





001 (DEFPROP TRACE  
002 (LAMBDA (L)  
003 (PROG (W X Y Z)  
004 B (COND ((NULL L) (RETURN Y)))  
005 (SETQ X (SETQ Z (CAR L)))  
006 (SETQ L (CDR L))  
007 C (COND ((NULL (CDR Z)) (GO B))  
008 ((EQ (CADR Z) (QUOTE TRACE)) (GO B))  
009 ((MEMBER (CADR Z) (QUOTE (EXPR FEXPR SUBR FSUBR)))  
010 (SETQ Z (CDR Z)))  
011 ((SETQ Z (CDDR Z)) (GO C)))  
012 (SETQ Y (INCONC Y (LIST X)))  
013 (RPLACD (CDDR (SETQ W (GENSYM))) Z)  
014 (RPLACD  
015 X  
016 (CONS  
017 (QUOTE TRACE)  
018 (CONS  
019 W  
020 (CONS  
021 (QUOTE FEXPR)  
022 (CONS  
023 (LIST  
024 (QUOTE LAMBDA)  
025 (QUOTE (/L))  
026 (INCONC  
027 (LIST (SUBST W X (QUOTE PROG))  
028 (QUOTE (/B /C))  
029 (LIST (SUBST W X (QUOTE PRINT))  
030 (LIST (SUBST W X (QUOTE QUOTE))  
031 (LIST (QUOTE ENTERING) X))))  
032 (APPEND  
033 (SUBST  
034 W  
035 X  
036 (COND  
037 ((OR (EQ (CAR Z) (QUOTE FSUBR))  
038 (EQ (CAR Z) (QUOTE FEXPR)))  
039 (QUOTE (PRINT (SETQ /-C /-L))))  
040 (T  
041 (QUOTE  
042 ((PRINT (SETQ /-B (EVAL (CONS (QUOTE LIST) /-L)))  
043 /-D  
044 (COND ((NULL /-B) (GO /-E))  
045 (SETQ /-C (INCONC /-C  
046 (LIST (CONS (QUOTE QUOTE)  
047 (LIST (CAR /-B))))))  
048 (SETQ /-B (CDR /-B))  
049 (GO /-D))))))  
050 (LIST (QUOTE /-E)  
051 (SUBST W X (LIST (QUOTE SETQ)  
052 (QUOTE /-B)  
053 (LIST (QUOTE EVAL)  
054 (LIST (QUOTE CONS)  
055 (LIST (QUOTE QUOTE)  
056 W)  
057 (QUOTE /-C))))))  
058 (LIST (SUBST W X (QUOTE PRINT))  
059 (LIST (SUBST W X (QUOTE QUOTE)))

060 (LIST (QUOTE VALUE) (QUOTE OF) X)))  
061 (SUBST W X (QUOTE (RETURN (PRINT /-B))))))  
062 (CDR X))))))  
063 (GO B)))  
064 FEXPR)  
065 (DEFPROP UNTRACE  
066 (LAMBDA (L)  
067 (PROG (X Y)  
068 B (COND ((NULL L) (RETURN Y)))  
069 (SETQ X (CAR L))  
070 C (COND ((NULL (CDR X)) (GO D))  
071 ((EQ (QUOTE TRACE) (CADR X))  
072 (RPLACD X (CDDR (CDDR (CDR X))))  
073 ((SETQ X (CDDR X)) (GO C)))  
074 (SETQ Y (CONS (CAR L) Y))  
075 (SETQ L (CDR L))  
076 (GO B)))  
077 FEXPR)  
078  
079  
080  
081 (LAMBDA (IBASE) (MAPCAR (FUNCTION (LAMBDA (L) (EVAL (CONS (QUOTE DEFPROP) L))) (EVAL (READ)))) (SETQ L  
082  
083 (LAMBDA (G F) (PROG2 (G (QUOTE / ) 17) (READ) (G (QUOTE / ) 7)))  
084 (FUNCTION (LAMBDA (L N) (EQ (EVAL (LIST (QUOTE ARRAY) L T N)) (F N)))  
085 (FUNCTION (LAMBDA (N) (AND (NOT (MINUSP N))  
086 (APPEND (GET L (QUOTE PNAME))  
087 (CADDR (STORE (L N) (LIST (CAR T) (QUOTE PNAME) (F (SUB1 N)))))))  
088 (CADDR (STORE (L N) (LIST (CAR T) (QUOTE PNAME) (F (SUB1 N)))))) (CADR L  
089  
090 (EVAL (LAMBDA (L) (PROG2 (PUTPROP (CADDR L) (SUBST (CAR L) (CADR L) (CADDR (SETQ L (CADDR L)))) (CADR L  
091 (GRINDEF  
092 (LAMBDA (L)  
093 (ERR (LAMBDA (H)  
094 (MAPC (FUNCTION (LAMBDA (C)  
095 (COND ((ATOM C)  
096 (MAPC (FUNCTION (LAMBDA (F)  
097 (AND (SETQ L (GET C F))  
098 (TURPRI)  
099 (TURPRI)  
100 (SPRINT (LIST (QUOTE  
101 LINEL  
102 0))))  
103 H))  
104 ((SETQ H (APPEND C H))))))  
105 L))  
106 (QUOTE (EXPR FEXPR VALUE MACRO))))  
107 FEXPR)  
108 (TURPRI (LAMBDA NIL (PROG2 (TERPRI) (SETQ LINCT (REMAINDER (PLUS 73 LINCT) 74)))) EXPR)  
109 (SPRINT (LAMBDA (L N M)  
110 (PROG (F G H)  
111 (AND (LESSP N CHRCT  
112 (\*DIF CHRCT N))  
113 (PRINC /  
114 (\*DIF CHRCT N))  
115 (PRINC /  
116 (\*DIF CHRCT N))  
117 (AND (OR (LESSP (PLUS M -1) (FLATSIZE L)) CHRCT) (ATOM L)) (RETURN (PRIN1 L)))  
118 (SETQ F (EQ (CAR L) (QUOTE PROG)))  
119 (PRINC (QUOTE /))  
120 (ERRSET (AND (NOT (ATOM (CDR L)))  
121 (DIFFERENCE CHRCT (FLATSIZE (CAR L)) 1))  
122 (SETQ N (MAXPAN (CDR L) (DIFFERENCE CHRCT (FLATSIZE (CAR L)) N))  
123 (OR (ATOM (CAR L)) (NOT (LESSP (MAXPAN (CDR L) CHRCT) N))))



```

120 (PROG NIL (ERRSET (SETQ G
121 (LESSP (MAXPAN (LAST L)
122 (*DIF (PLUS (FLATSIZE (LAST L)) CHRCT)
123 N)))
124 (PRINT (CAR L))
125 (PRINT (QUOTE / ))
126 (AND (CDR (SETQ L (CDR L))) G (GO A))))
127 (SETQ N CHRCT)
128 (SETQ M (MEMQ (CAR L) (QUOTE (DEFPROP LABEL LAMBDA))))
129 B
130 (SPRINT (CAR L)
131 (COND ((SETQ G (AND F (CAR L) (ATOM (CAR L))) (PLUS N 5)) (N))
132 (COND ((NULL (SETQ L (CDR L))) (ADD1 M)) ((ATOM L) (PLUS 4 M (FLAT
133 (COND ((ATOM L) (AND L (PRINT (QUOTE / / )) (PRINT L)) (RETURN (PRINC (Q
134 (COND ((H (SETQ H NIL) (PRINC (QUOTE / ))
135 (COND ((OR (LESSP CHRCT N) (AND G (ATOM (CAR L))) (TURPRI)))
136 (GO B)))
137 (GO B)))
138 (GO B)))
139 (MAXPAN (LAMBDA (L N) (PROG (G)
140 (SETQ G 0)
141 A (SETQ G (PLUS G (PANMAX (CAR L) N (COND ((NULL (SETQ L (CDR L))) (A
142 ((ATOM L) (PLUS M 4 (FL
143 (G)))
144 (AND (ATOM L) (RETURN G))
145 (GO A)))
146 (GO A)))
147 (EXPR)
148 (PANMAX (LAMBDA (L N M) (COND ((LESSP (PLUS M -1 (FLATSIZE L)) N) 1)
149 ((OR (LESSP N 3) (ATOM L)) (ERR(QUOTE(50))))
150 ((AND (NOT (ATOM (CDR L)))
151 (ATOM (CAR L))
152 (SETQ N (DIFFERENCE N 1 (FLATSIZE (CAR L)))
153 (SETQ L (CDR L))
154 NIL)
155 ((MAXPAN L (SUB1 N))))))
156 (EXPR)
157 (GRIND1 (LAMBDA (L)
158 (PROG (LINE1 G 10 10 10)
159 (SETQ LINE1 120)
160 (EVAL (CONS (QUOTE UREAD) L))
161 (EVAL (CONS (QUOTE UWRITE) (AND (CDR L) (LAST L))))
162 (SETQ 10 (SETQ LINE1 73))
163 (LOC RQ)
164 A (COND ((LESSP (TURPRI)
165 (CAR (ERRSET (REMAINDER (PANMAX (SETQ G (CAR (OR (ERRSET (REA
166 (RETURN (CONS (QUOT
167 CHRCT
168 0)
169 74)))
170 (SETQ LINE1 73)
171 (TYO 14)
172 (SETQ CHRCT LINE1)
173 ((TURPRI)))
174 (SPRINT G CHRCT 0)
175 (GO A)))
176 (FEXPR)) (QUOTE EDIT)
177
178
179

```

```

180 (COND ((GET (QUOTE COMPILING) (QUOTE NOW)))
181 (SETQ SCOUNT 0)
182 (DEFPROP GENSYM
183 (LAMBDA NIL
184 (MAKNAM
185 (APPEND
186 (EXPLODE (Q 5))
187 (EXPLODE (SETQ SCOUNT
188 (ADD1 SCOUNT))))))
189 (EXPR)
190 (SPECIAL
191 SCURSOR SCI $IND SOB
192 $AM SVT $FF $SL $LP $RP $SP $PT $H $CR
193 $LF $CH $TB $RO $CO)))
194 (DEFPROP MCONS
195 (LAMBDA (L)
196 (COND ((NULL (CDR L)) (CADR L))
197 (LIST (Q CONS)
198 (CADR L)
199 (CONS (CAR L)
200 (CDR L))))))
201 (MACRO)
202
203 (DEFPROP Q
204 (LAMBDA (L) (CONS (QUOTE QUOTE)
205 (CDR L)))
206 (MACRO)
207
208 (MAPC
209 (FUNCTION
210 (LAMBDA (X)
211 (SET
212 (CAR X)
213 (INTERN
214 (LIST (CAR NIL)
215 (Q PNAME)
216 (LIST (CDR (LSH (CADR X)
217 35))))))
218 (Q (LISTB 11) (SLF 12)
219 (SVT 13)
220 (SFF 14)
221 (SCR 15)
222 (SRD 177)
223 (SSP 40)
224 (SCQ 54)
225 (SLP 50)
226 (SRP 51)
227 (SPT 56)
228 ($AM 33)
229 ($SL 57)))
230
231 (SETQ SCURSOR NIL)
232
233 (ARRAY $LOCAR T 10)
234
235 (ARRAY $CONDAR T 10)
236
237 (DEFPROP SCURLOC
238 (LAMBDA (L) (CONS (Q CAR) (CDR L)))
239

```



```

240      MACRO)
241
242      (DEFPROP $CUREXPR
243      (LAMBDA (L)
244      (CONS (Q CAAR) (CDR L)))
245      MACRO)
246
247      (DEFPROP $CURCON
248      (LAMBDA (L) (CONS (Q CDR) (CDR L)))
249      MACRO)
250
251      (DEFPROP $RIGHT
252      (LAMBDA (C)
253      (COND ((AND C ($CURLOC C))
254      (CONS ($CURCON ($CURLOC C))
255      ($CURCON C))))
256      EXPR)
257
258      (DEFPROP $LEFT
259      (LAMBDA (C)
260      (PROG
261      (R S)
262      (COND ((EQ (SETQ S
263      ($CUREXPR ($CURCON C)))
264      (SETQ R ($CURLOC C)))
265      (RETURN NIL)))
266      LOOP
267      (COND ((EQ ($CURCON S) R)
268      (RETURN (CONS S ($CURCON C))))
269      (SETQ S (CDR S))
270      (GO LOOP)))
271      EXPR)
272
273      (DEFPROP $DOWN
274      (LAMBDA (C)
275      (COND
276      ((AND C
277      ($CURLOC C)
278      (NOT (ATOM ($CUREXPR C)))
279      (CONS ($CUREXPR C) C)))
280      EXPR)
281
282      (DEFPROP $UP
283      (LAMBDA (C)
284      (COND ((NULL C) NIL)
285      ((LAMBDA (X)
286      (COND ((AND X ($CURCON X)
287      X)))
288      ($CURCON C))))
289      EXPR)
290
291      (DEFPROP $SKILL
292      (LAMBDA (C)
293      (COND
294      (($CURLOC C)
295      (PROG
296      (CUR)
297      (COND
298      (($CURLOC (SETQ CUR ($RIGHT C))
299      (RPLACD (RPLACA ($CURLOC C)

```

```

300      (CAR ($CURLOC CUR)))
301      (CDR ($CURLOC CUR)))
302      (RETURN C))
303      ((SETQ CUR ($LEFT C))
304      (RPLACD ($CURLOC CUR) NIL)
305      (RETURN ($RIGHT CUR)))
306      ((SETQ CUR ($UP C))
307      (RPLACA ($CURLOC CUR) NIL)
308      (RETURN CUR)))
309      (RETURN NIL)))
310      EXPR)
311
312      (DEFPROP $OPER
313      (LAMBDA (OP N)
314      (PROG (CUR)
315      (SETQ CUR $CURSOR)
316      LOOP (COND ((AND (*GREAT (SETQ N
317      (SUB1 N))
318      -1)
319      (SETQ CUR (OP CUR)))
320      (SETQ $CURSOR CUR)
321      (GO LOOP)))
322      (RETURN ($CURLOC $CURSOR)))
323      EXPR)
324
325      (DEFPROP $R
326      (LAMBDA (L)
327      ($OPER (Q $RIGHT)
328      (COND (L (CAR L))
329      (1))))
330      FEXPR)
331
332      (DEFPROP $L
333      (LAMBDA (L)
334      ($OPER (Q $LEFT)
335      (COND (L (CAR L))
336      (1))))
337      FEXPR)
338
339      (DEFPROP $D
340      (LAMBDA (L)
341      ($OPER (Q $DOWN)
342      (COND (L (CAR L))
343      (1))))
344      FEXPR)
345
346      (DEFPROP $U
347      (LAMBDA (L)
348      ($OPER (Q $UP)
349      (COND (L (CAR L))
350      (1))))
351      FEXPR)
352
353      (DEFPROP $K
354      (LAMBDA (L)
355      ($OPER (Q $SKILL)
356      (COND (L (CAR L))
357      (1))))
358      FEXPR)
359

```



```

360 (DEFPROP $INSERT
361 (LAMBDA (S C)
362 (COND
363 ((NULL C)
364 ($RIGHT ($DOWN (LIST (LIST (LIST S))))))
365 (($CURLOC C)
366 (COND
367 (($CURLOC C)
368 (RPLACA
369 (RPLACD ($CURLOC C)
370 (CONS (CAR ($CURLOC C))
371 (CDR ($CURLOC C))))
372 S)
373 ($RIGHT C))
374 ((SETQ C ($LEFT C))
375 (RPLACD ($CURLOC C) (LIST S))
376 ($RIGHT ($RIGHT C))))))
377 EXPR)
378
379 (DEFPROP $S1
380 (LAMBDA (CMS C)
381 (COND ((OR (NULL CMS)
382 (NULL C))
383 C)
384 ((CAR CMS) (CDR CMS)
385 C)))
386 EXPR)
387
388 (DEFPROP $ATH
389 (LAMBDA (CMS C)
390 (COND ((AND ($CURLOC C)
391 (EQUAL (CAR CMS)
392 ($CUREXPR C)))
393 ($S1 (CDR CMS)
394 ($RIGHT C))))))
395 EXPR)
396
397 (DEFPROP $LPR
398 (LAMBDA (CMS C)
399 (SS1 CMS ($DOWN C)))
400 EXPR)
401
402 (DEFPROP $RPR
403 (LAMBDA (CMS C)
404 (COND ((NOT ($RIGHT C))
405 (SS1 CMS ($RIGHT ($SUP C))))))
406 EXPR)
407
408 (DEFPROP $SEARCH2
409 (LAMBDA (CMS C)
410 (PROG (SC1)
411 A (COND ((SETQ C
412 (SS1 CMS (SETQ SC1 C)))
413 ((SETQ C ($DOWN SC1)) (GO A))
414 ((SETQ C ($RIGHT SC1)) (GO A))
415 ((SETQ C ($RIGHT ($SUP SC1))
416 (GO A)))
417 (RETURN C))))
418 EXPR)
419

```

```

420 (DEFPROP $SEARCH1
421 (LAMBDA ()
422 (PROG
423 (CH CMS OP NAME CUR)
424 S1 (SETQ NAME (SETQ OP NIL))
425 S2 (SETQ CH (READCH))
426 (COND ((EQ CH SAM) (GO SAM))
427 ((EQ CH SLP) (GO SLP))
428 ((EQ CH SRP) (GO SRP))
429 ((EQ CH SSP) (GO SSP))
430 ((MEMQ CH (LIST $CR $LF)) (GO S3)))
431 (SETQ NAME (CONS CH NAME))
432 S3 (GO S2)
433 SLP (SETQ OP (Q SLP))
434 (GO SSP)
435 SRP (SETQ OP (Q SRP))
436 SSP (AND
437 NAME
438 (SETQ CMS
439 (MCONS (READLIST (REVERSE NAME))
440 (Q $ATH)
441 CMS)))
442 (AND OP (SETQ CMS (CONS OP CMS)))
443 (GO S1)
444 SAM (AND
445 NAME
446 (SETQ CMS
447 (MCONS (READLIST (REVERSE NAME))
448 (Q $ATH)
449 CMS)))
450 (AND (SETQ CUR ($SEARCH2 (REVERSE CMS)
451 SCURSOR))
452 (SETQ SCURSOR CUR))
453 (RETURN ($CURLOC SCURSOR)))
454 EXPR)
455
456 (DEFPROP $GL
457 (LAMBDA (X)
458 (COND ((EQ (CAR X) (Q PNAME))
459 (Q ((NOT (FOUNDS))))))
460 (ATOM (CADR X))
461 ($GL (CDR X))
462 (Q (SETQ SIND (CAR X)
463 (LIST (CADR X)
464 (LIST (CADR X))))))
465 EXPR)
466
467 (DEFPROP $YANK
468 (LAMBDA (O J)
469 ($CURLOC (SETQ
470 SCURSOR
471 (COND
472 ((ATOM (SETQ SOB O))
473 (COND
474 ((SETQ SIND I)
475 (LAMBDA (X)
476 (COND (X (LIST X (LIST X)))
477 (Q ((NOT (FOUNDS)))))))
478 (GET O J)))
479 (T (SETQ SIND NIL) ($GL (CDR O))))))

```



```

480 ((LAMBDA (X)
481 (COND ((ATOM X)
482 (Q ((NOT FOUND)))
483 (T (LIST X (LIST X))))))
484 (EVAL Q))))))
485
486 (DEFPROP $FILE
487 (LAMBDA (Q I)
488 (LAMBDA (X Y)
489 (COND ((ATOM X)
490 (PUTPROP X Y (COND (I) ($IND))))
491 (PROG2 (RPLACA X (CAR Y))
492 Y
493 (RPLACD X (CDR Y))))))
494 (COND (Q) ($OB))
495 (COND ($CURSOR (CAAR (LAST $CURSOR))))))
496
497 (DEFPROP $REMBREAK
498 (LAMBDA (N)
499 (LAMBDA (L)
500 (COND (L (RPLACA (CAR (CADDR L))
501 (CDUAAAR (CADDR L)))
502 (STORE ($LOCAR N) NIL)
503 (STORE ($CONDAR N) NIL)
504 T)))
505 ($LOCAR N)))
506
507 (DEFPROP $RB
508 (LAMBDA (X)
509 (COND
510 ((NULL X)
511 (PROG (N A)
512 (SETQ N 7)
513 LOOP (COND ($REMBREAK N)
514 (SETQ A (CONS N A)))
515 (COND ((MINUSP (SETQ N (SUB1 N)))
516 (RETURN A))
517 (GO LOOP))))))
518 ((MAPC (FUNCTION $REMBREAK) X)))
519
520 (DEFPROP $SETBREAK
521 (LAMBDA (Q I C N CND)
522 (PROG (PROGL)
523 (SETQ PROGL (COND (C (CAR C))))
524 (COND ((ATOM PROGL) (RETURN NIL)))
525 ($REMBREAK N)
526 (RPLACA (RPLACA PROGL
527 (CONS N
528 (CAR PROGL)))
529 (CONS (Q S) (CAR PROGL)))
530 (STORE ($LOCAR N) (LIST I O C))
531 (STORE ($CONDAR N)
532 (COND ((NULL CND) T) (CND)))
533 (RETURN ($CURLOC $CURSOR))))))
534
535 (DEFPROP $BREAK
536 (LAMBDA ($E $AL)
537 (LAMBDA ($C)
538 (LAMBDA ($L $CN)
539 (COND ((EVAL $CN $AL)
540 ($LISP (CAR $E))
541 (CDR $E))))
542 ($LOCAR $C)
543 ($CONDAR $C)))
544 (CAR $E)))
545
546 (DEFPROP $
547 (LAMBDA (X $A)
548 (EVAL ($BREAK X $A) $A))
549
550 (DEFPROP $I
551 (LAMBDA (X) ($INSERT (CAR X)
552 $CURSOR))
553
554 (DEFPROP $S
555 (LAMBDA (I) ($SEARCH1) EXPR)
556
557 (DEFPROP $Y
558 (LAMBDA (L)
559 (COND ((NULL L) NIL)
560 ((NULL (CDR L))
561 ($YANK (CAR L) NIL))
562 ((SYANK (CAR L)
563 (CADR L))))))
564
565 (DEFPROP $F
566 (LAMBDA (L)
567 (COND ((NULL L)
568 ($FILE NIL NIL))
569 ((NULL (CDR L))
570 ($FILE (CAR L) NIL))
571 ((SYANK (CAR L)
572 (CADR L))))))
573
574 (DEFPROP $C
575 (LAMBDA (L)
576 (LAMBDA (X)
577 (SETQ $CURSOR (LIST X (LIST X)))
578 (APPEND ($CURLOC (LAST $CURSOR)
579 NIL)))
580
581 (DEFPROP $B
582 (LAMBDA (L)
583 ($SETBREAK $OB
584 $IND
585 $CURSOR
586 (COND (L (CAR L)) (Q))
587 (COND ((AND L (CDR L))

```

```

540
541 (DEFPROP $BREAK
542 (LAMBDA ($E $AL)
543 (LAMBDA ($C)
544 (LAMBDA ($L $CN)
545 (COND ((EVAL $CN $AL)
546 ($LISP (CAR $E))
547 (CDR $E))))
548 ($LOCAR $C)
549 ($CONDAR $C)))
550 (CAR $E)))
551
552 (DEFPROP $
553 (LAMBDA (X $A)
554 (EVAL ($BREAK X $A) $A))
555
556 (DEFPROP $I
557 (LAMBDA (X) ($INSERT (CAR X)
558 $CURSOR))
559
560 (DEFPROP $S
561 (LAMBDA (I) ($SEARCH1) EXPR)
562
563 (DEFPROP $Y
564 (LAMBDA (L)
565 (COND ((NULL L) NIL)
566 ((NULL (CDR L))
567 ($YANK (CAR L) NIL))
568 ((SYANK (CAR L)
569 (CADR L))))))
570
571 (DEFPROP $F
572 (LAMBDA (L)
573 (COND ((NULL L)
574 ($FILE NIL NIL))
575 ((NULL (CDR L))
576 ($FILE (CAR L) NIL))
577 ((SYANK (CAR L)
578 (CADR L))))))
579
580 (DEFPROP $C
581 (LAMBDA (L)
582 (LAMBDA (X)
583 (SETQ $CURSOR (LIST X (LIST X)))
584 (APPEND ($CURLOC (LAST $CURSOR)
585 NIL)))
586
587 (DEFPROP $B
588 (LAMBDA (L)
589 ($SETBREAK $OB
590 $IND
591 $CURSOR
592 (COND (L (CAR L)) (Q))
593 (COND ((AND L (CDR L))

```



```
600      (CADR L))
601      (T)))
602      FEXPR)
603
604
605      (DEFPROP SLISP
606      (LAMBDA (NO)
607      (PROG (EX)
608      (PRINT (Q BREAK))
609      (PRINT (SCUREXPR SCURSOR))
610      LOOP
611      (TERPRI)
612      (SETQ EX (ERRSET (READ)))
613      (COND ((NULL EX) (GO LOOP)))
614      (SETQ EX (CAR EX))
615      (SETQ EX (ERRSET (EVAL EX)))
616      (COND ((NULL EX) (GO LOOP)))
617      (PRINC SSP)
618      (PRINT (CAR EX))
619      (GO LOOP)))
620      EXPR)
621
622      (DEFPROP SP (LAMBDA () (RETURN NIL)) EXPR)
```