, which details any assumptions you made during implementation, the reasons you made these assumptions and why they were justified.

Please upload your code to Github or Bitbucket and send the project url directly to me.

Task A - Valid Expression

Write program that reads a mathematical expression and determines whether such an expression is valid. For this exercise consider **only** the following operators: `+`, `-`, `*`, `/`, `sin`, `cos`, `tan`. **Do not use any external libraries for this part.**²

Task B - Evaluate Expression

Given a valid expression evaluate. For example, $\sin(30) + \cos(20) = -0.5799495623$. **Do not use any external libraries for this part.**

Task C - Calculator

Write a **Single Page Application** that allows the user to input an expression, validates the expression (on every keypress) and evaluates the result. If the expression inputted by the user is invalid an error message should be displayed. **Do not use any external libraries for this part.**³

¹ You may use any language you consider reasonable to tackle the problem at hand in the given time and the given evaluation criteria described above.

At a technical level, you are not being marked on a particular language, but rather on your ability to harness some language to solve the problem at hand.

² Use these test cases to validate your implementation:

Expression	Valid
$3 + 2 + 4$	T
$+2$	T
-2	T
$\sin(\sin(30) + \cos(20))$	T
$\sin(30) + \cos(20)$	T
$\sin(30)$	F
$3 + +$	F
$3 +$	F

Note that expressions are recursive.

³ Browser compatibility is not an issue here, consider latest Chrome as your test browser.

Add an area within the UI that will contain the last 5 expressions and their computed results.

Task D - Supr-Calculator

- Add a RAND operand that will generate a random number. The RAND should get its input from endpoint <https://www.random.org/integers/?num=1&min=1&max=100&col=1&base=10&format=plain&rnd=new>
- Display a calculator UI with buttons for every digit and operator allowing the user to input an expression without using the native mobile keyboard. The layout should respond to device orientation.