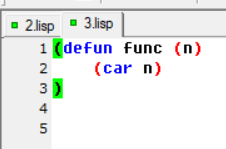
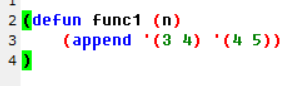
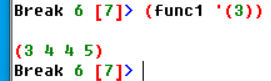
1)



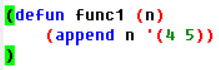


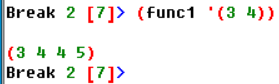
-----------------------------------------------------------------------------------------------



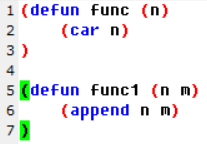


-----------------------------------------------------------------------------------------------



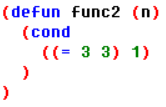


-----------------------------------------------------------------------------------------------



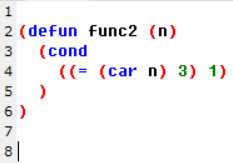


-----------------------------------------------------------------------------------------------



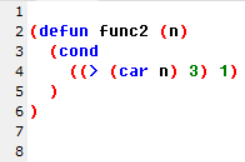


----------------------------------------------------------------------------------------------



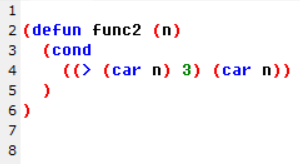


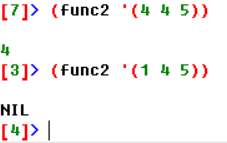
-----------------------------------------------------------------------------------------------



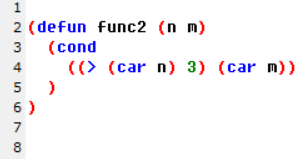


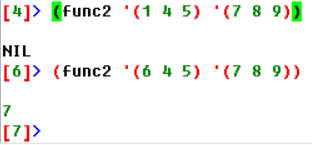
-----------------------------------------------------------------------------------------------



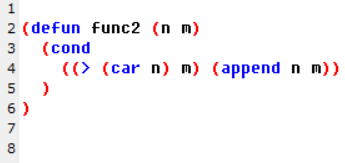


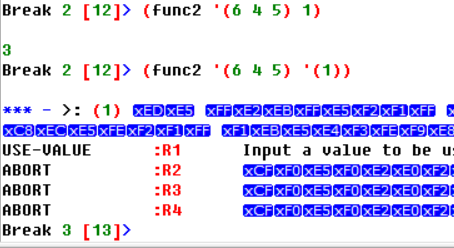
-----------------------------------------------------------------------------------------------



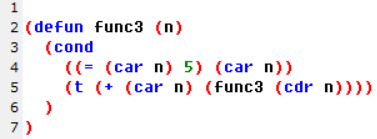


-----------------------------------------------------------------------------------------------



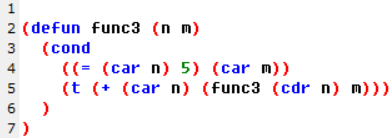


-----------------------------------------------------------------------------------------------



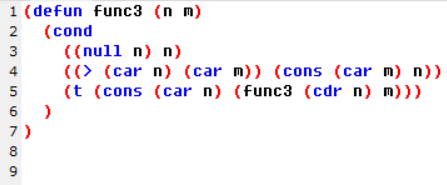


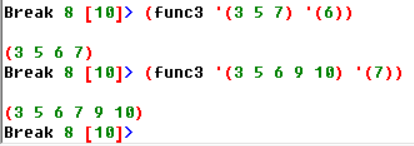
-----------------------------------------------------------------------------------------------



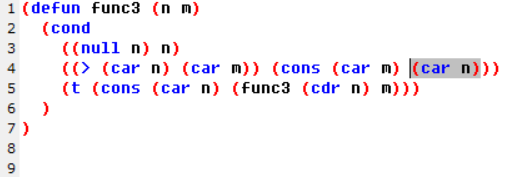


-----------------------------------------------------------------------------------------------



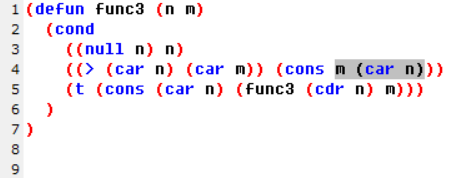


Есть пару вопросиков



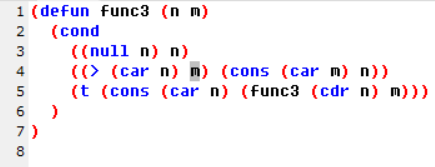


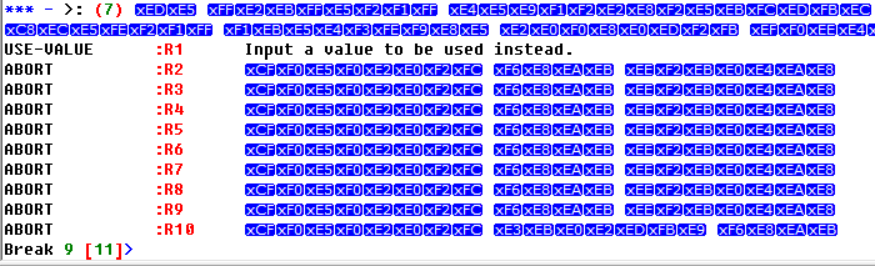
Почему так?





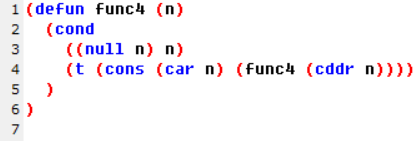
Или так:

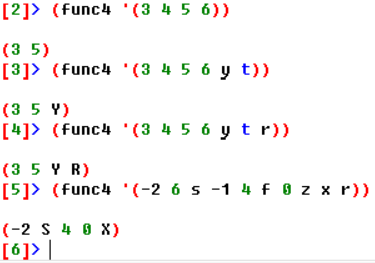




Какого х?

2)





(defun func (n lst)

(cond

((> (car lst) n) (append n lst))

(t (append car(lst) func(n lst)))

)

)

(defun func (n)

(car n)

)

(defun func1 (n m)

(append n m)

)

(defun func3 (n m)

(cond

((null n) n)

((= (car n) m) (car m))

(t (cons (car n) (func3 (cdr n) m)))

)

)

(defun func2 (n m)

(cond

((> (car n) m) (append n m))

(t (append (car n) (func2 ((cdr n) m))))

)

)

(defun func6 (n)

(cond

((= (car n) 5) 2)

(t (+ 1 (func6 (car n))))

)

)

(defun func3 (n m)

(cond

((null n) n)

((>= (car n) (car m)) (cons (car m) n))

(t (cons (car n) (func3 (cdr n) m)))

)

)

(defun func2 (lst)

(if (atom lst) ; является ли lst атомарным элементом

0 ; lst - атом, возвращается 0

(1+ (reduce #'max (mapcar #'func2 lst))) ;

)

)