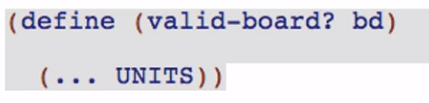
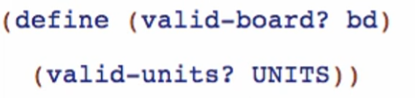
Working through valid-board? Function



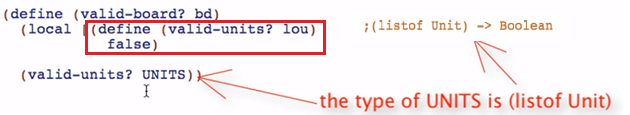
We are going to want to operate on UNITS (the list where ROWS, COLUMN, and BOXES are in)



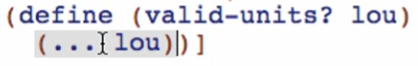
Then wish again for a function to check if all units are valid

Let’s define *valid-units*? LOCALLY

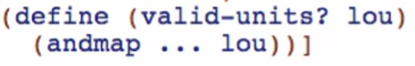
**Stub**



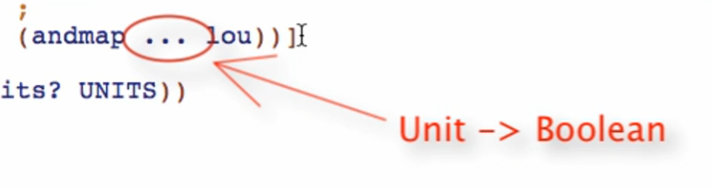
**Template**

****

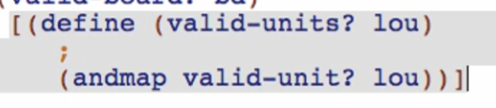
Take note that this needs to make sure that EVERY unit of this list should be valid. Thus we can use ANDMAP:



**Code body**

****

This needs to have a function to check every UNIT if valid, but we don’t have one so it will also be a wishlist entry

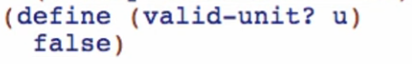


*valid-unit Function*

**Signature**

****

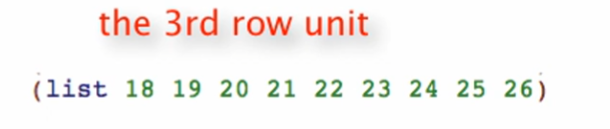
**Stub**

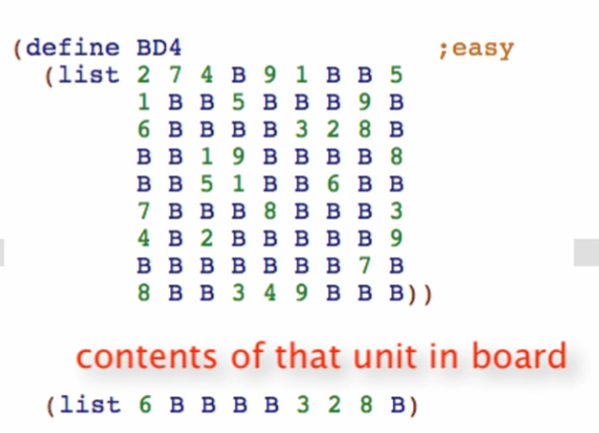
****

**Template**

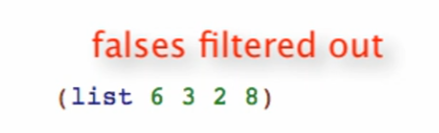
Main logic of identify if a unit is valid

Read the unit:





Filter out the falses:

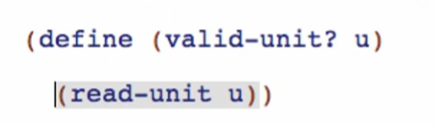


Then check if there are no duplicates:

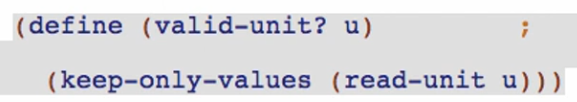


Let’s try function composition for the template based on our logic

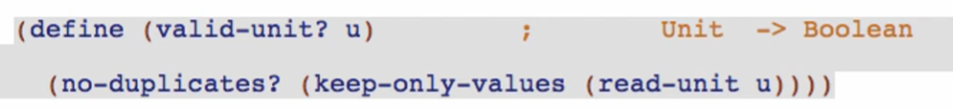
Reading the unit:



Filter out the falses/keep only the values



Make sure there are no dupes



Now we have 3 more wish list entries, read-unit, keep-only-values, and no-duplicates

*read-unit Function*

**signature & stub**

****

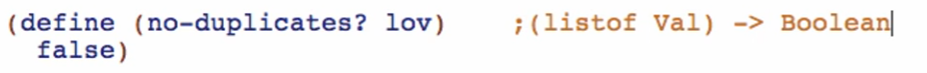
*keep-only-values Function*

**signature & stub**

****

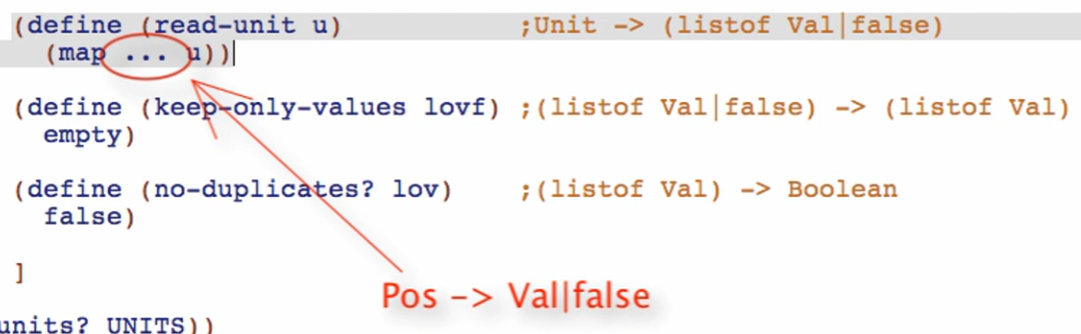
*no-duplicates* *Function*

**signature & stub**

****

Working through the *read-unit Function*

**Template**

****

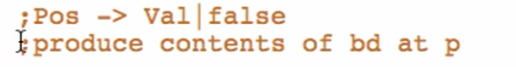
**Code body**

****

Wish for *read-pos* since we still don’t have a function like that

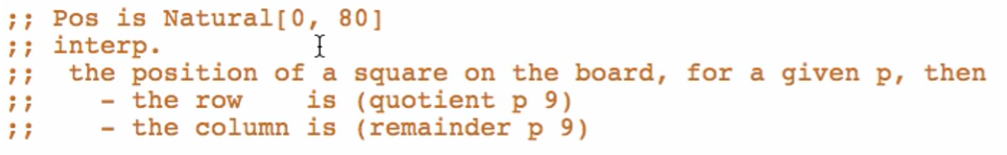
read-pos Function

**signature, purpose & stub**

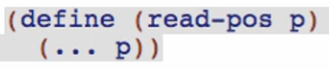
****

****

**Template**

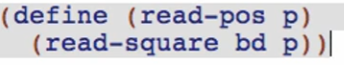
****

atomic non-distinct:

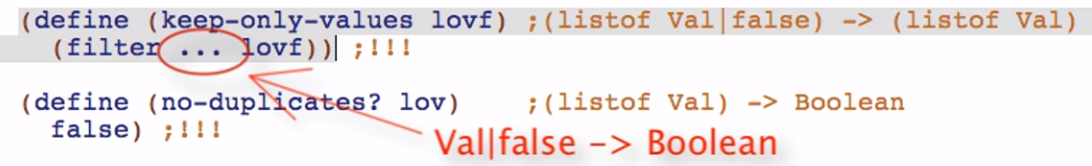


**Code body**

based on our signature, we can use *read-square* Function that we previously defined

****

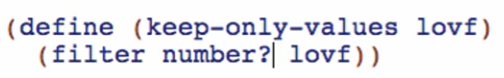
Working through *keep-only-values* Function

**Template** ****

**Code body**

****

number? To filter out “false” since it is a boolean

****

Working through *no-duplicates* Function

**Purpose**

****

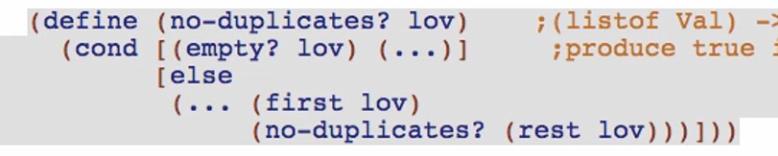
**Template**

****

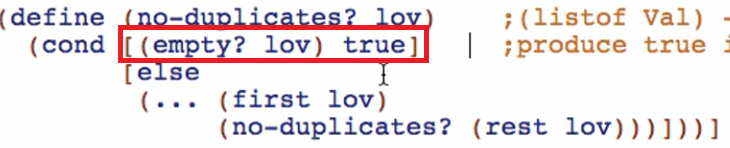
****

****

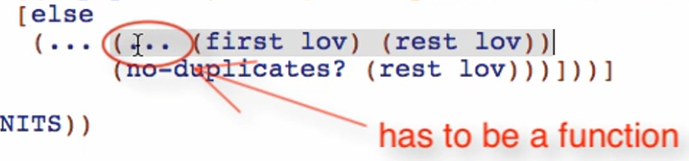
We need to loop through all the elements!



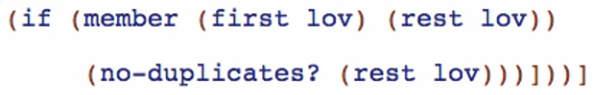
**Code body**



Getting at the end of the list without any duplicates will produce true 😊



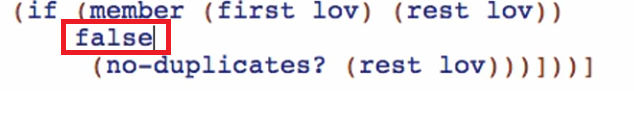
We need to ask if the first element appears within the rest of the elements



member is a built-in function that tells us whether the element is in the list

(member <element to be searched> <list to be searched on>) -> Boolean

This is like contains in java



If an element is found within the rest of the list, produce false

Else proceed to recursion to check for the next element

Run and debug! Don’t forget to comment out all the tests you’ve previously commented!