fox01{ wjk232}7: clisp

i i i i i i i ooooo o ooooooo ooooo ooooo

I I I I I I I 8 8 8 8 8 o 8 8

I \ `+' / I 8 8 8 8 8 8

\ `-+-' / 8 8 8 ooooo 8oooo

`-\_\_|\_\_-' 8 8 8 8 8

| 8 o 8 8 o 8 8

------+------ ooooo 8oooooo ooo8ooo ooooo 8

Welcome to GNU CLISP 2.49 (2010-07-07) <http://clisp.cons.org/>

Copyright (c) Bruno Haible, Michael Stoll 1992, 1993

Copyright (c) Bruno Haible, Marcus Daniels 1994-1997

Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998

Copyright (c) Bruno Haible, Sam Steingold 1999-2000

Copyright (c) Sam Steingold, Bruno Haible 2001-2010

Type :h and hit Enter for context help.

[1]> (cd "/home/wjk232/cs3723/assign2")

#P"/home/wjk232/cs3723/assign2/"

[2]> (load "p2wjk232.lisp" :echo T :print T)

;; Loading file p2wjk232.lisp ...

(defun SETDIFF (set1 set2)

(COND ( (NULL set1)())

( (MEMSET (CAR set1) set2)

(SETDIFF (CDR set1) set2))

(T (CONS (CAR set1) (SETDIFF (CDR set1) set2)))

)

)

SETDIFF

(defun MAKESET (set1)

(COND ( (NULL set1)() )

( (MEMSET (CAR set1) (CDR set1))

(MAKESET (CDR set1)))

(T (CONS (CAR set1) (MAKESET (CDR set1))))

)

)

MAKESET

(defun ATOMICLIST (set1)

(COND ( (NULL set1) (CONS () set1))

( (ATOM set1) (LIST set1))

( (NULL (CDR set1)) (ATOMICLIST (CAR set1)))

(T (APPEND (ATOMICLIST (CAR set1)) (ATOMICLIST (CDR set1))))

)

)

ATOMICLIST

(defun COUNTUNIQNONNIL (set1)

(COND ( (NULL set1) 0)

( (ATOM set1) 1)

(T (+ (COUNTUNIQNONNIL (CAR (MAKESET (ATOMICLIST set1))))

(COUNTUNIQNONNIL (CDR (MAKESET (ATOMICLIST set1))))))

)

)

COUNTUNIQNONNIL

(defun MEMSET (atm L)

(cond ( (NULL L) NIL )

( (EQL atm (CAR L)) T )

( T (MEMSET atm (CDR L)))

)

)

MEMSET

;; Loaded file p2wjk232.lisp

T

[3]> (setdiff '(x y z) '(w x z))

(Y)

[4]> (setdiff '(w x y z) '(x y))

(W Z)

[5]> (setdiff '(x y z) '(x y z))

NIL

[6]> (setdiff '(w r y i g x h t z) '(w x y z))

(R I G H T)

[7]> (makeset '(a b b a d a b b a))

(D B A)

[8]> (makeset '(e a t s e t))

(A S E T)

[9]> (atomiclist '(A (B F (H) G) J))

(A B F H G J)

[10]> (atomiclist '(L () (I () S) (((P ()))) ))

(L NIL I NIL S P NIL)

[11]> (countUniqNonNIL '((H (A) ()) P (P Y D) (A (Y S))))

6

[12]> (countUniqNonNIL '( (T (H A T)) (I S) I (T) ))

5

[13]>