# Programming Test

You have two days to complete the programming test.

Your answer should:

- **Contain solutions** to as many problems (and levels) as you have time to complete given the time.
- **Be possible to build** (compile/transpile/package) from source code on the command line by us - both applications and tests.
- **Be possible to execute/launch** from the command line by us - both applications and tests.
- **Include instructions** (README.txt) on how we do the two above. Use a build system (like Gradle, Maven, npm, Grunt, dotnet, or similar) or supply a launch script (like a BAT-file or shell-script).
- **Be delivered as a ZIP file** containing the complete source code and any documentation you wish to provide.
    - Put the resulting ZIP file on a service like Google Drive, Dropbox, OneDrive, or similar.
    - Share a link (URL) to the ZIP file in an email with your interviewer or contact person.
    - NOTE: Trying to mail a ZIP file directly as an attachment seems to fail a lot these days due to security constraints in popular mail services, so avoid doing that.

You can use a programming language of your choice, but pick one you are good at and think is relevant for the job you are applying for. Java, C#, JavaScript, TypeScript, Python, and Kotlin will work for Problem 2, so the recommendation is to choose one of those languages.

You should **not** use any external libraries other than:

- The libraries that are a standard part of your programming language's platform (like the JDK for Java)
- A unit test framework of your choice for creating unit tests (but not solutions)
- Selenium WebDriver for Problem 2

NOTE:

- You must own the copyright to all code that you present as your own.
- You must adhere to license agreements for libraries you use.
- You must be able to explain all algorithms you use in your solution.

We will review your solution and call for an additional interview to go through everything together with you.

Good luck!

## Problem 1: Detect Anagram

Write a **function** to check if two provided strings are anagrams of each other.

Your function should return `true` if two strings are anagram, `false` otherwise.

Two strings are anagrams if they are the same length and contain the same characters but in a different order. For example, "army" and "Mary" are anagrams.

You should ignore cases for this problem.

Show that the function works as intended with **unit tests**.

# Problem 2: Find all links on a web page

Write an **application** in your language that scans a web page (like http://google.com) for all existing web-links (<a> HTML tags) and prints them to the console or presents them on a web page.

Use **Selenium WebDriver** for navigating to and reading the DOM of the web page.

# Problem 3: Calculator

Create a **class** that works as a calculator and can handle addition, subtraction, multiplication, and division.

The calculator accepts a string with a mathematical expression to evaluate and returns the answer as a number.

Create a **console application** *or* **a web page** that accepts string input from a user and presents the result. No fancy user interface is required.

Do as many levels of this task as you can, given the time available. Present a different solution for each level.

## Level 1

The calculator handles exactly *one* operator and two numerical parameters.

```
calculator.evaluate("2+30"); // returns 32
calculator.evaluate("2 * 3"); // returns 6
calculator.evaluate("2 -3"); // returns -1
calculator.evaluate("2 / 3"); // returns 0.6666666...
```

## Level 2

The calculator handles multiple operators of the same precedence.

```
calculator.evaluate("2+30+4"); // returns 36
calculator.evaluate("2 - 3 + 4 + 15"); // returns 18
calculator.evaluate("2 * 3 * 4"); // returns 24
calculator.evaluate("2 * 3 / 4 * 20”); // returns 30
```

## Level 3

The calculator handles multiple operators of different precedence.

```
calculator.evaluate("2+3*40"); // returns 122
calculator.evaluate("2 * 3 + 4"); // returns 10
calculator.evaluate("2 / 3 + 4 - 1"); // returns 3.66666666....
calculator.evaluate("2 - 3 * 4"); // returns -10
```