**Assessment 1:**

The following file lists the things that you need to test: lodash-tests.test.js  
Think of appropriate test cases by yourself.

**NPM Package(s) that you will need:**

<https://www.npmjs.com/package/lodash-es>   
(Lodash is a popular library which contains many utility functions. lodash-es is lodash exported in the ES Module format, so that you can use import/export syntax. [The regular lodash library](https://www.npmjs.com/package/lodash) requires you to use require(‘lodash’))

**References:**

Lodash Reference: <https://lodash.com/docs/4.17.15>

**Assessment 2:**

The following file lists the test cases that you need to test ./pokemon/index.test.mjs

You might need to fix some bugs on ./pokemon/index.mjs

**Making Tests More Readable and Easy To Write:**

./test-utils.mjs file contains utility functions that you can use to write tests more easily. You might need to use these to do certain assertions. For a sample, see ./example-lodash-test.mjs. Also, run this file using node to get a feel for the tests.

**Recommended Order:**

- Part 1 of Assessment 2

- Assessment 1

- Part 2 of Assessment 2

**Hints (Disregard until you run into trouble if you don’t want to go into the assessment confused):**

- Remember there are different types of JavaScript modules (require, ESM) and there might be some requirements you need to fulfil in order to use them.

- You can pass a function into a function

- Know your tokens. Calling a function is not the same as referencing it:

myFunction // This is referencing a function

myFunction() // This is calling a function

- The this keyword works in mysterious ways

- JavaScript is weird! Once you create an object, you can easily overwrite its properties and methods. For example:  
const me = new Person(‘slim shady’);

me.sayName = () => { return ‘im not slim shady’ };