Homework 5 Code

Due: Monday, November 4th, 11pm

If you have not completed HW5 written. Close this file and finish the written component before starting the coding parts in this worksheet.

To get started, check out the starter code for this assignment:

```
git clone https://gitlab.cs.washington.edu/cse331-24au-materials/hw5-zip.git
```

Navigate to the `hw5-zip` directory and run `npm install --no-audit`. Run tests with the command `npm run test` and run the linter with the command `npm run lint`.

Submission

After completing all tasks to follow, submit your solutions on Gradescope. The following completed files should be submitted to **"HW5 Code"**:

`cipher.ts` `cipher_test.ts` `zip_sum.ts` `zip_sum_test.ts`

Wait after submitting to make sure the autograder passes, and leave yourself time to resubmit if there's an issue. The autograder will run your tests, additional staff tests, and the linter.

Task 6 – Spice Apple Cipher [12 pts]

In this problem, we will translate mathematical definitions for functions into TypeScript code.

We will treat the math definitions as the imperative specifications for the TypeScript functions, so the translations should be "straight from the spec" –a direct translation.

Unlike last week, we have not provided any tests for these functions. (You will write those in the next part!) However, you proved some claims related to these definitions in your written assignment, so if you translate those to code directly, you should already have some confidence that your code is correct.

(a) Translate your mathematical definitions for ns and cipher, into TypeScript code in `cipher.ts`.

Complete the TODO in the `@returns` statement in the function specification above each of these functions by filling in the mathematical definition that you translated into TypeScript in the function body.

As a reminder, we originally described these functions in English and had you formalize them in HW4 Task 3. If you have since realized that you made a mistake in your original mathematical definitions, feel free to fix those when you type them up here. No additional explanation is needed, as we will refer to your comments to validate your translation.

(b) Translate your mathematical definition for f from HW5 Written Task 5 into TypeScript code in `zip_sum.ts`.

Complete the TODO in the `@returns` statement in the function specification above each of these functions by filling in the mathematical definition that you translated into TypeScript in the function body.

Additionally, if you find a mistake in your mathematical definition that you wrote in Task 5 while translating/testing your code, feel free to fix that mistake in your `@returns` statement, and write a brief explanation of what you fixed for a chance to get some points back for that mistake in your written work.

We expect students may have questions about how to represent special characters when typing up your math definitions in a plain text comment, so below is an example for a TypeScript implementation of "sum" which will hopefully clarfiy these:

```
/**
 * Finds the sum of all elements in the list
 * @param L list of ints to sum elements of
 * @returns sum(L) where sum is defined as follows:
 *    sum: List<Z> -> Z
 *    sum(nil)  := nil
 *    sum(a::L) := a + sum(L)
 */
```

Task 7 – A Mult of Personality [8 pts]

Now, for our final step in gaining confidence our code is correct, it's time to test the functions we translated in the last part!

Your tests should follow the testing requirements for this course (see the notes on testing for a reminder). Additionally, write short labels describing which coverage requirement is met by each test. See the example from HW4 for reference.

(a) Write tests for ns and cipher in `cipher_test.ts`.

(b) Write tests for f in `zip_sum_test.ts`.