

General Information

Author: Datasoft

Language: LISP

Compiler/Interpreter: InterLisp/65

{code}

```
(EDIT E EF H L S C X PASTE EQUAL GETCMD MAPCAR COMMANDS PSET SETP R A D B LI RE CONZ P
(DEFINEQ EDIT '(LAMBDA (S-EXP) (PROG (CURR PREV X) T (PSET S-EXP) (SETQ PREV (LIST S-E
)
(DEFINEQ E '(LAMBDA NIL ((LAMBDA (X) (PROGN (SETQ S-EXP (COND ((EVAL X)) (T (POKE 128
)
(DEFINEQ EF '(LAMBDA NIL ((LAMBDA (X) (PROGN (SETQ S-EXP (COND ((GETD X)) (T (POKE 128
)
(DEFINEQ H '(LAMBDA NIL (PRIN1 (LIST (QUOTE COMMANDS:) COMMANDS)))
)
(DEFINEQ L '(LAMBDA NIL ((LAMBDA (FILE) (PROGN (LOAD FILE) (PRIN1 (QUOTE "Loaded--> "
)
(DEFINEQ S '(LAMBDA NIL (PROGN (SAVE (EVAL (GETCMD (QUOTE "Properties> "))) (GETCMD (Q
)
(DEFINEQ C '(LAMBDA NIL (DIR (READ)))
)
(SETQ X 'S
)
(DEFINEQ X '(LAMBDA NIL (PROGN (PASTE CURR (READ) (READ)) (PSET CURR)))
)
(DEFINEQ PASTE '(LAMBDA (X Y Z) (COND ((EQUAL X)) ((ATOM X)) ((EQUAL (CAR X) Y) (RPLAC
)
(DEFINEQ EQUAL '(LAMBDA (X Y) (COND ((ATOM X) (EQ X Y)) ((ATOM Y) (EQ X Y)) ((AND (EQU
)
(DEFINEQ GETCMD '(LAMBDA (X) (PROGN (POKE 128 0) (TERPRI) (PRIN1 X) (READ)))
)
(DEFINEQ MAPCAR '(LAMBDA (X FN) (COND ((EQ X) NIL) (T (CONS (APPLY* FN (CAR X)) (MAPCAR
)
(SETQ COMMANDS '(A D B P R DEL I G LI RE PP PRE C E EF X S L H PU LOCK UNLOCK)
)
(DEFINEQ PSET '(LAMBDA (L) (PRINT (CAR (SETQ CURR L))))
)
(DEFINEQ SETP '(LAMBDA (FUNCTION) (COND ((ATOM (APPLY* FUNCTION CURR)) (PRINT (QUOTE E
)
(DEFINEQ R '(LAMBDA NIL (PRINT (CAR (RPLACA CURR (READ))))))
)
(SETQ A '(A)
)
(DEFINEQ A '(LAMBDA NIL (SETP (QUOTE CDR)))
)
(DEFINEQ D '(LAMBDA NIL (SETP (QUOTE CAR)))
)
(DEFINEQ B '(LAMBDA NIL (COND ((EQ (CDR PREV)) (PRINT (QUOTE Top_of_list)) (PSET CURR)
)
(DEFINEQ LI '(LAMBDA NIL (PSET (RPLACA CURR (LIST (CAR CURR))))))
)
(DEFINEQ RE '(LAMBDA NIL (COND ((ATOM (CAR CURR)) (PRIN1 (QUOTE Not_a_list)) (TERPRI))
)
(DEFINEQ CONZ '(LAMBDA (X Y Z) (RPLACA (RPLACD X Z) Y))
)
(DEFINEQ PRE '(LAMBDA NIL (PSET (RPLACA CURR (CONS (READ) (LIST (CAR CURR))))))
)
(DEFINEQ I '(LAMBDA NIL (PROG NIL (RPLACD CURR (CONS (READ) (CDR CURR)) (A)))
```

```

)
(DEFINEQ DEL '(LAMBDA NIL (PROG NIL ((LAMBDA (X) (COND ((EQ CURR X) (CONZ CURR (CAR (C
)
(DEFINEQ G '(LAMBDA NIL (PROG (X Y) (SETQ Y CURR) L (SETQ X (GETCMD (QUOTE "Group> ")))
)
(DEFINEQ PP '(LAMBDA NIL (PPRINT S-EXP 2))
)
(DEFINEQ P '(LAMBDA NIL (PPRINT (CAR CURR) 2))
)
(DEFINEQ PPRINT '(LAMBDA (X IND) (PROG (TFLG) (PPAUX X IND) (TERPRI)))
)
(DEFINEQ PPAUX '(LAMBDA (X INDTN) (COND ((ATOM X) (PRIN2 X) (SETQ TFLG)) ((ATOM (CAR X)
)
(DEFINEQ MULTARGS '(LAMBDA (X INDTN L) ((LAMBDA (INDTN2) (PROGN (TABRET INDTN) (PRIN1
)
(DEFINEQ PPARGS '(LAMBDA (X INDTN) (PROG NIL LOOP (COND ((EQ X) (RETURN))) (PRIN1 BLAN
)
(DEFINEQ TABRET '(LAMBDA (N) (PROG NIL (COND (TFLG (RETURN)) ((TERPRI))) (SETQ TFLG T)
)
(SETQ FORMATS '((COND 6) (LAMBDA 2) (PROG 6) (PROGN 7) (AND 5) (OR 4) (MAPCAR 8) (NLAM
)
(SETQ LPAR '"("
)
(SETQ BLANK '" "
)
(SETQ RPAR ')
)
(SETQ LINE-WIDTH '39
)
(DEFINEQ NCONC '(LAMBDA (X Y) (PROGN (RPLACD (LAST X) Y) X))
)
(DEFINEQ PU '(LAMBDA NIL ((LAMBDA (X) (PROGN (COND ((EQ (GETCMD (QUOTE "Enter 'Y' to d
)
(DEFINEQ LOCK '(LAMBDA NIL ((LAMBDA (X) (PROGN (XIO 35 7 0 0 X) (CLOSE 7))) (READ)))
)
(DEFINEQ UNLOCK '(LAMBDA NIL ((LAMBDA (X) (PROG (STAT) (SETQ STAT (XIO 36 7 0 0 X)) (C
)
(DEFINEQ GETLST '(LAMBDA (X) (COND ((ATOM X) (LIST X)) (X)))
)
NIL

```

{code}