Macro-Validation Assignment – sm11326 Mettu Sai Nishanth

Compiler used : https://rextester.com/l/common_lisp_online_compiler
Online LISP Compiler (For Testing my code since its an open-end assignment)

Test cases :

1. VALID FUNCTION TEST - (PARAMS, NAMES, & CASE SENSITIVITY (x y z X Y Z))

(validate `(defun abc123 (X y Z) (+ x y))) ;

(validate `(defun fact (x y) (+ x y))) ;

(validate `(defun abc123 (X y Z) (+ x y)));

```
DEFUN : Valid Keyword! Expected keyword DEFUN found.
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.


DEFUN : Valid Keyword! Expected keyword DEFUN found.
FACT : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.


DEFUN : Valid Keyword! Expected keyword DEFUN found.
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.
```

These are valid, irrespective of Case sensitivity, as shown above!

2. INVALID FUNCTION NAMES TEST (FUNCTION SHOULD BE AN ATOM)/ CANNOT START WITH A DIGIT

(validate `(defun 123 (x y z) "abc"))

(validate `(define 0.0 (a b c)))

(validate `(defun 123xyz (x x) 12))

These will fail because the Function cannot start with a digit, it should be an Atom

```
DEFUN : Valid Keyword! Expected keyword DEFUN found.
123 : Not a valid function name!
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Invalid Function Definition.


The form does not have the correct number of items.

First item is not defun, hence not a valid function!
0.0 : Not a valid function name!
A: Parameters are not allowed. Not X, Y, or Z!
B: Parameters are not allowed. Not X, Y, or Z!
C: Parameters are not allowed. Not X, Y, or Z!
Invalid Function Body. No Body Present!
Invalid Function Definition.




DEFUN : Valid Keyword! Expected keyword DEFUN found.
123XYZ : Function name cannot start with a digit!
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Invalid Function Definition.
```

3. INVALID NUMBER OF LIST ITEMS TEST

(validate '(defun xyc abd 234 (n)))

(validate '(defun xyc (n)(x y)(x x)))

These should fail because the function doesn’t have valid number of Items present!

```
The form does not have the correct number of items.

DEFUN : Valid Keyword! Expected keyword DEFUN found.
XYC : Valid function name.
Valid Function Body
Invalid Function Definition.


The form does not have the correct number of items.

DEFUN : Valid Keyword! Expected keyword DEFUN found.
XYC : Valid function name.
N: Parameters are not allowed. Not X, Y, or Z!
Valid Function Body
Invalid Function Definition.
```

4. EMPTY PARAMETERS TEST – IT SHOULD ONLY ALLOW, X, Y, Z, x,y,z or nothing(empty) as the parameters

(validate `(defun func-no-params () (+ 1 2))) ;

(validate `(defun another-func () "Hello, world!")) ;

(validate `(defun empty-func () nil)) ;

(validate `(defun no-params () (format t "No parameters required."))) ;

```
DEFUN : Valid Keyword! Expected keyword DEFUN found.
FUNC-NO-PARAMS : Valid function name.
No parameters present, it is still legal!
Valid Function Body
Valid Function Definition.



DEFUN : Valid Keyword! Expected keyword DEFUN found.
ANOTHER-FUNC : Valid function name.
No parameters present, it is still legal!
Valid Function Body
Valid Function Definition.



DEFUN : Valid Keyword! Expected keyword DEFUN found.
EMPTY-FUNC : Valid function name.
No parameters present, it is still legal!
Invalid Function Body. No Body Present!
Invalid Function Definition.



DEFUN : Valid Keyword! Expected keyword DEFUN found.
NO-PARAMS : Valid function name.
No parameters present, it is still legal!
Valid Function Body
Valid Function Definition.
```

```
DEFUN : Valid Keyword! Expected keyword DEFUN found.
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.


DEFUN : Valid Keyword! Expected keyword DEFUN found.
FACT : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.


DEFUN : Valid Keyword! Expected keyword DEFUN found.
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Valid Function Definition.
```

It is still legal if we are passing Empty parameters, and X, Y, Z, x, y, z

5. EMPTY BODY TEST – IT CANNOT HAVE EMPTY BODY – SELF DECLARED RULE!

(validate `(defun empty-func () nil)) ;

(validate `(defun empty-func () ())) ;

```
DEFUN : Valid Keyword! Expected keyword DEFUN found.
EMPTY-FUNC : Valid function name.
No parameters present, it is still legal!
Invalid Function Body. No Body Present!
Invalid Function Definition.



DEFUN : Valid Keyword! Expected keyword DEFUN found.
EMPTY-FUNC : Valid function name.
No parameters present, it is still legal!
Invalid Function Body. No Body Present!
Invalid Function Definition.
```

It will fail because I have enforced Failure if there is no body, but we can easily modulate this based on our needs, since the professor said we can create our own rules, I created one just to experiment!

6. FIRST VALUE IS "defun" TEST

The first value should always be defun! Or else it should fail

(validate `(func abc123 (X y Z) (+ x y))) ;

(validate `(func fact (x y) (+ x y))) ;

(validate `(func abc123 (X y Z) (+ x y)));

```
First item is not defun, hence not a valid function!
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Invalid Function Definition.



First item is not defun, hence not a valid function!
FACT : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Invalid Function Definition.



First item is not defun, hence not a valid function!
ABC123 : Valid function name.
X: Parameters Allowed. X, Y, and Z are the only valid parameters.
Y: Parameters Allowed. X, Y, and Z are the only valid parameters.
Z: Parameters Allowed. X, Y, and Z are the only valid parameters.
Valid Function Body
Invalid Function Definition.
```

It is failing because it doesn't start with defun!

7. PASSING OTHER PARAMETERS TEST !

(validate `(func abc123 (P Q R) (+ x y))) ;

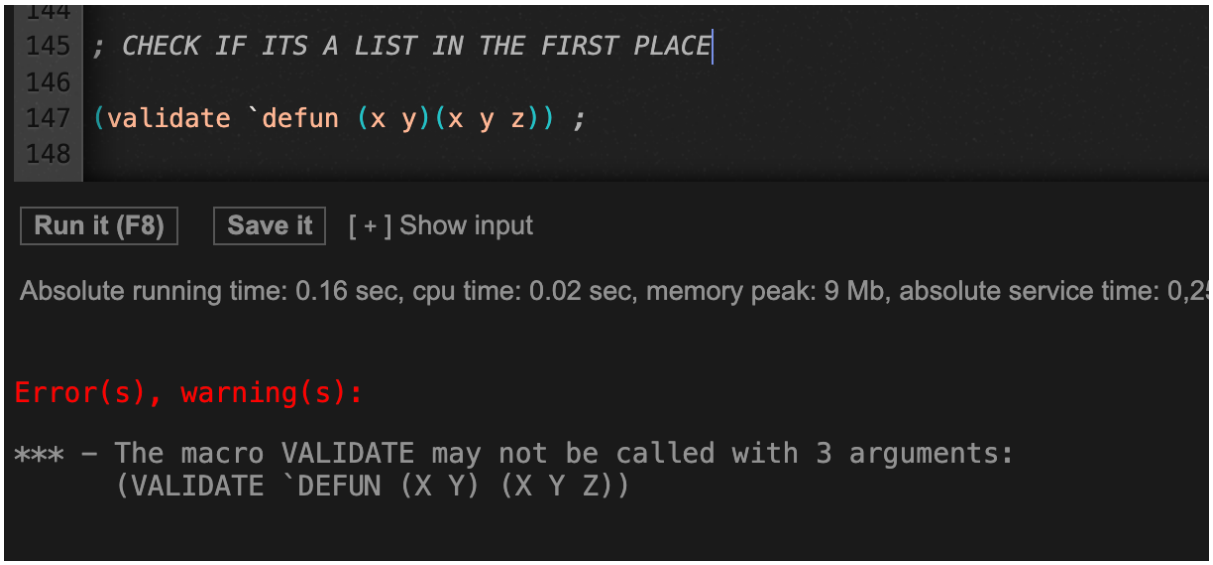
(validate `(func fact (p q) (+ x y))) ;

(validate `(func abc123 (a b c) (+ x y)));

```
First item is not defun, hence not a valid function!
ABC123 : Valid function name.
P: Parameters are not allowed. Not X, Y, or Z!
Q: Parameters are not allowed. Not X, Y, or Z!
R: Parameters are not allowed. Not X, Y, or Z!
Valid Function Body
Invalid Function Definition.



First item is not defun, hence not a valid function!
FACT : Valid function name.
P: Parameters are not allowed. Not X, Y, or Z!
Q: Parameters are not allowed. Not X, Y, or Z!
Valid Function Body
Invalid Function Definition.



First item is not defun, hence not a valid function!
ABC123 : Valid function name.
A: Parameters are not allowed. Not X, Y, or Z!
B: Parameters are not allowed. Not X, Y, or Z!
C: Parameters are not allowed. Not X, Y, or Z!
Valid Function Body
Invalid Function Definition.
```

Here we passed P, Q, R , a, b, c etc… and it fails as those Parameters aren’t allowed.

PERSONAL CHECK : IF ITS NOT A LIST ITEM, THE MACRO AUTOMATIOCALLY THROWS AN IMMEDIATE ERROR



since we need a list enclosure to validate the nth 0 , nth 1, nth 2. Nth 3 enclosures while validating.. so it anyways would fail and be illegal.

THANK YOU

SAI NISHANTH METTU
sm11326
NYU TANDON SCHOOL OF ENGINEERING