Exercises

- **4.1 File concatenation**: Write the implementation of `concatFiles()`, a callback-style function that takes two or more paths to text files in the filesystem and a destination file:

```
function concatFiles (srcFile1, srcFile2, srcFile3, ... ,
                      dest, cb) {
  // ...
}
```

  This function must copy the contents of every source file into the destination file, respecting the order of the files, as provided by the arguments list. For instance, given two files, if the first file contains *foo* and the second file contains *bar*, the function should write *foobar* (and not *barfoo*) in the destination file. Note that the preceding example signature is not valid JavaScript syntax: you need to find a different way to handle an arbitrary number of arguments. For instance, you could use the **rest parameters** syntax (`nodejsdp.link/rest-parameters`).

- **4.2 List files recursively**: Write `listNestedFiles()`, a callback-style function that takes, as the input, the path to a directory in the local filesystem and that asynchronously iterates over all the subdirectories to eventually return a list of all the files discovered. Here is what the signature of the function should look like:

```
function listNestedFiles (dir, cb) { /* ... */ }
```

  Bonus points if you manage to avoid callback hell. Feel free to create additional helper functions if needed.

- **4.3 Recursive find**: Write `recursiveFind()`, a callback-style function that takes a path to a directory in the local filesystem and a keyword, as per the following signature:

```
function recursiveFind(dir, keyword, cb) { /* ... */ }
```

The function must find all the text files within the given directory that contain the given keyword in the file contents. The list of matching files should be returned using the callback when the search is completed. If no matching file is found, the callback must be invoked with an empty array. As an example test case, if you have the files `foo.txt`, `bar.txt`, and `baz.txt` in `myDir` and the keyword `'batman'` is contained in the files `foo.txt` and `baz.txt`, you should be able to run the following code:

```
recursiveFind('myDir', 'batman', console.log)
// should print ['foo.txt', 'baz.txt']
```

Bonus points if you make the search recursive (it looks for text files in any subdirectory as well). Extra bonus points if you manage to perform the search within different files and subdirectories in parallel, but be careful to keep the number of parallel tasks under control!