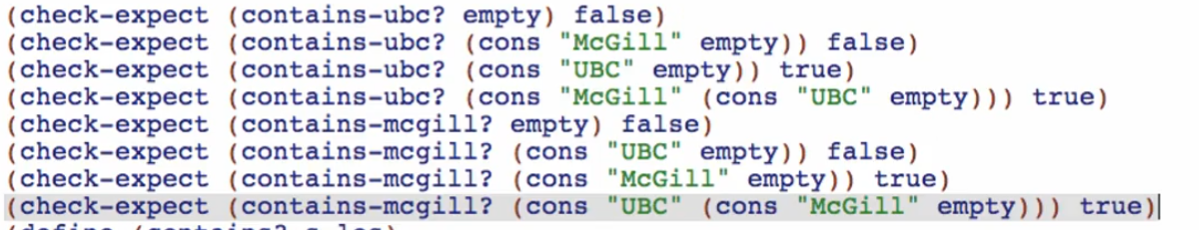
After the working function definition, we produce the check-expects and the purpose.

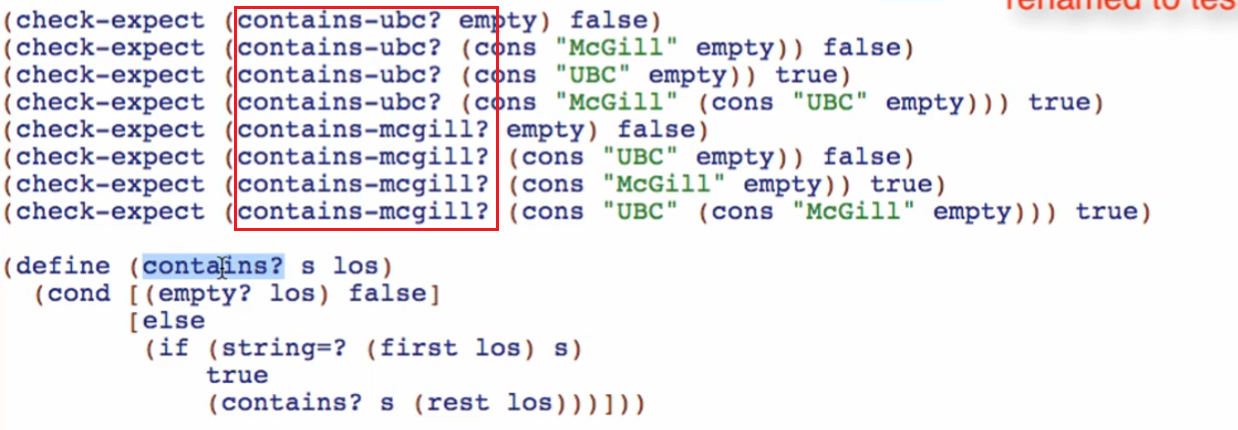
*Working through the function definition of contains? (backwards)*

**Examples**

1. Copy first the ALL the original check-expects (copy from all the functions)



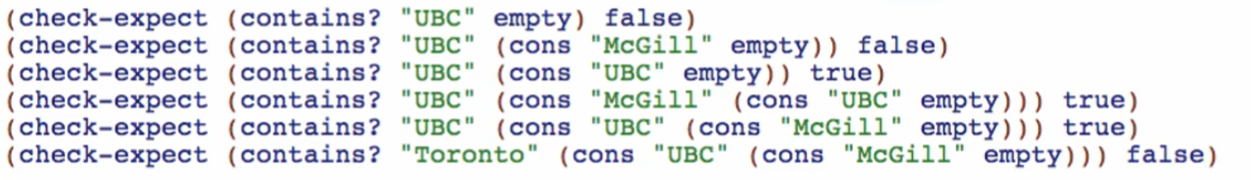
1. Rename the function calls to our more general function name



1. Add the additional argument (your added parameter for the point of variance) and delete duplicate tests and also not necessary tests

Necessary tests:

* One base case only
* At least One 1 long (success and fail)
* At least One 2 long (success and fail)



Check if running and tests are passing

**Purpose**

1. Copy the less abstract purposes/the original purposes



* Sometimes purpose for abstract function can be abstracted from the original purposes, but NOT ALWAYS



* It’s backwards

1. 

* We abstracted this purpose from the original purposes, but take note that this will NOT BE ALWAYS the case

*Working through the function definition of map2 (backwards)*

**Examples**

1. Copy first the ALL the original check-expects (copy from all the functions)

OR you can use the original principles that we are using if we are defining functions in normal manner (from scratch)

base case: 

2 long for list parameter:



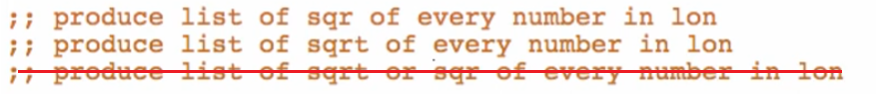
For the function parameter:



Check if running and tests are passing

**Purpose**

1. Copy the less abstract purposes/the original purposes



* Sometimes purpose for abstract function can be abstracted from the original purposes, but NOT ALWAYS, and THIS IS ONE OF THAT CASES
* WE ARE NOT LIMITING THE FUNCTION TO sqr AND sqrt FUNCTIONS ONLY

Eg:



1. So try and try to produce the best purpose for this function

* It is natural or perfectly normal to NOT ALWAYS get the abstract purpose on the first try!

FINAL:



BASIS OF THE PURPOSE:

