

)(CCR IN)) (PRDPWR EXPS (FUNCTION SETEXP)))))) (CADDR IN))  
INPLT,T121,1027,LISP,LIBRY,R1

.. LISP LIBRARY NOVEMBER 1963  
.. ALL CARDS WITH .. IN COLUMNS 1-2 ARE COMMENTS AND SHOULD  
.. BE REMOVED BEFORE USING.  
.. PUNCHLAP() EXPR PSEUDO-FUNCTION  
.. CCNTENTS 1 DEFINE,1 LAP

DEFINE ((  
(PUNCHLAP (LAMBDA NIL (PROG (X L N EXP)  
A (SETQ X (READ))  
(PUNCH BLANK)  
(COND ((NULL X) (RETURN (PUNCH NIL))))  
(SETQ L (LENGTH (CADDR, X)))  
(SETQ A (CAR X))  
(SETQ EXP (CADR X))  
(SETQ X (PHASE2 (PASSONE N EXP) N))  
(PUNCH (LIST (CONS (LIST N (QUOTE SUBR) L) (CAR X))  
(CADR X))))  
(GC-A) )))

LAP (((LAP SUBR 2) (SXA X 4) (STO (\$ALIST 3)) P  
(XCA) (LDQ (QUOTE NIL)) (TSX CONS 4)  
(XCA) (CLA (\$ALIST 3)) (TSX CONS 4)  
X (AXT 0 4) (TRA PUNCH) ) ()))

.. PRINTPRCP(X) EXPR PSEUDO-FUNCTION  
.. CCNTENTS 1 DEFINE

DEFINE ((  
(PRINTP1, (LAMBDA, (L), (COND, ((NULL, L), NIL), (T, (PROG2, (PRINT, (CAR, L)), (SEABFT1051  
RCH, (QUOTE, (FLCAT SUBR, FSUBR, PNAME, APVAL, FIX)), ( BFT1052  
FUNCTION (LAMBDA, (J), (EQUAL, (BFT1053  
CAR, J), (CAR, L))))), (FUNCTION BFT1054  
(LAMBDA, (J), (PRINTP1, (CDR, (CDR, L))))), (FUNCTION BFT1055  
(LAMBDA, (J), (PRINTP1, (CDR, L))))))))) (LBFT1056

AMBCA, (J), (PRINTP1, (CDR, L))))))))) BFT1057  
(PRINTPRCP, (LAMBDA, (L), (PROG2, (PRINT, (LIST, (QUOTE, PROPERTIES), (QUOTE, OF  
, C)), (PRINTP1, (CDR, L)))))) BFT1059

.. PUNCHDEF(L) EXPR PSEUDO-FUNCTION  
.. CCNTENTS 1 DEFINE

DEFINE((  
(PUNCHDEF (LAMBDA (L) (MAPCON L (FUNCTION (LAMBDA (X) ((LAMBDA (L1) (CONLISP041  
D ((NULL L1) (LIST (CAR X))) (T (PROG2 (PUNCH (LIST (CAR X) (CAR L1))) NLISP042  
IL)))) (PRCP (CAR X) (QUOTE EXPR) (FUNCTION (LAMBDA NIL NIL))))))))) LISP04

.. TRACESET(L) EXPR PSEUDO-FUNCTION  
.. CCNTENTS 1 DEFINE

DEFINE ((  
(TRACESET (LAMBDA (L) (TRACSQ L (FUNCTION TRACSS)))) ARYT003  
(UNTRACESET (LAMBDA (L) (TRACSQ L (FUNCTION TRACSU)))) ARYT004  
ARYT005  
ARYT006

(TRACSQ (LAMBDA (L FN) (MAPCON L (FUNCTION (LAMBDA (J) (PROG (A) ARYT007  
(SETQ A (PROP (CAR J) (QUOTE EXPR) (FUNCTION (LAMBDA () (PROG2 (PRINT  
(CONS (CAR J) (QUOTE (IS NOT EXPR DEFINED)))) NIL)))))) ARYT009  
(COND ((NULL A) (RETURN NIL))) ARYT010  
(SETQ A (CADDR A))

(COND ((NOT (EQ (CAR A) (QUOTE PROG))) (RETURN (PROG2 (PRINT (CONS (CAR J)  
(QUOTE (IS NOT A PROGRAM)))) NIL))) ARYT013  
(SETQ A (CDR A)) ARYT014

TEST (SETQ A (CDR A)) ARYT015  
(COND ((NULL A) (RETURN (LIST (CAR J)))) ARYT016

```

((EQ (CAAR A) (QUOTE SETQ)) (FN A))
(GC TEST) ))) )))
-----
(TRACE (LAMBDA (L) (RPLACD L (COND (LIST
(QUOTE (PRINT BLANK))
(LIST (QUOTE PRINT) (LIST (QUOTE QUOTE) (LIST (CADAR L) EQSIGN)))
(LIST (QUOTE PRINT) (CADAR L)) (CDR L))))))
-----
(TRACE (LAMBDA (L) (RPLACD L (COND (L))))
-----
))
.. COMPILER EXPR PSEUDO-FUNCTION
.. CONTENTS I DEFINE
DEFINE ((
(LENGTH (LAMBDA (M) (PROG (N) (SETQ N 0)
A (COND ((NULL M) (RETURN N)) ) (SETQ N (ADD1 N)) (SETQ M (CDR M))
(GC A) )))
(REVERSE (LAMBDA (X) (PROG (Y) A (COND ((NULL X) (RETURN Y))) (SETQ Y (CLISPO590
ONS (CAR X) Y)) (SETQ X (CDR X)) (GC A)))) LISP0600
(MEMBER (LAMBDA (U V) (COND ((NULL V) NIL) ((EQUAL (CAR V) U) T)
(T (MEMBER U (CDR V))))))
(COMVAL (LAMBDA (EXP STOMAP NAME) (PROG NIL
(COND ((OR (ATOM EXP) (MEMBER (CAR EXP) (QUOTE (QUOTE SPECIAL))))
(LAC EXP))
((EQ (CAR EXP) (QUOTE SETQ)) (PROG NIL
(COMVAL (CADR EXP) STOMAP NAME)
(ATTACH (LIST (CONS (QUOTE STO) (LOCATE (CADR EXP)))))) ))
((EQ (CAR EXP) (QUOTE COND)) (COMCOND (CDR EXP) T))
((EQ (CAR EXP) (QUOTE PROG)) (COMPROG (CDR EXP) (CADR EXP) NAME))
((EQ (CAR EXP) (QUOTE OR)) (COMBCOL F F (CDR EXP) NIL))
((EQ (CAR EXP) (QUOTE AND)) (COMBCOL T F (CDR EXP) NIL))
(ATOM (CAR EXP)) (CALL (CAR EXP) (COMLIS (CDR EXP))))
(T (PROG NIL (COMPLY (CAR EXP) (CDR EXP))
(COMVAL (CADR EXP) STOMAP NAME)))
(SETQ AC NAME)
(RETURN NAME)
(COMPLY (LAMBDA (FN ARGS) (MAP (PAIR (CADR FN) ARGS)
(FUNCTION (LAMBDA (J) (PROG NIL (COMVAL (CDR J) STOMAP
(GENSYM)) (STORE (CAAR J) T))))))
(COMLIS (LAMBDA (EXP) (PROG (X) (RETURN (MAPLIST EXP (FUNCTION (LAMBDA (LISP0170
J) (COND ((OR (EQ (CAAR J) (QUOTE QUOTE)) (ATOM (CAR J))) (CAR J)) (X (PLISP0180
RG2 (STORE AC T) (COMVAL (CAR J) STOMAP (GENSYM)))) (T (PROG2 (SETQ X TLISP0190
) (COMVAL (CAR J) STOMAP (GENSYM)))))))))) LISP0200
(LAC (LAMBDA (X) (COND ((EQUAL AC X) NIL) (T (ATTACH (LIST (CONS (QUOTE LISP0880
CLA) (LOCATE X)))))) LISP0890
(STORE (LAMBDA (X Y) (PROG NIL (COND ((OR (NULL X) (EQ (CAR X) (QUOTE QULISP0900
CTE))) (RETURN NIL))) (SETQ STOMAP (CONS (CONS X (LIST (LIST (ADD1 (CAADLISP0910
AR STOMAP)) (QUOTE *N)) 1)) STOMAP)) (COND (Y (ATTACH (LIST (CONS (QUOTE LISP0920
STO) (LOCATE X)))))) (SETQ LENGTH (MAX LENGTH (CAADAR STOMAP)))))) LISP0930
(PHASE2 (LAMBDA (EXP NAME) (PRG (AC LISTING STOMAP LENGTH) (COND ((LAMLISP0980
BCA (J) (AND (EQ (CAADR EXP) (CADAR J)) (EQ (CAAR J) (QUOTE NULL)) (EQUALISP0990
L (CADR J) (QUOTE (QUOTE NIL)))) (CADAR (CDR EXP))) (PROG2 (ATTACH (QULISP1000
CTE) (TZE 1 4)))) (SETQ EXP (LIST (CAR EXP) (CADR EXP) (CONS (QUOTE CONDLISP1010
) (CADR (CDR EXP)))))) (ATTACH (LIST (LIST (QUOTE INX) (LIST (QUOTE LISP1020
E) NAME) 1 (QUOTE *MN)) (APPEND (QUOTE (TSX *MOVE 1)) ((LAMBDA (J) (LISTLISP1030
(COND ((LESSP J 3) C) (T (DIFFERENCE (TIMES J 2) 4)))) (LENGTH (CADR ELISP1040
XP)))))) (SETQ LENGTH 0) (SETQ STOMAP (QUOTE ((NIL (0 *N) 1)))) (MAP (CALISP1050
DR EXP) (FUNCTION (LAMBDA (J) (STORE (CAR J) F)))) (SETQ AC NIL) (COMVALLISP1060
(CADR EXP) STOMAP NIL) (COND ((NOT (MEMBER (CAADDR EXP) (QUOTE (COND PLISP1070
RCG)))) (ATTACH (QUOTE ((TXI *RETURN 1 *MN)))))) (SETQ EXP (REVERSE LISLISP1080
ING)) (RETURN (LIST EXP (LIST (CONS (QUOTE *MN) (PLUS LENGTH 2)) (CONS (LISP1090

```

ARYT0190  
 ARYT0200  
 ARYT0210  
 ARYT0220  
 ARYT0240  
 ARYT0250  
 ARYT0260  
 ARYT0270

```

QUOTE *N) (DIFFERENCE -2 LENGTH))))))))) LISP1100
(CCMPCRG (LAMBDA (EXP PROGLIS RETN) (PROG (GCLIST HOLD NAME SETS S) (SET
C HOLD EXP) A (COND ((NULL HOLD) (GO B))) ((ATOM (CAR HOLD)) (SETQ GOLIST LISP0430
(CONS (CONS (CAR HOLD) (GENSYM)) GCLIST))) ((NOT SETS) (COND ((EQ (CAAR LISP0440
HOLD) (QUOTE SPECBIND)) (SETQ S (CADADR HOLD))) (T (SETQ SETS T)))) (SLISP0450
ETC HOLD (CDR HOLD)) (GO A) B (SETQ HOLD PROGLIS) C (CLISP0460
ND ((NULL HOLD) (GO G))) (STORE (CAR HOLD) NIL) (COND ((NOT (EQ (CAR HOLLISP0470
D) S)) (ATTACH (LIST (CONS (QUOTE STZ) (LOCATE (CAR HOLD)))))) (SETQ HCLISP0480
LD (CDR HOLD)) (GO C) G (SETQ HOLD EXP) D (SETQ AC NIL) (SETQ NAME (GENSL9SP0490
YM)) (COND ((NULL HOLD) (GO E)) ((ATOM (CAR HOLD))
(ATTACH (LIST (CDR (SASSOC (CAR HOLD) GOLIST NIL)))))) ((EQ (CAAR HOLD) LISP0510
(QUOTE GO)) (ATTACH (LIST (LIST (QUOTE TRA) (CDR (SASSOC (CADADR HOLD) GCLISP0520
LIST (FUNCTION (LAMBDA NIL (ERROR (QUOTE GO)))))))))) ((EQ (CAAR HOLD) (LISP0530
QUOTE COND)) (COMCOND (CDR HOLD) F)) (T (COMVAL (CAR HOLD) STOMAP NAME) LISP0540
)) (SETQ HOLD (CDR HOLD)) (GO D) E (COND (RETN (ATTACH (LIST RETN)))))) L9SP0550
(CMPACT (LAMBDA (EXP NAME) (COND
((EQ (CAR EXP) (QUOTE NULL)) (PROG2 (SETQ SWITCH (NOT SWITCH))
(CMPACT (CADR EXP) NAME)))
((EQUAL EXP (QUOTE (QUOTE *T* ))) (COND (SWITCH (ATTACH (LIST
(LIST (QUOTE TRA) NAME)))) (T (SETQ FLAG F))))
((EQ (CAR EXP) (QUOTE OR)) (COMBOCL F T (CDR EXP) SWITCH))
((EQ (CAR EXP) (QUOTE AND)) (COMBOCL T T (CDR EXP) SWITCH))
(T (PROG2
(CCOND ((EQ (CAR EXP) (QUOTE EQ))
(CEQ EXP STOMAP))
(T (COMVAL EXP STOMAP (GENSYM))))
(ATTACH (LIST (LIST (COND (SWITCH (QUOTE TNZ)) (T(QUOTE TZE)))
NAME))) ) ) ) )
(COMBOCL (LAMBDA (FN MODE EXP A) (PROG (GEN SWITCH)
(SETQ GEN (GENSYM))
A (SETQ SWITCH NIL)
(CCOND ((NULL EXP) (GO C))
((AND MODE (NULL (CDR EXP)) (EQ A FN) ) (GO B)))
(CMPACT (COND (FN (CAR EXP)) (T (LIST (QUOTE NULL) (CAR EXP))))
(CCOND ((AND MODE (NOT A) ) (COND (FN NAME) (T GEN)))
(T (COND ((NOT MODE) GEN) (FN GEN) (T NAME))) ) )
(SETQ AC (COND ((EQ (CAAR LISTING) (QUOTE TNZ)) (QUOTE (QUOTE NIL))
) (T (QUOTE (QUOTE *T* )))))
(SETQ EXP (CDR EXP))
(GO A)
B (CMPACT (COND (FN (LIST (QUOTE NULL) (CAR EXP)))
(T (CAR EXP)) ) NAME)
C (COND ((NOT MODE) (ATTACH (LIST (QUOTE (TRA (* 2))) (LIST (QUOTE
CLA) (LIST (QUOTE QUOTE) FN) ))))
(ATTACH (LIST GEN))
(CCOND ((NOT MODE) (ATTACH (LIST (LIST (QUOTE CLA) (LIST
(QUOTE QUOTE) (NCT FN))))))
)))
(CCMCOND (LAMBDA (EXP MODE) (PROG (FLAG SWITCH GEN)
(SETQ FLAG T)
A (COND ((NULL EXP) (GO B)))
(SETQ GEN (GENSYM))
(SETQ SWITCH NIL)
(CCOND((AND (NCT MODE) (EQ (CAADADR EXP) (QUOTE GO))) (GO C)))
(CMPACT (CAAR EXP) GEN)
(SETQ AC (COND (SWITCH (QUOTE (QUOTE NIL))) (T NIL)))
(COMVAL (CADADR EXP) STOMAP NAME)
(CCOND ((DR (AND NAME (NULL (CDR EXP)))
(MEMBER (CAADADR EXP) (QUOTE (RETURN GO))))
(GO L)))
(ATTACH (LIST (COND (NAME (LIST (QUOTE TRA) NAME))

```

```

L (ATTACH (LIST GEN))
D (SETQ EXP (CDR EXP))
  (SETQ AC (COND (SWITCH (QUOTE NIL)) (T (QUOTE (QUOTE NIL)))))
  (GC A)
B (COND (NAME (ATTACH (LIST NAME))))
  (RETURN NIL)
C (COMPACT (LIST (QUOTE NULL) (CAAR EXP))
  (CDR (SASSCC (CADR (CADAR EXP)) (GOLIST (FUNCTION
    (LAMBDA (V) (ERROR (QUOTE GO)))))))
  (GC D)
  )))
(EQ (LAMBDA (EXP STCMAP) (PROG (A) (SETQ A (COMLIS (CDR EXP))) (COND ((LISP1130
  EQUAL (CAR A) (ATTACH (LIST (CONS (QUOTE SUB) (LOCATE (CADR A)))))) (T (LISP1140
  (PROG2 (LAC (CADR A)) (ATTACH (LIST (CONS (QUOTE SUB) (LOCATE (CAR A)))) (LISP1150
  ))) (SETQ SWITCH (NOT SWITCH)))))) (LISP1160
  (CALL (LAMBDA (FN ARGS) (PROG (HOLD ITEM NUM)
    (COND ((MEMBER FN (QUOTE (SPECBIND SPECRSTR LIST RETURN GO)))
      (GO E))
      ((NULL ARGS) (GO D))
      ((NULL (CDR ARGS)) (GO C)))
    (SETQ FOLD (REVERSE (CDR ARGS)))
    (SETQ NUM (LENGTH ARGS))
    (COND ((GREATERP NUM 20) (ERROR (QUOTE ARGS))))
    (COND ((NULL HOLD) (GO B)))
    (SETQ ITEM (CAR FOLD))
    (COND ((EQUAL ITEM (QUOTE (QUOTE NIL))) (ATTACH (LIST (LIST
      (QUOTE STZ) (LIST (QUOTE $ALIST) NUM))))))
      ((EQUAL ITEM AC) (ATTACH (LIST (LIST (QUOTE STO)
        (LIST (QUOTE $ALIST) NUM))))))
      (T (ATTACH (LIST (LIST (QUOTE STQ) (LIST (QUOTE $ALIST) NUM))
        (CONS (QUOTE LDQ) (LOCATE ITEM) )))))
    (SETQ HOLD (CDR HOLD))
    (SETQ NUM (SUB1 NUM))
    (GO A)
  B (COND ((EQUAL AC (CADR ARGS)) (COND ((EQUAL AC (CAR ARGS))
    (ATTACH (QUOTE (LDQ ($ALIST 2)) (STO ($ALIST 2))))))
    (T (ATTACH (QUOTE (XCA))))))
    (T (ATTACH (LIST (CONS (QUOTE LDQ) (LOCATE (CADR ARGS)))))))
  C (LAC (CAR ARGS))
  D (ATTACH (LIST (LIST (QUOTE STR) (LIST (QUOTE E) FN) 7 (LENGTH
    ARGS))))
    (RETURN NIL)
  E (COND ((EQ FN (QUOTE GO)) (ERROR (QUOTE GO)))
    ((EQ FN (QUOTE RETURN)) (PROG NIL (LAC (CAR ARGS)) (ATTACH
    (LIST (COND (RETN (LIST (QUOTE TRA) RETN)) (T (QUOTE (TXI *RETURN 1
    *MN)))))))))
    ((EQ FN (QUOTE LIST)) (PROG (X)
    (COND ((NULL ARGS) (RETURN (ATTACH (QUOTE ((CLA (QUOTE NIL))))))
      )))
    (COND (AC (LOCATE AC)))
    (ATTACH (QUOTE ((TSX *LIST 4))))
    (ATTACH (LIST (CONS (TIMES (LENGTH ARGS) 1Q6) (LOCATE
    (CAR ARGS))))))
    (SETQ X (CDR ARGS))
  A (COND ((NULL X) (RETURN NIL)))
  (ATTACH (LIST (CONS 0 (LOCATE (CAR X))))))
  (SETQ X (CDR X))
  (GC A)
  ))
  ((MEMBER FN (QUOTE (SPECBIND SPECRSTR))) (PROG NIL
  (ATTACH (LIST (LIST (QUOTE TSX) FN 4)))
  (MAPLIST (CADAR ARGS) (FUNCTION (LAMBDA (J)

```

```

-----
(ATTACH (LIST (LIST (COND ((CDR J) 0) (QUOTE STR)))
(CAR (LOCATE (CAR J))) (LIST (QUOTE SPECIAL) (CAR J))))))
-----
(ATTACH (LAMBDA (A) (COND
  ((AND (EQUAL (CAR A) (QUOTE (TXI *RETURN 1 *MN)))
    (MEMBER (CAAR LISTING) (QUOTE (TXI TRA)))) NIL)
  (T (SETQ LISTING (APPEND A LISTING))))))
-----
(LOCATE (LAMBDA (X) (COND ((OR (EQ (CAR X) (QUOTE QUOTE)) (EQ (CAR X) (QUOTE SPECIAL)) (EQ X (QUOTE $ALIST))) (LIST X)) (T (CDR (SASSOC X STOMAPLISP095
(FUNCTION (LAMBDA NIL (COND ((EQ X AC) (PROG NIL
(STORE AC T) (RETURN (SASSOC X STOMAP (FUNCTION NIL))))))
(T (ERROR (LIST X (QUOTE UNDECLARED)))))))))
(DELETEL (LAMBDA (L M) (MAPCON M (FUNCTION (LAMBDA (J) (COND ((MEMBER (CLISP200
AR J) L) NIL) (T (LIST (CAR J)))))))) LISP1210
(PASSONE (LAMBDA (NAME FN) (PALM (PROGITER NAME FN) NIL)))
(PA1 (LAMBDA (L) (MAPLIST L (FUNCTION (LAMBDA (J) (PAFORM (CAR J) B)))))) LISP1520
) LISP1530
(PA4 (LAMBDA (CCMS SPECS G) (COND ((AND (NULL COMS) (NULL SPECS)) (LIST LISP1630
(QUOTE LAMBDA) (CADR FN) (PAFORM (CADDR FN) (APPEND (CADR FN) B)))) (T (LISP1640
LIST (QUOTE LAMBDA) (CADR FN) (CCNC (LIST (QUOTE PROG) (LIST G)) (PA11 CLISP1650
CCMS (QUOTE COMBIND)) (PA9 SPECS (QUOTE SPECBIND)) (LIST (LIST (QUOTE SETLISP1660
C) G (PAFCRM (CADR FN) (APPEND (CADR FN) B)))) (PA9 SPECS (QUOTE SPECRLISP1670
TR)) (PA14 COMS) (PA12 G)))))) LISP1680
(PA3 (LAMBDA (L) (COND ((NULL (CDR L)) (LIST (LIST (QUOTE (QUOTE *T* LISP1560
) (PAFORM (CAR L) B)))) (T (CONS (LIST (LIST (QUOTE EQ) G (PAFORM (CAAR LISP1570
L) B)) (PAFORM (CADAR L) B)) (PA3 (CDR L)))))) LISP1580
(PA5 (LAMBDA (VARS PRCP) (COND ((NULL VARS) NIL) ((GET (CAR VARS) PROP) LISP1690
(CONS (CAR VARS) (PA5 (CDR VARS) PROP))) (T (PA5 (CDR VARS) PROP)))) LISP1700
(PA6 (LAMBDA (KIND VAR) (LIST (LIST KIND (LIST (QUOTE QUOTE) VAR) (CONS LISP1710
(QUOTE LIST) VAR)))) LISP1720
(PA7 (LAMBDA (L B) (COND ((NULL L) (QUOTE ((RETURN (QUOTE NIL)))))) ((ANCLISP1730
(NULL (CDR L)) (EQ (CAAR L) (QUOTE GO))) L) ((ATOM (CAR L)) (CONS (CAR LISP1740
L) (PA7 (CDR L) B))) (T (CONS (PAFORM (CAR L) B) (PA7 (CDR L) B)))) LISP1750
(PA11 (LAMBDA (VARS FUNC) (COND (VARS (PA6 FUNC VARS)) (T NIL))) LISP1830
(PA14 (LAMBDA (COMS) (COND (COMS (LIST (LIST (QUOTE COMSTR) (LIST (QUOTLISP1840
E QUOTE) (LENGTH CCMS)))))) (T NIL))) LISP1850
(PA12 (LAMBDA (G) (LIST (LIST (QUOTE RETURN) G)))) LISP1860
(COMPPILE (LAMBDA (L) (MAPLIST L (FUNCTION (LAMBDA (J) (COM1 (CAR J) (GETLISP2090
(CAR J) (QUOTE EXPR)) (GET (CAR J) (QUOTE FEXPR)))))) LISP2100
(COM1 (LAMBDA (N A B) (PROG2 (COND (A (COM2 (QUOTE SUBR) (LENGTH (CADR ALISP2110
)) A N)) (B (COM2 (QUOTE FSUBR) (LENGTH (CADR B)) B N)) (T (PRINT (LIST LISP2120
N) (QUOTE UNDEFINED)))))) N))) LISP2130
(COM2 (LAMBDA (TYPE LENGTH EXP NAME) (PROG (LISTING) (SETQ LISTING (PHASLISP2140
E2 (PASSONE NAME EXP) NAME)) (TERPRI) (TERPRI) (PRINT (LIST NAME) LISP2150
E TYPE LENGTH)) (MAP (CAR LISTING) (FUNCTION (LAMBDA (J) (PRINT (CAR J)) LISP2160
))) (TERPRI) (LAP (CONS (LIST NAME TYPE LENGTH) (LISP2170
CAR LISTING)) (CADR LISTING)) (REMPROP NAME (QUOTE EXPR)) (REMPROP NAME LISP2180
(QUOTE FEXPR)) (RETURN NAME)))) LISP2190
(COMMON (LAMBDA (L) (FLAG L (QUOTE COMMON)))) LISP2200
(UNCOMMON (LAMBDA (L) (REMEFLAG L (QUOTE COMMON)))) LISP2210
(SPECIAL (LAMBDA (X) (MAPLIST X (FUNCTION (LAMBDA (J)
(DEFLIST (LIST (LIST (CAR J) (LIST NIL))) (QUOTE SPECIAL)))))) LISP2220
(UNSPECIAL (LAMBDA (L) (MAP L (FUNCTION (LAMBDA (J) (REMPROP (CAR J)
(QUOTE SPECIAL)))))) LISP2230
(PROGITER (LAMBDA (NAME EXP) (COND
  ((AND (EQ (CAADDR EXP) (QUOTE COND)) (P11 (CADR EXP))))
  ((LAMBDA (G1 G2 VS GS) (LIST (QUOTE LAMBDA) VS (CONS (QUOTE PROG) (CONS
GS (CONS G1 (P13 (CADR EXP) NIL (CONS G2 (PAIRMAP VS GS (FUNCTION P12)
(LIST (LIST (QUOTE GC) G1)))))))))) (GENSYM) (GENSYM) (CADR EXP)
(MAPLIST (CADR EXP) (FUNCTION GENSYM))))
(T EXP) )))
-----

```

```
(P1) (LAMBDA (L) (CONS  
(NULL L) F)  
(EQ (CADAR L) NAME) T)  
(T (P1 (CDR L))) )))  
(P12 (LAMBDA (J K) (LIST (QUOTE SETQ) J K)))  
(P13 (LAMBDA (L C S) (COND  
(NULL L) (CONS (CONS (QUOTE COND) C) S))  
(EQ (CADAR L) NAME) ((LAMBDA (G3) (P13 (CDR L) (INCONC C  
(LIST (LIST (CAAR L) (LIST (QUOTE GO) G3)))) (CONS G3  
(PAIRMAP GS (CADAR L)  
(FUNCTION (P12) (CONS (LIST (QUOTE GO) G2) S)))))) (GENSYM))  
(T (P13 (CDR L) (INCONC C (LIST (LIST (CAAR L) (LIST (QUOTE RETURN  
(CADAR L)))))) S)) S))  
(PALAM (LAMBDA (FN B) (COND ((ATOM FN) FN) ((EQ (CAR FN) (QUOTE LAMBDA)) LISP1590  
(PA4 (PA5 (CADR FN) (QUOTE COMMON)) (PA5 (CADR FN) (QUOTE SPECIAL)) (GELISP1600  
NSYM)) ((EQ (CAR FN) (QUOTE LABEL)) (COMP (CADR FN) (CADDR FN))) (T (ERLISP1610  
RDR (CONS FN (QUOTE (NOT FUNCTION))))))  
(PAFORM (LAMBDA (FORM B) (COND ((ATOM FORM) (COND ((OR (NUMBERP FORM) (MLISP1220  
EMBER FORM) (QUOTE (NIL *T* )))) (LIST (QUOTE QUOTE) FORM)) ((EQ FORM (LISP1230  
QUOTE T)) (QUOTE (QUOTE *T* ))) ((EQ FORM (QUOTE F)) (QUOTE (QUOTE NIL LISP1240  
T))  
(GET FORM (QUOTE COMMON)) (LIST (QUOTE EVAL) (LIST (QUOTE QUOTE)  
FORM) (QUOTE $ALIST)))  
(GET FORM (QUOTE SPECIAL)) (LIST (QUOTE SPECIAL) FORM))  
(MEMBER FORM B) FORM)  
(T (PROG NIL (PRINT (CONS FORM (QUOTE ( UNDECLARED ))))  
(RETURN (LIST (QUOTE EVAL) (LIST (QUOTE QUOTE) FORM) (QUOTE $ALIST))))  
) (ATOM (CAR FORM)) (COND ((LISP1270  
CR (GET (CAR FORM) (QUOTE FSUBR)) (GET (CAR FORM) (QUOTE FEXPR))) (COND LISP1280  
(MEMBER (CAR FORM) (QUOTE (AND CR))) (CONS (CAR FORM) (PA1 (CDR FORM))) LISP1290  
) (MEMBER (CAR FORM) (QUOTE (MAX MIN PLUS TIMES LOGOR LOGAND LOGXOR))) LISP1300  
(LIST (CAR FORM) (CONS (QUOTE LIST) (PA1 (CDR FORM))) (QUOTE $ALIST))) (LISP1310  
T (SELECT (CAR FORM) ((QUOTE COND) (CONS (QUOTE COND) (MAPLIST (CDR FORM) LISP1320  
) (FUNCTION (LAMBDA (J) (LIST (PAFORM (CAAR J) B) (PAFORM (CADAR J) B)))) LISP1330  
))) (QUOTE LIST) (CONS (QUOTE LIST) (PA1 (CDR FORM))) (QUOTE QUOTE) LISP1340  
FORM) ((QUOTE PROG) (PA8 (PA5 (CADR FORM) (QUOTE COMMON)) (PA5 (CADR FORMLISP1350  
M) (QUOTE SPECIAL)) (GENSYM))) (QUOTE FUNCTION) (LIST (QUOTE FUNC) (LISP1360  
T (QUOTE QUOTE) (COMP (GENSYM) (CADR FORM))) (QUOTE $ALIST))) (QUOTE SELISP1370  
TC) (COND ((GET (CADR FORM) (QUOTE COMMON)) (LIST (QUOTE SETC) (LIST (QULISP1380  
CTE QUOTE) (CADR FORM)) (PAFORM (CADDR FORM) B) )) (T (LISP1390  
T (QUOTE SETQ) (PAFORM (CADR FORM) B) (PAFORM (CADDR FORM) B)))) (QUOTE LISP1400  
E GO) FORM) ((QUOTE CSETQ) (LIST (QUOTE CSET) (LIST (QUOTE QUOTE) (CADR LISP1410  
FORM)) (PAFORM (CADDR FORM) B))) (QUOTE SELECT) ((LAMBDA (G) (LIST (LISP1420  
T (QUOTE LAMBDA) (LIST G) (CONS (QUOTE COND) (PA3 (CDR FORM))) (PAFORM LISP1430  
(CADR FORM) B))) (GENSYM))) (QUOTE CONC) (PA2 (CDR FORM))) (LIST (CAR LISP1440  
FORM) (LIST (QUOTE QUOTE) (CDR FORM)) (QUOTE $ALIST)))))) ((EQ (CAR FORM) LISP1450  
) (QUOTE NOT)) (LIST (QUOTE NULL) (PAFORM (CADR FORM) B))) ((EQ (CAR FORMLISP1460  
M) (QUOTE SET)) (LIST (QUOTE SETC) (PAFORM (CADR FORM) B) (PAFORM (CADDR LISP1470  
FORM) B) )) (T (CONS (CAR FORM) (PA1 (CDR FORM)))) (OR LISP1480  
(EQ (CAAR FORM) (QUOTE LAMBDA)) (EQ (CAAR FORM) (QUOTE LABEL))) (CONS (LISP1490  
PALAM (CAR FORM) B) (PA1 (CDR FORM))) (T (LIST (QUOTE APPLY) (PAFORM (CLISP1500  
AR FORM) B) (CONS (QUOTE LIST) (PA1 (CDR FORM)) (QUOTE $ALIST)))) LISP1510  
(PAIRMAP (LAMBDA (L M FARG Z) (PROG (A B)  
(COND ((NULL L) (RETURN Z)))  
(SETC A (SETQ B (CONS (FARG (CAR L) (CAR M)) Z)))  
A  
(SETQ L (CDR L))  
(SETQ M (CDR M))  
(COND ((NULL L) (RETURN A)))  
(SETQ B (CDR (RPLACD B (CONS (FARG (CAR L) (CAR M)) Z))))  
(GC A) ))))
```

```

(PA8 (LAMBDA (CCMS SPECS G) (COND ((AND (NULL COMS) (NULL SPECS)) (CONS LISP1760
(QUOTE PRG) (CONS (CADR FORM) (PA7 (CDR FORM) (APPEND (CADR FORM) B)))LISP1770
)) (T (CCNC (LIST (QUOTE PRG) (CONS G (APPEND COMS SPECS))) (PA11 COMS(LISP1780
(QUOTE COMBIND)) (PA9 SPECS (QUOTE SPECBIND)) (LIST (LIST (QUOTE SETQ) G LISP1790
(CONS (QUOTE PRG) (CONS (DELETEL (APPEND COMS SPECS) (CADR FORM)) (PA7 LISP1800
(CDR FORM) (APPEND (CADR FORM) B)))))) (PA9 SPECS (QUOTE SPECRSTR)) (PALISP1810
14 COMS) (PA12 G))))))
LISP1820
(COMP (LAMBDA (N E) (COND
((ATOM E) E)
(T (COM2 (QUOTE SUBR) (LENGTH (CADR E)) E N) )))
(PA9 (LAMBDA (V K) (COND
(V (LIST (LIST K (LIST (QUOTE QUOTE) V))))
(T NIL))))
(PA2 (LAMBDA (L) (COND
(NULL L) (QUOTE (QUOTE NIL)))
(T (LIST (QUOTE APPEND) (PAFORM (CAR L) B) (PA2 (CDR L))))))
)

```

..SIMPLIFY CORRESPONDING TO AI MEMO 27  
..CONTENTS ONE SETSET

```

SETSET SIMPLIFY DEFINITIONS FOR EXPORT LISP 1.5A
OPDEFINE (((TLQ 4008)))
LAP (( (GREATER SUBR 2) (TLQ (* 3)) (CLA (QUOTE NIL)) (TRA 1 4)
(CLA (QUOTE *TRUE*)) (TRA 1 4)) NIL)
DEFINE ((
(INFIX (LAMBDA (L S) (COND ((NULL L) NIL) ((NULL (CDR L)) (LIST (MAKALG LISP0970
(CAR L)))) (T (CONS (MAKALG (CAR L)) (CONS S (INFIX (CDR L) S)))))) LISP0980
(MAKALG (LAMBDA (E) (COND ((OR (NUMBERP E) (ATOM E)) E) (T (SELECT (CAR LISP1020
E) ((QUOTE PLUS) (INFIX (CDR E) PLUSS)) ((QUOTE PRDCT) (INFIX (CDR E) SLISP1030
AR)) ((QUOTE MINUS) (LIST DASH (MAKALG (CDR E)))) ((QUOTE RECIP) (LIST 1LISP1040
SLASH (MAKALG (CDR E)))) ((QUOTE POWER) (INFIX (CDR E) STARR)) ((QUOTE LISP1050
SUBT) (INFIX (CDR E) DASH)) (INFIX (CDR E) SLASH)))))) LISP1060
(SIMPLS (LAMBDA (L) (PROG (NUMS NEGIND IN DIVS COLLST QUTSW
CCM PRCS PWRS GCD)
(SETQ NUMS 0)
(SETQ IN L)
TEST1 (COND
((NULL IN) (GO ENDT1))
((NUMBERP (CAR IN)) (SETQ NUMS (COND
(NEGIND (DIFFERENCE NUMS (CAR IN)))
(T (TOSM (CAR IN) NUMS)) )))
((ATOM (CAR IN)) (SUBATM COLLST))
((EQ (CAAR IN) (QUOTE MINUS)) (GO MINTR))
((EQ (CAAR IN) (QUOTE PLUS)) (GO PLSTR))
((EQ (CAAR IN) (QUOTE DIVIDE)) (SUBDIV DIVS (PLOTNUM (CADAR IN))))
((EQ (CAAR IN) (QUOTE RECIP)) (PROG () (RPLACA IN (LIST (QUOTE
DIVIDE) 1 (CDAR IN)) (SUBDIV DIVS (CONS 1 1)) ))
((EQ (CAAR IN) (QUOTE PRDCT)) ((LAMBDA (A) (SUBPRD COLLST)) (PLOTNUM
(CAR IN))))
((EQ (CAAR IN) (QUOTE SUBT)) (GO SUBTR))
(T (SUBATM COLLST)))
(SETQ IN (CDR IN))
(SETQ NEGIND F)
(GO TEST1)
SUBTR (SETQ IN (APPEND (CDR (UNSUBT (CAR IN)) (CDR IN)))
MINTR (SETQ NEGIND T)
(COND ((EQ (CAAR IN) (QUOTE PLUS)) (SETQ IN (COND (CONCMN (CCDAR IN))
(CDR IN))))
(SETQ IN (CONS (CDAR IN) (CDR IN)))
(GO TEST1)
PLSTR (SETQ IN (APPEND (CDAR IN) (CDR IN)))
(GO TEST1)

```

```

ENDT (COND
  ((NULL DIVS) (GO TEST2))
  (MAP DIVS (FUNCTION (LAMBDA (J) ((LAMBDA (D) (MAP D (FUNCTION
    (LAMBDA (K) (SEARCH COM
    (FUNCTION (LAMBDA (I) (EQUAL (CAAR I) (CAAR K))))
    (FUNCTION (LAMBDA (I) (COND
      ((GREATERP (CDAR I) (CDAR K)) NIL)
      (T (RPLACD (CAR I) (CDAR K)))))))
    (FUNCTION (LAMBDA (J) (SETQ COM (CONS (CAR K) COM)))))))))
    (DECCMP (CADDR (CAAR J))) ) )))
  (MAPRECONST COLLST)
  (MAPDIVCONST DIVS)
  (MAP COM (FUNCTION (RECOM))
    (COND ((NOT (ZEROP NUMS)) (SETQ COLLST (CONS NUMS COLLST))))
    (RETURN (SIMDIV (SIMPLS (MAPLIST (CONC DIVS COLLST)
    (FUNCTION (LAMBDA (J) (SIMPRD (CONS (CAR J)
    COM)))))) (CONS (QUOTE PRDCT) COM)))
    TEST2 (COND ((ZEROP NUMS) (GO COLLECT))
    COLNUM (COND
      ((NOT (AND (OR (FIXP NUMS) (ZEROP NUMS)) (MAPAND COLLST (FUNCTION
        (LAMBDA (J) (FIXP (CDR J)))))) (GO ASBLOT))
      (SETQ GCD NUMS)
      (MAP COLLST (FUNCTION (LAMBDA (J) (SETQ GCD (GCD GCD (CDAR J))))))
      (COND
        ((ZEROP GCD) (RETURN 0))
        ((OR (CNEP GCD) (EQUAL GCD -1)) (GO ASBLOT))
        (MAP COLLST (FUNCTION (LAMBDA (J) (RPLACD (CAR J)
        (QUOTIENT (CDAR J) GCD))))))
        (SETQ NUMS (QUOTIENT NUMS GCD))
        (SETQ OUTSW (CONS GCD OUTSW))
        ASBLOT (MAP COLLST (FUNCTION (LAMBDA (J) ((LAMBDA (A) (COND
          ((EQUAL A 0) NIL)
          ((EQ (CAR A) (QUOTE MINUS)) (SETQ COM (INSORD (CDR A) COM)))
          (T (SETQ IN (INSORD A IN)))) (RECONST (CAR J))))))
          (COND ((ZEROP NUMS) NIL)
            ((MINUSP NUMS) (SETQ COM (CONS (MINUS NUMS) COM)))
            (T (SETQ IN (CONS NUMS IN))))
            (SETQ COM (RESUM2 COM))
            (SETQ IN (RESUM2 IN))
            (SETQ IN (RESUM1 IN COM))
            (COND ((NULL OUTSW) NIL)
              ((NULL (CDR OUTSW)) (SETQ OUTSW (CAR OUTSW)))
              (T (SETQ OUTSW (SIMPRD OUTSW))))
            (COND ((NULL OUTSW) (RETURN IN))
              ((AND (NUMBERP IN) (CNEP IN)) (RETURN OUTSW))
              ((AND (NUMBERP IN) (NUMBERP OUTSW)) (RETURN (TOTIM IN OUTSW))))
            (SETQ OUTSW (SETNEG OUTSW))
            (SETQ IN (SETNEG IN))
            (COND ((EQ (CAR OUTSW) (QUOTE PRDCT)) (SETQ OUTSW (CONS (QUOTE PRDCT)
            (INSORD IN (CDR OUTSW))))))
            (T (SETQ OUTSW (CONS (QUOTE PRDCT) (INSORD IN (LIST OUTSW))))))
            (RETURN (COND
              (NEGINC (SIMMIN OUTSW))
              (T OUTSW)))
            COLLECT (COND
              ((NULL COLLST) (RETURN C))
              ((NULL (CDR COLLST)) (RETURN (RECONST (CAR COLLST))))
              (SETQ DIVS (MAPCON COLLST (FUNCTION (LAMBDA (I) (COND
                ((ZEROP (CDAR I)) NIL)
                (T (LIST (DECCMP (CAAR I))))))))))
              (SETQ OUTSW (CAR DIVS))

```



```

TEST3 (SETQ DIVS (CDR DIVS))
(COND ((NULL DIVS) (GO CONT)) )
(SETQ OUTSW (CCMPAR (CAR DIVS) OUTSW))
(COND ((NULL OUTSW) (GO COLNUM)) )
(GO TEST3)
CONT (MAP OUTSW (QUOTE RECCM))
(MAP COLIST (FUNCTION (LAMBDA (J) (RPLACA (CAR J) (SIMPRD
(LIST (CAAR J) (APPEND (QUOTE (RECIP PRDCT)) OUTSW)))))))
(GO COLNUM) )))
(TCSM (LAMBDA (P Q) (PLUS P Q)))
(TCTIM (LAMBDA (P Q) (TIMES P Q)))
(DELETE (LAMBDA (X Y) (COND
((NULL Y) NIL)
((EQ X (CAR Y)) (CDR Y))
(T (CONS (CAR Y) (DELETE X (CDR Y)) ) ) )))
(LARGER (LAMBDA (X Y) (COND
((NULL X) F)
((NULL Y) T)
((NUMBERP X) (AND (NUMBERP Y) (GREATERP X Y)))
((NUMBERP Y) T)
((ATOM X) (AND (ATOM Y) (GREATER X Y)))
((ATOM Y) T)
((EQUAL (CAR X) (CAR Y)) (LARGER (CDR X) (CDR Y)))
(T (LARGER (CAR X) (CAR Y))) )))
(INSCRD (LAMBDA (X L) (COND
((NULL L) (LIST X))
((LARGER (CAR L) X) (CONS X L))
(T (CONS (CAR L) (INSCRD X (CDR L)))) )))
(CRD (LAMBDA (L R) (COND
((NULL L) R)
(T (CRD (CDR L) (INSCRD (CAR L) R)) )))
(ORDER (LAMBDA (L) (CRD L NIL)))
(ABS (LAMBDA (N) (COND
((MINUSP N) (MINUS N))
(T N) )))
(CCMPAR (LAMBDA (A B) (MAPCON A (FUNCTION (LAMBDA (J) (SEARCH B
(FUNCTION (LAMBDA (K) (EQUAL (CAAR K) (CAAR J))))
(FUNCTION (LAMBDA (K) (LIST (CONS (CAAR K) (COND
((LESSP (CDAR K) (CDAR J)) (CDAR K)) (T (CDAR J))))))
(QUOTE (LAMBDA (K) NIL)) ))))))
(CCNCPN (LAMBDA (L) (MAPLIST L (FUNCTION (LAMBDA (J)
(CONS (QUOTE MINUS) (CAR J)))))))
(DECOMP (LAMBDA (E) (COND
((EQ (CAR E) (QUOTE PRDCT)) (MAPCON (CDR E) (FUNCTION (LAMBDA (J)
(DECOMP (CAR J))))))
((EQ (CAR E) (QUOTE POWER)) ((LAMBDA (N) (COND
((AND (NUMBERP N) (FIXP N)) (MAPLIST (DECOMP (CADR E))
(FUNCTION (LAMBDA (J) (RPLACD (CAR J) N))))
((EQ (CAADR E) (QUOTE PRDCT)) (MAPLIST (CDADR E)
(FUNCTION (LAMBDA (J) (CONS (LIST (QUOTE POWER) (CAR J) N) 1))))
(T (LIST (CONS E 1)) ) ) (CADR E))
(T (LIST (CONS E 1)) ) )))
(DIVP (LAMBDA (X Y) (ZEROP (REMAINDER X Y))))

```

```
(DIVST (LAMBDA (Y) (COND
  ((DIVINC -1)
   (T 1) )))
```

```
(EXP (LAMBDA (X Y) (COND
  ((MINUSP Y) (RECIP (EXPT X (DIFFERENCE 0.0 Y))))
  (T (EXPT X (DIFFERENCE Y 0.0))))))
```

```
(GCD (LAMBDA (M N) (COND
  ((ZEROP N) M)
  ((ZEROP M) N)
  ((GREATERP (ABS M) (ABS N)) (GCD N M))
  ((DIVP N M) M)
  (T (GCD (REMAINDER N M) M) )))
```

```
(INSPWR (LAMBDA (X L FN) (COND
  ((NULL L) (FN))
  ((EQUAL (CADDR X) (CADDR L)) (RPLACA (CDAR L) (CONS (QUOTE PRDCT)
    (INSCRD (CADR X) (COND
      ((EQ (CAADR L) (QUOTE PRDCT)) (CDADR L))
      (T (LIST (CADAR L))))))))))
  (T (INSPWR X (CDR L) FN) )))
```

```
(MAPAND (LAMBDA (L FN) (OR (NULL L) (AND (FN (CAR L)) (MAPAND (CDR L)
  FN) )))
```

```
(MAPRECIPCONST (LAMBDA (K) (MAP K (FUNCTION (LAMBDA (L) (RPLACA L
  (LIST (QUOTE PRDCT) (CDAR L) (CADAR L) (CONS (QUOTE RECIP) (CADDR
  (CAAR L) ))))))))
```

```
(MAPEND (LAMBDA (X FN) (MAPCON X (FUNCTION (LAMBDA (J) (APPEND (FN J)
  NIL))))))
```

```
(MAPRECONST (LAMBDA (L) (MAP L (FUNCTION (LAMBDA (J)
  (RPLACA J (RECONST (CAR J)))))))
```

```
(MPYLSS (LAMBDA (S SL) (COND
  ((EQ (CAR S) (QUOTE SUBT)) (MPYLSS (UNSUBT S) SL))
  ((NULL SL) (CDR S))
  (T (MAPCON (CDR S) (FUNCTION (LAMBDA (J) (MAPLIST SL (FUNCTION
  (LAMBDA (K) (LIST (QUOTE PRDCT) (CAR J) (CAR K))))))))))
```

```
(PLOTNUM (LAMBDA (X) (COND
  ((AND (EQ (CAR X) (QUOTE PRDCT)) (NUMBERP (CADR X))) (CONS (COND
  ((NULL (CADDR X)) (CADDR X))
  (T (CONS (QUOTE PRDCT) (CADDR X)) ) (CADR X)))
  (T (CONS X 1) )))
```

```
(PRDATM (LAMBDA (L FN) (COND
  ((NULL L) (FN))
  ((EQUAL (CAAR L) (CAR IN)) (RPLACD (CAR L) (PLUSPR
  (CDR L) (DVTST))))
  ((LARGER (CAR IN) (CAAR L)) (RPLACA (RPLACD L (CONS (CAR L)
  (CDR L))) (CONS (CAR IN) (DVTST))))
  (T (PRDATM (CDR L) FN) )))
```

```
(PRDPWR (LAMBDA (L FN) (COND
  ((NULL L) (FN))
  ((EQUAL (CAAR L) (CADAR IN)) (RPLACD (CAR L) (PLUSPR
  (CDR L) (SP4))))
  ((LARGER (CADAR IN) (CAAR L)) (RPLACA (RPLACD L (CONS (CAR L) (CDR L)
  (CADAR IN) (DVTST))))
```

```
)) (CCNS (CADAR IN) (SP4)))  
(T (PRCPWR (CDR L) FN)) ) ) )
```

```
(RECCM (LAMBDA (J) (RPLACA J (SIMPWR (CAAR J) (CDAR J)))))
```

```
(RECONST (LAMBDA (X) (COND  
  ((ZEROP (CDR X)) 0)  
  ((CNEP (CDR X)) (CAR X))  
  ((NUMBERP (CAR X)) (TOTIM (CAR X) (CDR X)))  
  ((MINUSP (CDR X)) (SIMMIN (RECONST (CONS (CAR X)  
  (MINUS (CDR X))))))  
  ((EQ (CAAR X) (QUOTE PRDCT)) (CONS (QUOTE PRDCT) (CONS  
  (CDR X) (CDAR X))))  
  (T (LIST (QUOTE PRDCT) (CDR X) (CAR X)) ) ) )
```

```
(RESUM (LAMBDA (L P N) (COND  
  ((NULL L) (RESUM1 (RESUM2 P) (RESUM2 N)))  
  ((EQ (CAAR L) (QUOTE MINUS)) (RESUM (CDR L) P (NCONC N (LIST  
  (CDAR L)))))  
  (T (RESUM (CDR L) (NCONC P (LIST (CAR L))) N)) ) ) )
```

```
(RESUM1 (LAMBDA (P N) (COND  
  ((AND (NULL N) (NULL P)) 0)  
  ((NULL N) P)  
  ((NULL P) (SIMMIN N))  
  (T (LIST (QUOTE SUBT) P N)) ) ) )
```

```
(RESUM2 (LAMBDA (X) (COND  
  ((NULL X) NIL)  
  ((NULL (CDR X)) (CAR X))  
  (T (CONS (QUOTE PLUS) X)) ) ) )
```

```
(SETEXP (LAMBDA () (SETQ EXPS (NCONC EXPS (LIST (CCNS (CADAR IN)  
  (SP4)))))) )
```

```
(SETNEG (LAMBDA (L) (COND  
  ((AND (NUMBERP L) (MINUSP L)) (PRG2 (SETQ NEGIND (NOT NEGIND))  
  (MINUS L)))  
  ((EQ (CAR L) (QUOTE MINUS)) (PRG2 (SETQ NEGIND (NOT NEGIND)) (CDR L)  
  )) (T L)) ) ) )
```

```
(SETSUM (LAMBDA () (SETQ SUMS (NCONC SUMS (LIST  
  (CCNS (CADAR IN) (SP4)))))) )
```

```
(SIMDIV (LAMBDA (X Y) (SIMPRD (LIST X (SIMRCP Y)))))
```

```
(SIMMIN (LAMBDA (X) (COND  
  ((NULL X) (ERROR (QUOTE SIMMIN)))  
  ((NUMBERP X) (MINUS X))  
  ((EQ (CAR X) (QUOTE MINUS)) (CDR X))  
  ((EQ (CAR X) (QUOTE SUBT)) (LIST (QUOTE SUBT) (CADDR X) (CADR X)))  
  (T (CONS (QUOTE MINUS) X)) ) ) )
```

```
(SIMPLIFY (LAMBDA (L) (COND  
  ((NULL L) (ERROR (QUOTE SIMPLIFY)))  
  ((OR (ATOM L) (NUMBERP L)) L)  
  ((EQ (CAR L) (QUOTE MINUS)) (SIMMIN (SIMPLIFY (CDR L))))  
  ((EQ (CAR L) (QUOTE PLUS)) (SIMPLS (MAPLIST (CDR L) (FUNCTION (LAMBDA  
  (J) (SIMPLIFY (CAR J)))))) ) ) )  
  ((EQ (CAR L) (QUOTE PRDCT)) (SIMPRD (MAPLIST (CDR L) (FUNCTION (LAMBDA  
  (J) (SIMPLIFY (CAR J)))))) ) ) )
```

```

((EQ (CAR L) (QUOTE DIVIDE)) (SIMDIV (SIMPLIFY (CADR L))
(SIMPLIFY (CADDR L))))
((EQ (CAR L) (QUOTE SUBT)) (SIMSUB (SIMPLIFY (CADR L)) (SIMPLIFY
(CADDR L))))
((EQ (CAR L) (QUOTE POWER)) (SIMPWR (SIMPLIFY (CADR L)) (SIMPLIFY
(CADDR L))))
(T (SIMRCP (SIMPLIFY (CADR L))))))

```

```

(SP3 (LAMBDA (I N) (COND
((NULL D) (COND
((NULL N) NIL)
((GREATERP I 1) (SIMPLS (MAPLIST N (FUNCTION
(LAMBDA (J) (SIMPRD (LIST (CAR J))))))))
(T (RESUM N NIL NIL) ))
(T (NULL N) (COND
((NULL (CADR D)) (CAR D))
(T (CONS (QUOTE PRDCT) D) ))
(T (SIMPLS (MAPLIST N (FUNCTION (LAMBDA (J) (SIMPRD
(APPEND D (LIST (CAR J))))))))))

```

```

(PLUSPR (LAMBDA (X Y) (COND
((AND (NUMBERP X) (NUMBERP Y)) (TOSM X Y))
(T (SIMPLS (LIST X Y)) )))

```

```

(SIMPWR (LAMBDA (X Y) (COND
((CR (NULL X) (NULL Y)) (ERROR NIL ))
((AND (NUMBERP Y) (ZEROP Y)) I)
((AND (NUMBERP X) (ZEROP X)) C)
((AND (NUMBERP Y) (ONEP Y)) X)
((AND (NUMBERP Y) (NUMBERP X)) (COND
((NOT (MINUSP X)) (EXP X Y))
((AND (MINUSP X) (DIVP Y 2)) (EXP (MINUS X) Y))
((AND (MINUSP X) (DIVP (ADD1 Y) 2)) (MINUS (EXP (MINUS X) Y)))
((MINUSP Y) (LIST (QUOTE RECIP) (QUOTE POWER) X (MINUS Y)))
(T (LIST (QUOTE POWER) X Y))))
((AND (NUMBERP Y) (MINUSP Y)) (SIMRCP (SIMPWR X (MINUS Y))))
((EQ (CAR Y) (QUOTE MINUS)) (SIMRCP (SIMPWR X (CADR Y))))
((CR (NUMBERP X) (ATOM X)) (LIST (QUOTE PCWER) X Y))
((EQ (CAR X) (QUOTE RECIP)) (SIMRCP (SIMPWR (CADR X) Y)))
((EQ (CAR X) (QUOTE DIVIDE)) (SIMDIV
(SIMPWR (CADR X)
Y) (SIMPWR (CADDR X) Y)))
((AND (EQ (CAR X) (QUOTE MINUS)) (NUMBERP Y) (FIXP Y)) (COND
((DIVP Y 2) (SIMPWR (CADR X) Y))
((DIVP (ADD1 Y) 2) (SIMMIN (SIMPWR (CADR X) Y)))
(T (LIST (QUOTE POWER) X Y) )))
((EQ (CAR X) (QUOTE POWER)) (SIMPWR (CADR X) (SIMPRD (LIST (CADDR
X) Y))))
((EQ (CAR X) (QUOTE PRDCT)) (SIMPRD (MAPLIST (CADR X) (FUNCTION (LAMBDA
(J) (SIMPWR (CAR J) Y))))))
(T (LIST (QUOTE POWER) X Y) )))

```

```

(SIMRCP (LAMBDA (X) (COND ((NUMBERP X) (QUOTIENT 1.0 X))
((ATOM X) (CONS (QUOTE RECIP) X))
((EQ (CAR X) (QUOTE MINUS)) (SIMMIN (SIMRCP (CADR X))))
((EQ (CAR X) (QUOTE RECIP)) (CADR X))
((EQ (CAR X) (QUOTE DIVIDE)) (COND
((NUMBERP (CADR X)) (RECONST (CONS (CADR X) (QUOTIENT 1.0 (CADR
X))))))
(T ((LAMBDA (A) (LIST (QUOTE DIVIDE) (RECONST (CONS (CADR X)
(QUOTIENT 1.0 (CADR A)))) (CAR A))) (PLOTNUM (CADR X) )) ))
((EQ (CAR X) (QUOTE PRDCT))

```

```
(SIMPRD (MAPLIST (CDR X) (FUNCTION (LAMBDA (J) (SIMRCP (CAR J))))))
```

```
)  
(T (CONS (QUOTE (RECIP) X)) )))
```

```
(SIMSUB (LAMBDA (X Y) (SIMPLS (LIST X (SIMMIN Y)))))
```

```
(SP1 (LAMBDA (N) (COND
```

```
((ZEROP N) NIL)
```

```
(T (PRCG ()
```

```
(SETQ DENIND (ADD1 DENIND))
```

```
(SETQ DENSUM (MPYLSS (CAAR J) DENSUM))
```

```
(SP1 (ADD1 N)) )) ))
```

```
(SP2 (LAMBDA (N) (COND
```

```
((ZEROP N) NIL)
```

```
(T (PRCG ()
```

```
(SETQ NUMIND (ADD1 NUMIND))
```

```
(SETQ NUMSUM (MPYLSS (CAAR J) NUMSUM))
```

```
(SP2 (SUB1 N)) )) ))
```

```
(SP4 (LAMBDA () (COND
```

```
(DIVIND (SIMMIN (CACCAR IN)))
```

```
(T (CACCAR IN)) )))
```

```
(SUBATM (LAMBDA (L) (COND
```

```
((NULL L) (SETQ COLLST (CONS (CONS (CAR IN) (COND (NEGIND -1)
```

```
(T 1))) COLLST)))
```

```
((EQUAL (CAAR L) (CAR IN)) (RPLACD (CAR L) (COND (NEGIND (SUB1 (CDAR L)))
```

```
(T (ADD1 (CDAR L))))))
```

```
(T (SUBATM (CDR L)))))
```

```
(SUBDIV (LAMBDA (L A) (COND
```

```
((NULL L) (SETQ DIVS (CONS (CONS (CONS (QUOTE DIVIDE) (CONS (CAR A)
```

```
(CDDR IN))) (COND
```

```
(NEGIND (MINUS (CDR A)))
```

```
(T (CDR A)))) DIVS)))
```

```
((EQUAL (CADR (CDDR L)) (CACCAR IN)) ((LAMBDA (X) (COND
```

```
((ZEROP (CDR X)) (RPLACD (RPLACA L (CADR L)) (CDDR L)))
```

```
(T (RPLACA L (CONS (LIST (QUOTE DIVIDE) (CAR X) (CACCAR IN))
```

```
(CDR X))))))
```

```
(PLOTNUM (SIMPLS (LIST (RECONST (CONS (CADAAR L) (CDAR L))) (RECONST
```

```
(COND (NEGIND (CONS (CAR A) (MINUS (CDR A)))) (T A))))))
```

```
(T (SUBDIV (CDR L) A)) )))
```

```
(SUBPRD (LAMBDA (L) (COND
```

```
((NULL L) (SETQ CELLST (CONS (COND
```

```
(NEGIND (CONS (CAR A) (MINUS (CDR A))))
```

```
(T A)) CELLST)))
```

```
((EQUAL (CAAR L) (CAR A)) (RPLACD (CAR L) (TOSM (COND
```

```
(NEGIND (MINUS (CDR A)))
```

```
(T (CDR A)) (CDDR L))))
```

```
(T (SUBPRD (CDR L)))))
```

```
(UNSUBT (LAMBDA (A) (CONS (QUOTE PLUS) (NCONC (COND
```

```
((EQ (CAADDR A) (QUOTE PLUS)) (CONC (CDADDR A))
```

```
(T (LIST (CONS (QUOTE MINUS) (CADDR A))))))
```

```
(COND ((EQ (CAADDR A) (QUOTE PLUS)) (CADDR A)
```

```
(T (LIST (CADR A)))))
```

```
(SIMPRD (LAMBDA (L) (PROG (IN_EXPS DENOM RCPPWR NUMS TOP SUMS DIVIND
```

```
SIGN NUMSUM DENSUM NUMIND DENIND)
```

```

(SETQ NUMIND 0)
(SETQ DENIND 0)
(SETQ NUMS '1)
(SETQ IN L)
TEST1 (COND
  ((NULL IN) (GO ENCT1))
  ((NUMBERP (CAR IN)) (SETQ NUMS (COND
    (DIVIND (QUOTE NUMS (CAR IN)))
    (T (TOTM NUMS (CAR IN))))))
  ((ATOM (CAR IN)) (PRDATM EXPS (FUNCTION (LAMBDA () (SETQ EXPS
    (NCONC EXPS (LIST (CONS (CAR IN) (DVIST))))))))
  ((EQ (CAAR IN) (QUOTE MINUS)) (GO MINTR))
  ((EQ (CAR IN) (QUOTE PRDCT)) (GO PRDTR))
  ((EQ (CAAR IN) (QUOTE RECIP)) (GO RCPTR))
  ((EQ (CAR IN) (QUOTE POWER)) (GO PWTR))
  ((EQ (CAAR IN) (QUOTE DIVIDE)) (GO DIVTR))
  (T (PRDATM SUMS (FUNCTION (LAMBDA () (SETQ SUMS (NCONC SUMS
    (LIST (CONS (CAR IN) (DVIST))))))))))
RESET (SETQ IN (CDR IN))
(SETQ DIVIND F)
(GO TEST1)
ENCT1 (MAP EXPS (FUNCTION (LAMBDA (J) ((LAMBDA (X) (COND
  ((NUMBERP X) NIL)
  ((EQ (CAR X) (QUOTE POWER)) (INSPWR X IN (FUNCTION (LAMBDA NIL
    (SETQ IN (CONS X IN))))))
  ((EQ (CAR X) (QUOTE RECIP)) (COND
    ((EQ (CADR X) (QUOTE POWER)) (INSPWR (CDR X)
    RCPPWR (FUNCTION (LAMBDA () (SETQ RCPPWR (CONS (CDR X) RCPPWR))))))
    (T (SETQ DENOM (CONS (CDR X) DENOM))))
  (T (SETQ TOP (CONS X TOP))))
  (SIMPLR (CADR J) (CDR J))))
(MAP SUMS (FUNCTION (LAMBDA (J) (COND
  ((MINUSP (CDR J)) (SP1 (CDR J))
  (T (SP2 (CDR J))))))
(MAP IN (FUNCTION (LAMBDA (J) (SETQ TOP (INSCRD (CAR J) TOP))))
(MAP RCPPWR (FUNCTION (LAMBDA (J) (SETQ DENOM (INSCRD (CAR J) DENOM))))
(COND ((MINUSP NUMS) (PROG2 (SETQ SIGN (NOT SIGN)) (SETQ NUMS
  (MINUS NUMS))))
(COND ((ZEROP NUMS) (RETURN 0))
  ((CNEP NUMS) NIL)
  (T (SETQ TOP (CONS NUMS TOP))) )
(SETQ DENOM (SP3 DENOM DENIND DENSUM))
(SETQ TOP (SP3 TOP NUMIND NUMSUM))
(COND ((AND (NULL TOP) (NULL DENOM)) (SETQ TOP 1))
  ((NULL TOP) (SETQ TOP (CONS (QUOTE RECIP) DENOM))
  ((NULL DENOM) NIL)
  (T (SETQ TOP (LIST (QUOTE DIVIDE) TOP DENOM))))
(COND (SIGN (SETQ TOP (SIMPLR TOP))) )
(COND ((OR (GREATERP DENIND 1) (GREATERP NUMIND 1)) (RETURN (SIMPLR
  (LIST TOP)))) )
(RETURN TOP)
MINTR (SETQ IN (CONS (CDR IN) (CDR IN)))
(SETQ SIGN (NOT SIGN))
(GO TEST1)
PRDTR (SETQ IN (APPEND (CDR IN) (CDR IN)))
(GO TEST1)
DIVTR (SETQ IN (CONS (CONS (QUOTE RECIP) (CADR IN)) (CONS (CADAR
  IN) (CDR IN))))
RCPTR (COND
  ((EQ (CADAR IN) (QUOTE PRDCT)) (SETQ IN (NCONC (MAPLIST (CDR AR
  IN) (QUOTE (LAMBDA (J) (CONS (QUOTE RECIP) (CAR J)))))) (CDR IN)))) )

```

```
(SETQ IN (CONS (CADR IN) (CDR IN)))
(SETQ DIVIND 1)
(GO TEST1)
PWRTR (CCND
  ((EQ (CAADAR IN) (QUOTE PRDCT))) ((LAMBDA (X) (MAP (CDADAR IN)
(FUNCTION (LAMBDA (J) (PRG2 (SETQ IN (CONS (LIST (QUOTE POWER) (CAR J)
X) (CDR IN))) (PRDPWR EXPS (FUNCTION SETEXP)))))) (CADDAR IN)))
  ((AND (NUMBERP (CADDAR IN)) (FIXP (CADDAR IN)) (EQ (CAADAR IN)
(QUOTE PLUS))) (PRDPWR SUMS(FUNCTION SETSUM)))
  (T (PRDPWR EXPS (FUNCTION SETEXP))) )
(GO RESET)
))
```

```
(DIFF (LAMBDA (E X) (CCND
  ((OR (NULL E) (NCT (ATOM X))) (ERROR (QUOTE DIFF))))
  ((ATOM E) (CCND
    ((EQ X E) 1)
    (T C) ))
  (T (SELECT (CAR E)
    ((QUOTE MINUS) (SIMMIN (DIFF (CDR E) X)))
    ((QUOTE RECIP) (SIMMIN (SIMDIV (DIFF (CDR E) X) (SIMPWR (CDR E)
2))))
    ((QUOTE PLUS) (SIMPLS (MAPLIST (CDR E) (FUNCTION (LAMBDA (J)
(DIFF (CAR J) X))))))
    ((QUOTE PRDCT) (SIMPLS (MAPLIST (CDR E) (FUNCTION (LAMBDA (J)
(SIMPRD (CONS (DIFF (CAR J) X) (DELETE (CAR J) (CDR E)))))))))
    ((QUOTE DIVIDE) (SIMDIV (SIMSUB (SIMPRD (LIST (CADDR E) (DIFF
(CADR E) X))) (SIMPRD (LIST (CADR E) (DIFF (CADDR E) X)))) (SIMPWR
(CADDR E) 2)))
    ((QUOTE POWER) (SIMPRD (LIST (CADDR E) (SIMPWR (CADR E) (SIMSUB
(CADDR E) 1)) (DIFF (CADR E) X))))
    ((QUOTE SBT) (SIMSUB (DIFF (CADR E) X) (DIFF (CADDR E) X)))
    (ERROR (QUOTE DIFF) )) ))
  ))
```

```
STOP))))))))))
..METEOR CORRESPONDING TO AI MEMO 51
..CONTENTS ONE SETSET
  SETSET M2447 BOBROW METEOR CCT 1 1963
CPDEFINE(( (NCP 76108) ))
LAP ((
(CCMPRINT SUBR 1) (LDQ A) (TZE (* 2)) (LDQ B)
(STQ 74204Q)(TRA 1 4) A (NCP) B (TSX PRINT,4) ) NIL)
CCMPRINT(NIL)
SPECIAL((MPAIRS PRS WORKSPACE DISPCH SHELE TRACK))
(LAMBDA (X) (CCPILE (DEFINE X))) ((
(METEOR
```

```
(LAMBDA (RULES WORKSPACE) (METRIX RULES WORKSPACE
NIL NIL NIL)))
(METRIX
  (LAMBDA (RULES WORKSPACE SHELE DISPCH TRACK)
(METRIX2 RULES WORKSPACE)))
(METRIX2
```

```
(LAMBDA (RULES WORKSPACE) (PROG (PC GT A)
(SETQ RULES (MAPLIST
RULES (FUNCTION (LAMBDA (X) (PROG (A B)(SETQ B(CAR X)) (SETQ A
(LIST (CAR B))) (SETQ B (CDR B)) (COND ((NCT (ATOM (CAR B)))
(GO NTATM)) ((NCT (EQ (CAR B) (QUOTE *))) (RETURN (CAR X))))
(SETQ A (ADDLAST A (CAR B))) (SETQ B (CDR B))
NTATM
(RETURN (CONC A (CONS (NAPER (CAR B)) (CDR B)))))))))
(SETQ PC RULES) START (COND ((NULL PC ) (RETURN(LIST(QUOTE(NO END))
```

```

WORKSPACE RULES)))
(COND ( (NULL TRACK) (GO TRACK)))
(PRINT (QUOTE RULE)) (PRINT (CAR PC)) (PRINT (QUOTE WORKSPACE))
(PRINT WORKSPACE) (TERPRI) TRACK
(SETQ GT (DISPATCH (COMITRULE (CDR PC))))
(COND ((EQ GT (QUOTE *)) (GO NEXT)) ((EQ GT (QUOTE END)) (RETURN
WORKSPACE))) ((EQUAL GT (CAAR PC)) (GO START)))
(SETQ A (TRANSFER GT RULES)) (COND
((EQ (CAR A) (QUOTE NONAME)) (RETURN (LIST A WORKSPACE
(LIST (QUOTE (FROM RULE)) (CAR PC)) (LIST (QUOTE (SHELF IS)) SHELF)))))
(SETQ PC A)
(GO START) NEXT (SETQ PC (CDR PC)) (GO START)))
(TRANSFER
(LAMBDA (GT RL) (PROG ()
START (COND
((NULL RL) (RETURN (LIST (QUOTE NONAME) GT)))
((EQ GT (CAAR RL)) (RETURN RL))) (SETQ RL (CDR RL)) (GO START))))
(DISPATCH
(LAMBDA (GT) (PROG (A)
(COND ((EQ GT (QUOTE *)) (RETURN GT)))
(SETQ A (GTPAIR GT DISPATCH)
(COND ((NULL A) (RETURN GT)) (RETURN (CAR A))))))
(GTPAIR
(LAMBDA (NAME X) (PROG (A)
START (COND ((NULL X) (RETURN NIL))
((EQUAL (CAR X) NAME) (RETURN (CDR X))))
(SETQ X (CDR X)) (GO START))))
(COMITRULE
(LAMBDA (RULE) (PROG (LEFT A B C D E)
(SETQ A (CAR RULE)) (SETQ E (QUOTE *)))
(COND ((NOT (ATOM A)) (GO START)
((EQ A (QUOTE *)) (GC STAR))) (DEFLIST (CDR RULE) A)
(RETURN (QUOTE *)) STAR
(SETQ RULE (CDR RULE)) (SETQ E (FSTATM RULE))
START (SETQ LEFT (COMITMATCH2(CAR RULE) WORKSPACE))
(COND ((NULL LEFT) (RETURN E)))
LOOP (SETQ RULE (CDR RULE)) (SETQ A (CAR RULE))
(COND ((NULL RULE) (RETURN (QUOTE *)))
((EQ A (QUOTE $)) (SETQ A (CAR WORKSPACE)))
((EQUAL A C) (GO CN))
((ATOM A) (GO SW)) ((EQ (CAR A) (QUOTE /)) (GO SV)) (T (GO ON)))
SW (COND ((EQ E (QUOTE *)) (RETURN A))
(RETURN (QUOTE *)) CN
(SETQ WORKSPACE (COMITR LEFT A)) (GO LOOP)
SV (SHELVE LEFT A) (GO LOOP))))
(FSTATM
(LAMBDA (RULE) (PROG (A) START (SETQ A (CAR RULE))
(COND ((NULL RULE) (RETURN (QUOTE *)))
((EQUAL A O) (GC ON)) ((ATOM A) (RETURN A)) ON
(SETQ RULE (CDR RULE)) (GC START))))
(SHELVE
(LAMBDA (PAIRS INST) (PROG (A B C D) START
(SETQ INST (CDR INST)) (COND ((NULL INST) (RETURN SHELF)))
(SETQ A (CAR INST)) (SETQ B (CAR A)) (SETQ C (CADR A))
(SETQ D (CCDR A)) (COND
((EQ B (QUOTE *P)) (GO PR))
((EQ B (QUOTE *D)) (RETURN (SETDIS C (CAR D))))
((NOT (EQ C (QUOTE *))) (GO GETD)))
(SETQ C (INDIRECT (CAR D) PAIRS)) (SETQ D (CDR D))
GETD (SETQ D (COMITRIN PAIRS D))
(SETQ A (GTSHLF C))

```



```

(COND ((EQ B (QUOTE *S)) (GO ST1)) ((EQ B (QUOTE *Q))(GO QU1))
((EQ B (QUOTE *X)) (GO EX)))
(PRINT (LIST (QUOTE (SHELVING ERROR IN)) (CAR INST))) (GO START)
PR (COND ((EQ C (QUOTE /)) (RETURN (PRINT SHELF)))) PR1
(PRINT (LIST (QUOTE SHELF) C (QUOTE CONTAINS) (CAR (GTSHLF C))))
(COND ((NULL D) (GO START))) (SETQ C (CAR D))
(SETQ D (CDR D)) (GO PR1)
EX (SETQ B (CAR A)) (RPLACA A WORKSPACE) (SETQ WORKSPACE B)(GO START)
QU1 (RPLACA A (NCCNC (CAR A) D)) (GO START)
ST1 (RPLACA A (APPEND D (CAR A))) (GO START)))
(SETDIS
(LAMBDA (X Y) (PROG (A)
(SETQ A (GTPAIR X DISPCH))
(COND ((NULL A) (SETQ DISPCH (CONS X (CONS Y DISPCH))))
(T (RPLACA A Y)) (RETURN DISPCH))))
(GETDCT
(LAMBDA (X Y) (PROG (A) (COND ((NOT (ATOM X)) (RETURN (LIST X))))
(SETQ A (GET X Y))
(COND ((NULL A) (RETURN X)) (RETURN A)))
(INDIRECT
(LAMBDA (X PAIRS) (GTNAME X PAIRS))
))
(LAMBDA (X) (CCMPLE (DEFINE X))) ((
(COMTR
(LAMBDA (LEFT ORDER)(PROG (A B C)
(SETQ A (GTNAME O LEFT)) (COND ((EQUAL A O)(SETQ A NIL))
((NULL A) (GO ON)) ((ATOM A) (SETQ A (LIST A))))
ON (SETQ B (GTNAME (QUOTE WSEND) LEFT))
(COND((EQUAL ORDER O) (SETQ C NIL))
(T (SETQ C (COMTRIN LEFT ORDER)))) (RETURN (APPEND A (APPEND C B))))))
(COMTRIN
(LAMBDA (LEFT ORDER) (PROG (A B)
START (COND ((NULL ORDER)(RETURN A)))
(SETQ B (GTNAME (CAR ORDER) LEFT)) (COND ((NULL B) (GO ON))
((ATOM B) (SETQ B (LIST B)))) (SETQ A (NCCNC A B)) ON
(SETQ ORDER (CDR ORDER)) (GO START) ))
(GTNAME
(LAMBDA (NAME PRS) (PROG (A B C)(SETQ C (CAR NAME))
(COND ((ATOM NAME)(GO START))
((EQ C (QUOTE FN)) (RETURN (COPYTP (APPLY (CADR NAME)
(COMTRIN PRS (CDR NAME)) NIL))))
((EQ C (QUOTE *K)) (RETURN (LIST (COMTRIN PRS (CDR NAME))))))
((EQ C (QUOTE *C)) (RETURN (COMPRESS (COMTRIN PRS (CDR NAME))))))
((EQ C (QUOTE *) (RETURN (COPYTP (EVAL (CADR NAME) NIL))))))
((EQ C (QUOTE *w)) (RETURN (WRITES (COMTRIN PRS (CDR NAME))))))
((EQ C (QUOTE *E))(RETURN(EXPAND(GTNAME (CADR NAME) PRS))))
((EQ C (QUOTE */)) (RETURN (LIST (SBMERGE (CDR NAME))))))
((EQ C (QUOTE *N))(RETURN (NEXT (CDR NAME))))
((EQ C (QUOTE *R)) (RETURN (MTREAD)))
((EQ C (QUOTE *A)) (RETURN (ALL (CDR NAME)))).((EQ C (QUOTE QUOTE))
(RETURN (CADR NAME))))
START (COND ((NULL PRS) (RETURN NAME))) (SETQ A (CAR PRS))
(COND ((EQUAL NAME (CAR A)) (RETURN (COPYTP (CDR A))))
(SETQ PRS (CDR PRS)) (GO START))))
(COPYTP
(LAMBDA (X) (COND ((ATOM X) X)(T (APPEND X NIL))))))
(EXPAND
(LAMBDA (X) (COND
((ATOM X) (MAPCON (GET (CDR X) (QUOTE PNAME))
(FUNCTION (LAMBDA (Y) (UNPACK (CAR Y))))))
(T (CAR X))))))

```

---(COMPRESS

(LAMBDA (X) (PROG ()))

---(CLEARBUFF)---(MAP X (FUNCTION (LAMBDA (X) (PACK (CAR X))))))  
(RETURN (INTERN (MKNAM))))))

---(TREAD

(LAMBDA (A) (PROG (A B C) (SETQ A (STARTREAD)) (GO A)

---START---(SETQ A (ADVANCE))---A---(COND ((EQ A (QUOTE \$EOF\$))(RETURN A))  
(EQ A (QUOTE \$EOR\$)) (RETURN B))

---(EQ A (QUOTE \$\$\$ \$))---(SETQ C (NCONC C (LIST A))))

---(T (GO B)) (GO START) B (SETQ B (NCONC B (NCONC C (LIST A))))

---(SETQ C NIL) (GO START))))

---(ALL

---(LAMBDA (X)---(PROG (A B)

---(COND ((EQ (CAR X) (QUOTE \*))) (SETQ X (INDIRECT (CADR X) PRS)))

---(T---(SETQ X (CAR X))))

---(SETQ A (GTSHLF X))

---(SETQ B (CAR A)) (RPLACA A NIL) (RETURN B))))

---(NEXT

---(LAMBDA (X)---(PROG (A B C)

---(COND ((EQ (CAR X) (QUOTE \*))) (SETQ X (INDIRECT (CADR X) PRS)))

---(T---(SETQ X (CAR X))))

---(SETQ A (GTSHLF X))

---(SETQ C (CAR A))

---(COND ((NULL C) (RETURN NIL)))

---(SETQ B (CAR C)) (RPLACA A (CDR C)) (RETURN (LIST B))))

---(GTSHLF

---(LAMBDA (X)---(PROG (A)

---(SETQ A (GTPAIR X SHELF)) (COND ((NULL A) (GO A)))

---(RETURN A) A (SETQ A (CONS NIL SHELF))

---(SETQ SHELF (CONS X A)) (RETURN A))))

---(SBMERGE

(LAMBDA (X) (PROG (A B C D)

---(SETQ A (CAR X)) (SETQ B (GNAME (CADR X) PRS))

---(COND ((ATOM B) (GO B))) (SETQ B (CAR B))

---B (SETQ C (CDR (GNAME (CADDR X) PRS)))

---(COND ((NOT (EQ (CAR C) (QUOTE /))) (GO ERROR))) (SETQ C (CDR C))

---(COND ((ATOM B) (SETQ B (LIST B (QUOTE /))))

---(SETQ D (LIST (CAR B) (QUOTE /)))

---(SETQ B (CDR B))

---(COND ((EQ A (QUOTE AND)) (GO AND)) ((EQ A (QUOTE OR)) (GO OR))

---((EQ A (QUOTE SUBST)) (GO SUBST)))

---ERROR (PRINT (LIST (QUOTE (SUBSCRIPT ERROR)) X))

---(RETURN (CADR X))

---AND (SETQ A NIL) A (COND ((NULL B) (RETURN (NCONC D A)))

---((MEMBER (CAR B) C) (SETQ A (ADDLAST A (CAR B))))

---(SETQ B (CDR B)) (GO A)

---OR (SETQ A NIL) B (COND ((NULL B) (RETURN (NCONC D (APPEND A C)))

---((NOT (MEMBER (CAR B) C)) (SETQ A (ADDLAST A (CAR B))))

---(SETQ B (CDR B)) (GO B)

---SUBST (COND ((NULL C) (RETURN (CAR D))) (RETURN (NCONC D C))))

---))

---(LAMBDA (X) (COMPILE (DEFINE X))) ((

---(COMITMATCH

---(LAMBDA (RULE WORKSPACE) (COMITMATCH2 (NAMES\_RULE) WORKSPACE)))

---(COMITMATCH2

---(LAMBDA (RULE WORKSPACE) (PROG (A B) (SETQ A (CMATCH\_RULE

---WORKSPACE NIL)) (COND ((NULL A) (RETURN NIL))

---((EQ A (QUOTE \$IMP)) (RETURN NIL))) (SETQ B (CONS

---(QUOTE WSEND) (CDR A)) (RETURN (ADDLAST (CAR A) B))))

---(CMATCH

---(LAMBDA (RULE WORKSPACE MPAIRS) (PROG

```

(RNAME A B C D E G F) (SETQ RNAME (CAR RULE ))
(SETQ RULE (CDR RULE)) (SETQ B (CAR RULE ))
(COND ((NULL RULE) (RETURN (CONS MPAIRS WORKSPACE)))
((EQ B (QUOTE *T)) (GO *T))
((EQ B (QUOTE *U)) (GO *U))
((EQ B (QUOTE $)) (GO PDOLL)) )
(SETQ H (CAR B)) (COND
((EQ H (QUOTE *P)) (GO PRINT))
((EQ H (QUOTE FN)) (SETQ B (CDR B)))
((NULL WORKSPACE) (RETURN (QUOTE $IMP))))
((EQ H (QUOTE $)) (GO NDOLL))
((EQ H (QUOTE *)) (GO EVAL))
((EQ H (QUOTE QUOTE)) (GO ATB1))
(T (GO ATB)))
(SETQ E (CONS WORKSPACE (CCMITRIN MPAIRS (CDR B))))
(SETQ B (COPYTP (APPLY (CAR B) E NIL)))
WATB (COND ((NULL B) (RETURN NIL))
((EQ B (QUOTE $IMP)) (RETURN B))
(T (RETURN (CMATCH (CONS (CDR RNAME) (CDR RULE)) (CDR B)
(ADDLAST MPAIRS (CONS (CAR RNAME) (CAR B)))))))
PDOLL (SETQ D (CDR RNAME)) (SETQ RULE (CDR RULE))
(COND ((NULL RULE) (RETURN (LIST (ADDLAST MPAIRS
(CONS (CAR RNAME) WORKSPACE))))))
DLCOP (SETQ B (CMATCH (CONS D RULE) WORKSPACE MPAIRS))
(COND ((NULL WORKSPACE) (RETURN NIL))
((EQ B (QUOTE $IMP)) (RETURN B))
(B (RETURN (CONS (ADDLAST (CAR B) (CONS (CAR RNAME) C
)) (CDR B))))))
(SETQ C (ADDLAST C (CAR WORKSPACE)))
(SETQ WORKSPACE (CDR WORKSPACE)) (GO DLCOP)
SUBMCH (SETQ B (SUBMCH B WORKSPACE)) (GO WATB)
*T (SETQ TRACK *T*) (GO $IMP)
*U (SETQ TRACK NIL) (GO $IMP)
PRINT (PRINT (CDR B)) (PRINT WORKSPACE)
$IMP (RETURN (QUOTE $IMP))
EVAL (SETQ B (EVAL (CADR B) NIL)) (GO ATB2)
ATB1 (SETQ B (CADR B)) (GO ATB2)
ATB (COND ((ATOM B) (SETQ B (GTNAME B MPAIRS))))
ATB2 (SETQ H (CAR WORKSPACE))
(COND ((ATOM B) (GO B))
((EQ (CADR B) (QUOTE /)) (GO SUBMCH))
((EQUAL B H) (SETQ B (CONS (LIST B) (CDR WORKSPACE))))
(T (SETQ B NIL))) (GO WATB)
B (COND ((EQUAL B H) (SETQ B WORKSPACE)) ((AND
(EQUAL B (CAR H)) (EQ (CADR H) (QUOTE /))) (SETQ B (CONS (LIST H)
(CDR WORKSPACE)))) (T (SETQ B NIL))) (GO WATB)
NDOLL (SETQ G (CDR B)) (SETQ B (DOLNM G WORKSPACE )) (GO WATB )))
(NAMER
(LAMBDA (X) (PROG (A B C D E G) (SETQ D O)
(SETQ A (CAR X))
(COND ((EQ A (QUOTE $)) (GO START)) ((EQ (CADR A) (QUOTE $)) (GO START)))
(SETQ A (QUOTE $)) (GO IN)
START (SETQ D (ADD1 C)) (COND ((NULL X) (RETURN (CONS E C))))
(SETQ A (CAR X)) (SETQ B (CDR A)) (SETQ X (CDR X))
(SETQ G (CAR A))
IN (COND ((ATOM A) (GO SNAME))
((CR (EQ G (QUOTE $)))
(EQ G (QUOTE FN))
(EQ G (QUOTE *))
(EQ G (QUOTE *P))
(EQ G (QUOTE QUOTE))
(EQ (CAR B) (QUOTE /)))

```

```

(GO SNAME))
((NULL B) (GO SNMA)))
(SETQ E (ADCLAST E (CAR A))) (SETQ A (CAR B)) (GO OUT)
SNMA (SETQ A (CAR A)) SNAME (SETQ E (ADCLAST E D))
OUT (SETQ C (ADCLAST C A)) (GO START) )))
(SUBMCH
(LAMBDA (X Y) (PRG (A B C)
(SETQ A (CAR X))
(SETQ B (CAR Y)) (COND
((NOT (OR (EQUAL A (QUOTE ($ .1))) (EQUAL A (CAR B)))) (RETURN NIL)))
(COND
((EQ (CADR B) (QUOTE /)) (GO ON)) (T (RETURN NIL)))
ON (SETQ A (CADR X)) (COND ((EQ (CAR A) (QUOTE /)) (GO A)))
(PRINT (LIST (QUOTE (SUBSCRIPT ERROR SUBMCH)) X)) (RETURN NIL)
A (SETQ A (CADR A)) (SETQ C (CADR B))
START (COND ((NULL A) (RETURN (CONS (LIST B) (CADR Y))))
((MEMBER (CAR A) C)
(SETQ A (CADR A)) (T (RETURN NIL))) (GO START))))
(CCLNM
(LAMBDA (NUM WSPACE) (PRG (A) (COND
((NOT (EQUAL NUM 1)) (GO START)))
(COND ((ATOM (CAR WSPACE)) (RETURN WSPACE)))
(RETURN (CONS (LIST (CAR WSPACE)) (CADR WSPACE)))
START (COND ((EQUAL NUM 0) (RETURN (CONS A WSPACE)))
((NULL WSPACE) (RETURN (QUOTE $IMP)))) (SETQ A (ADCLAST A (CAR WSPACE)))
(SETQ WSPACE (CADR WSPACE)) (SETQ NUM (SUB1 NUM)) (GO START)))
(ADCLAST
(LAMBDA (X Y) (APPEND X (LIST Y))))

```

```

(WRITES
(LAMBDA (X) (PRG (A) START (SETQ A (CAR X))
(COND ((NULL X) (RETURN NIL)) ((EQ A (QUOTE $EOR$)) (GO ON))
(ATOM A) (PRIN1 A)) (T (PRIN1 (QUOTE ***))))
(SETQ X (CADR X)) (GO START)
ON (TERPRI) (RETURN NIL) )))
(PRINTWS (LAMBDA (X Y) (PRG () (PRINT Y) (PRINT X)
(RETURN (QUOTE $IMP))))
(TIME (LAMBDA () (PRG ()) (TEMPUS-FUGIT) (RETURN NIL)))
(TRACE RULE (LAMBDA (X) (PRG ()
(SETQ TRACK *T*) (RETURN (QUOTE $IMP))))
(UNTRACE RULE (LAMBDA (X) (PRG ()
(SETQ TRACK NIL) (RETURN (QUOTE $IMP))))
))
(STOP))))))

```

..MACRO -- CORRESPONDING TO AI MEMO 57

..CONTENTS ONE SETSET

SETSET MACRO IS DEFINED HEREIN

```

DEFINE(( 00010
(MACRO (LAMBDA (L) (DEFLIST L (QUOTE MACRO)))) 00020
00030
(MDEF (LAMBDA (L) (COND 00040
((ATOM L) L) 00050
((EQ (CAR L) (QUOTE QUOTE)) L) 00060
((MEMBER (CAR L) (QUOTE (LAMBDA LABEL PRG)))) 00070
(CONS (CAR L) (CONS (CADR L) (COND
((EQ (CAR L) (QUOTE PRG)) (MAPLIST (CADR L) (FUNCTION (LAMBDA (J)
(MDEF (CAR J))))))
(T (MDEF (CADR L)))))) ))
((GET (CAR L) (QUOTE MACRO)) (MDEF ((GET (CAR L) (QUOTE MACRO))
L))) 00090
(T (MAPLIST L (FUNCTION (LAMBDA (J) (MDEF (CAR J)))))) )) 00100
)) 00110
)) 00120

```

```

DEFLIST (LAMBDA (L PRO) (DEFLIST (MDEF L))) 00140
)) 00150
STOP)))))))))))))
..DEFINELIST D J EDWARDS
..BASIS FOR PRINT IN NICE FORMAT(DOESNT CONTAIN ,PRETTYPRINT,
SET
DEFLIST 00010
(( (DEFINELIST (LAMBDA (L A) (PROG (LL DL) (SETQ LL(CAR L))
00020
(SETQ DL T)) 00030
(PRINT (QUOTE $$$/FUNCTION DEFINITIONS/2)) 00040
(TERPRI) (PRIN1 (QUOTE $$$DEFINELIST (($)) 00050
A (TERPRI) (TERPRI) (COND ((NULL LL) (GO AA))) 00060
(PRIN1 LPAR) 00070
(PRINT (CAAR LL)) 00080
(PRINTDEF (CADAR LL)) (COND ((NULL (CDDAR LL)) (GO AC))) 00090
(TERPRI) 00100
(PRINT (QUOTE $$$WRONG FORMAT HERE, NO DEFINE$)) 00110
(PRINT (QUOTE $$$LEFT OVER$)) (PRINT (CDDAR LL)) 00120
(SETQ DL NIL) AC (SETQ LL (CDR LL)) 00130
(PRIN1 RPAR) 00140
(GO A) 00150
AA (PRINT (QUOTE $$$))$)) (COND ((NULL (CDR L)) (GO AB))) 00160
(TERPRI) (PRINT (QUOTE $$$LEFT OVER$)) (PRINT (CDR L)) 00170
(SETQ DL NIL) 00180
AB (COND (DL (RETURN (DEFINE (LIST (CAR L))))) 00190
(T (RETURN NIL))) 00200
)) ) FEXPR) DEFINE (( 00210
(PRINTDEF (LAMBDA (E) (PROG (I IUNIT IUNITL) 00220
(SETQ I 1) 00230
(SETQ IUNIT (QUOTE $$$ $)) 00240
(SETQ IUNITL 3) 00250
(PRIN1 IUNIT) 00260
(SUPERPRINT E) 00270
(RETURN NIL) 00280
))) 00290
00300
(SUPERPRINT (LAMBDA (E) (COND
00310
((ATCM E) (PRIN1 E)) 00320
(T (PROG (EP M) 00330
(SETQ EP E) 00340
(PRIN1 LPAR) 00350
A (COND 00360
((MEMBER (CAR EP) (QUOTE (AND OR COND PLUS TIMES IF SELECT LIST PROG2
00000370
MAX MIN))) 00000375
(GO PL)) 00380
((EQ (CAAR EP) (QUOTE LAMBDA)) (GO PL)) 00390
((EQ (CAR EP) (QUOTE PROG)) (GO PP))) 00400
(SUPERPRINT (CAR EP)) 00410
(SETQ EP (CDR EP)) 00420
(COND ((NULL EP) (RETURN (PRIN1 RPAR))) 00430
((ATCM EP) (GO PD))) 00440
(PRIN1 BLANK) 00450
(GO A) 00460
PK (SETQ I (SUB1 I)) 00470
PD (PRIN1 BLANK) 00480
(PRIN1 PERIOD) 00490
(PRIN1 BLANK) 00500
(PRIN1 EP) 00510
(RETURN (PRIN1 RPAR)) 00520
PL (SETQ I (ACC1 I)) 00530

```

(SUPERPRINT (CAR EP))	00540
PM	00550
(SETQ EP (CDR EP))	00560
)	00570
(COND ((NULL EP) (GO PJ))	00580
((ATOM EP) (GO PK)))	00590
)	00600
(ENDLINE)	00610
(SUPERPRINT (CAR EP))	00620
(GO PM)	00630
PJ (SETQ I (SUB1 I))	00640
(RETURN (PRIN1 RPAR))	00650
FP (PRIN1 (CAR EP))	00660
(SETQ EP (CDR EP))	00670
(SETQ I (ADD1 I))	00680
(COND ((NULL EP) (GO PJ))	00690
((ATOM EP) (GO PK)))	00700
(PRIN1 BLANK)	00710
(SUPERPRINT (CAR EP))	00720
PY (SETQ EP (CDR EP))	00730
(COND ((NULL EP) (GO PJ))	00740
((ATOM EP) (GO PK)))	00750
(ENDLINE)	00760
(COND ((ATOM (CAR EP)) (GO PZ)))	00770
(PRIN1 IUNIT) (PRIN1 IUNIT)	00780
PX (SETQ I (PLUS I 2))	00790
(SUPERPRINT (CAR EP))	00800
(SETQ I (PLUS I -2))	00810
(GO PY)	00820
PZ (PRIN1 (CAR EP))	00830
(SETQ M (PLUS IUNIT IUNIT (MINUS (LENGTH	00840
(UNPACK (CAR (GET ( CAR EP) (QUOTE PNAME))))))	00850
AA (SETQ M (SUB1 M))	00860
(PRIN1 BLANK)	00870
(COND ((NOT (OR (ZEROP M) (MINUSP M))) (GO AA)))	00880
(SETQ EP (CDR EP))	00890
(COND ((NULL EP) (GO PJ))	00900
((ATOM EP) (GO PK)))	00910
((ATOM (CAR EP)) (GO PZ)))	00920
(GO PX)	00930
))))	00940
(ENDLINE (LAMBDA () (PROG (J)	00950
(SETQ J I)	00960
(TERPRI)	00970
A (COND ((ZEROP J) (RETURN NIL))	00980
((MINUSP J) (ERROR I)))	00990
(PRIN1 IUNIT)	01000
(SETQ J (SUB1 J))	01010
(GO A) )	01020
)	01030
STOP )))))))) ))))))) )))))))	01040
..MATHREAD        THERE IS A STANFORD A I MEMO ABOUT THIS	
TEST        MATHREAD	
DEFINE ((	
(MATHREAD (LAMBDA () (PROG ()	
(STARTREAD)	
(CLEARBUFF)	
LOCK    (COND ((EQ CURCHAR LPAR) (GO START) ))	
(ADVANCE)	

```
      (CC LOCK)
START (RETURN (GRCUP (LIST (LIST (INPUT) )) (INPUT) (INPUT) ))
      )))
```

```
(GROUP (LAMBDA (U F S) (PRCG (X Y Z)
 (SETQ X (GET H (QUOTE PREFIX) ))
 (SETQ Y (GET S (QUOTE PR) ))
 (SETQ Z (GET (CAAR U) (QUOTE PL) ))
 (CCND ((NULL X) (GO IND))
        ((EQ X (QUOTE YES)) (RETURN
 (GRCUP (CCND ((EQ H (QUOTE PLUS)) U)
           (T (CONS (LIST H) U) )) S (INPUT) )))))
```

```
(PRINT (LIST (QUOTE INPUTERRCR1) H))
 (RETURN (GRCUP (NCCNC U (LIST NIL)) S (INPUT) ))
IND (CCND ((NOT (NULL Y)) (GC OK)))
      (PRINT (LIST (QUOTE INPUTERRCR2) S))
```

```
(RETURN (GRCUP (NCCNC U (LIST NIL)) F (INPUT) ))
OK (RETURN (CCND
 ((EQ S (QUOTE LPAR)) (GROUP (CONS (LIST S) U) H (QUOTE COMMA) ))
 ((GREATERP Y Z) (GRCUP (CONS (LIST S H) U) (INPUT) (INPUT) ))
 ((LESSP Y Z) (GROUP (CDR U) (NCCNC (CAR U) (LIST H) S))
 ((ZEROP Y) (COND ((NULL (CDR U)) H)
```

```
                ((NULL (CDR U)) (ERROR H))
                (T (GRCUP (CDR U)
 (CCND ((EQ (CAR H) (QUOTE COMMA)) (CDR H)) (T H)) (INPUT) )) ))
 (T (GRCUP (CONS (NCCNC (CAR U) (LIST H) (CDR U)) (INPUT) (INPUT) ))
      )))
```

```
(INPUT (LAMBDA ( ) (PRCG (X)
AA (CCND ((LITER CURCHAR) (GO LET))
          ((DIGIT CURCHAR) (GO DIG))
 (SETQ X (GET CURCHAR (QUOTE SWITCH) ))
 (CCND ((EQ X (QUOTE BLANK)) (GO IG))
        ((EQ X (QUOTE STAR)) (GO CASTOR))
        ((EQ X (QUOTE DIFFERENCE)) (GO SUB))
        ((NULL X) (GC ERR))
```

```
(ADVANCE)
 (RETURN X)
LET (PACK CURCHAR)
     (CCND ( (OR (LITER (ADVANCE)) (DIGIT CURCHAR)) (GO LET)))
     (RETURN (INTERN (MKNAM))))
```

```
DIG (PACK CURCHAR)
     (CCND ((OR (DIGIT (ADVANCE))
                (EQ CURCHAR PERIOD))
            (EQ CURCHAR (QUOTE E))) (GO DIG)))
     (RETURN (NUMOB))
```

```
IG (ADVANCE)
    (GC AA)
CASTOR (CCND ((EQ (ADVANCE) STAR) (GO POLLUX) ))
        (RETURN (QUOTE TIMES) )
```

```
POLLUX (ADVANCE)
        (RETURN (QUOTE EXPT) )
```

```
SUB (CSETQ CURCHAR (QUOTE MINUS) )
     (RETURN (QUOTE PLUS) )
```

```
ERR (ERRCR1)
     (GC JG)
     )))
```

```
DEFLIST ( ( ($+$ PLUS) ($-$ DIFFERENCE) (MINUS MINUS)
           ( / STAR) ( / QUOTIENT) (= EQUAL) ($+$ COMMA)
           ($$$ $ DCT) ($$$($ LPAR) ($$$) $ RPAR)
           )
```

(3EGR\$ BLANK) (### \$ BLANK) SWITCH)

DEFLIST ( (PLUS 30) (DIFFERENCE 50) (MINUS 50) (TIMES 60)  
(QUOTIENT 70) (EXPT 90) (EQUAL 20) (COMMA 10)  
(DOT 110) (LPAR 0) (RPAR 0) ) PL)

DEFLIST ( (PLUS 30) (DIFFERENCE 40) (MINUS 40) (TIMES 60)  
(QUOTIENT 80) (EXPT 100) (EQUAL 20) (COMMA 10)  
(DOT 120) (LPAR 0) (RPAR 0) ) PR)

DEFLIST ( (PLUS YES) (DIFFERENCE NO) (MINUS YES) (TIMES NO)  
(QUOTIENT NO) (EXPT NO) (EQUAL NO) (COMMA NO)  
(DOT NO) (LPAR YES) (RPAR NO) ) PREFIX)

MATHREAD ( )

MATHREAD ( )

MATHREAD ( )

MATHREAD ( )

MATHREAD ( )

MATHREAD ( )

STCP ))) ))) ))) )))

(4.2)

(A + 4E2 + 4.2E3)

(A+F(.23 + 23.))

(A + \* (B+C) + F(.23

\* (Y + 5.6.7)))

(A + B C (X,,Y,(Z + 1.2E6E7)))

(3.3.3E3E3 \*\*\*\* F)

FIN END OF LISP RUN

..SAUNDERS READABLE 7090 LISP 1.5 PUNCHUP OF THE COMPIER

TEST TRACE COMPILER TEST

DEFINE ( (

(TRACASET (LAMBDA (L) (TRACSQ L (FUNCTION TRACSS))))

ARYT0030

(UNTRACASET (LAMBDA (L) (TRACSQ L (FUNCTION TRACSU))))

ARYT0040

(TRACSQ (LAMBDA (L FN) (MAPCON L (FUNCTION (LAMBDA (J) (PRG2 (A)  
(SETQ A (PROP (CAR J) (QUOTE EXPR) (FUNCTION (LAMBDA () (PRG2 (PRINT  
(CONS (CAR J) (QUOTE (IS NOT EXPR DEFINED)))) NIL))))))

ARYT0060

ARYT0070

ARYT0090

ARYT0100

(SETQ A (CADDR A))

(COND ((NOT (EQ (CAR A) (QUOTE PROG))) (RETURN (PRG2 (PRINT (CONS (CAR J)

(QUOTE (IS NOT A PROGRAM)))) NIL) )))

ARYT0130

(SETQ A (CDR A))

ARYT0140

TEST (SETQ A (CDR A))

ARYT0150

(COND ((NULL A) (RETURN (LIST (CAR J))))

ARYT0160

((EQ (CAAR A) (QUOTE SETQ)) (FN A)) )

(GC TEST) ))) )))

ARYT0190

(TRACSS (LAMBDA (L) (RPLACD L (COND (LIST

ARYT0210

(QUOTE (PRINT BLANK))

ARYT0220

(LIST (QUOTE PRINT) (LIST (QUOTE QUOTE) (LIST (CADAR L) EQSIGN))))

(LIST (QUOTE PRINT) (CADAR L) ) (CDR L))))

ARYT0240

(TRACSU (LAMBDA (L) (RPLACD L (CDDDDR L))))

ARYT0260

(LENGTH (LAMBDA (M) (PRG2 (N) (SETQ N 0)

A (COND ((NULL M) (RETURN N)) ) (SETQ N (ADD1 N)) (SETQ M (CDR M))

(GC A) )))

(REVERSE (LAMBDA (X) (PRG2 (Y)