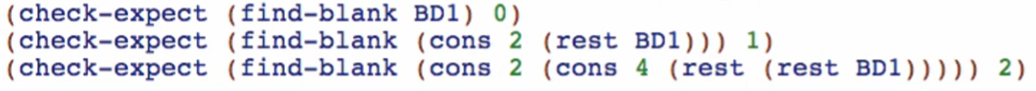
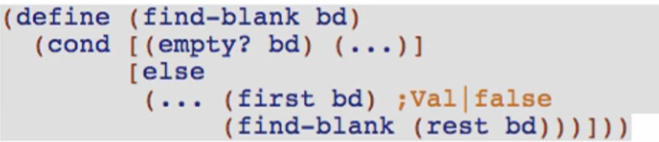
*find-blank Function*

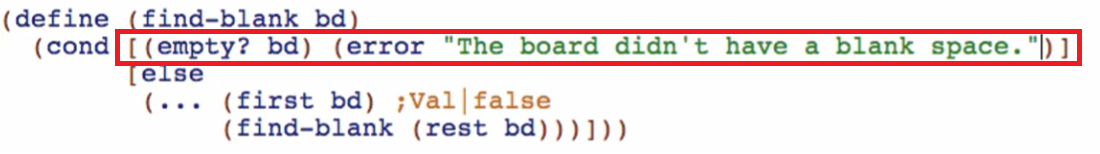
**Examples**



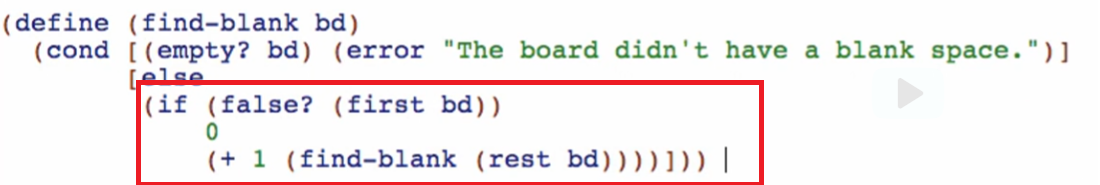
**Template**



**Code body**

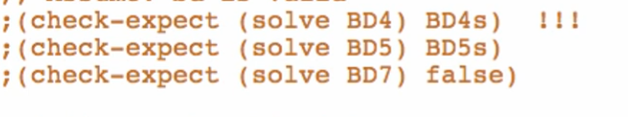


Because we assume (surely) that the board will always have at least one blank space (based on our generative recursion), this empty? bd isn’t supposed to happen thus we throw an error.



We add 1 for the position every time we traverse through the board for the position of our blank space within the rest of the board

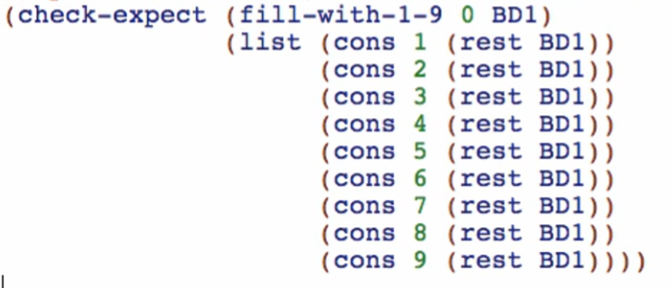
Run and debug!



Comment out solve test cases since it will always fail. This is to simplify our debugging process of the other helpers

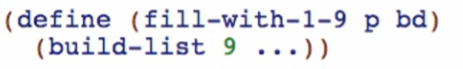
*fill-with-1-9 Function*

**Examples**



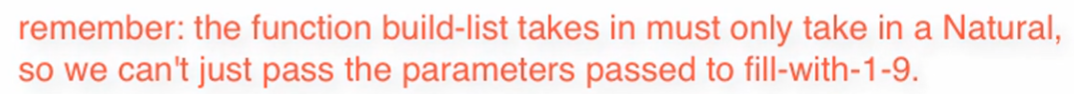
Check if well-formed

**Template**

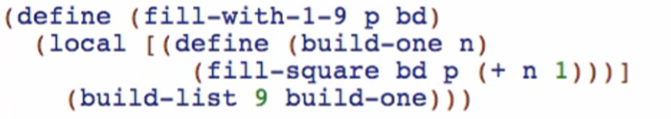


(build-list <number of elements> …)

**Code body**



Thus we define this local:



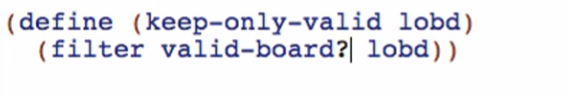
Remember that we have previously defined fill-square and the argument for the value should be n+1 since we used 0-based indexing for our board while in real-life sudoku, 1-9 are the possible values

*keep-only-valid Function*

**Examples**



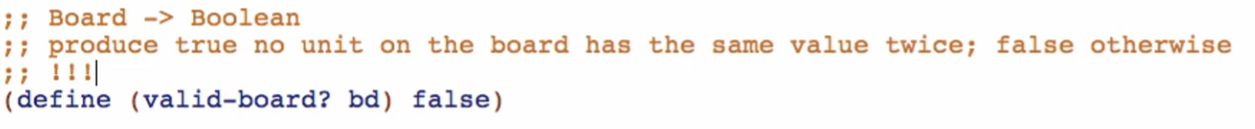
**Template**

****

We need to wish valid-board in top-level. It doesn’t have to be a closure because it doesn’t have to see lobd argument

Wish for *valid-board? Function*

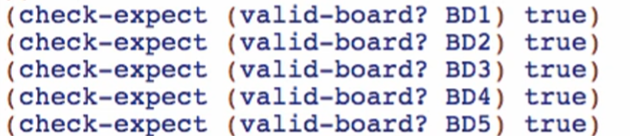
**Signature, purpose, and stub**



Work your way because this is the last (so far) function

**Examples**

Examples that are from our constant definitions



Examples with modified values of our existing boards

* Invalid in row
* Invalid in column
* Invalid in box

