Make Translator Listing

```
//JCRLISP JOB (205203,20,0,5),'REYNOLDS COL 1 ROW 3',MSGLEVEL=1
 REYNOLDS J  05203AMD344X51430-01145050
//JOBLIB DD DSNAME=C145.B17648.LISP.OBJECT,DISP=SHR
//  EXEC PGM=LISP
//COMP DD DSNAME=C145.B17648.LISP.COMPILER,DISP=SHR
//LISPOUT DD SYSOUT=A,DCB=BLKSIZE=798
//GEDANK DD UNIT=CELL,VOL=SER=CELLBE,SPACE=(CYL,(10,5),RLSE),
//         DSNAME=C145.B05203.GEDANKEN.APRIL29A,
//         DISP=(NEW,KEEP)
//LISPIN DD *
OPEN (COMP SYSFILE INPUT)
RESTORE (COMP)
CLOSE (COMP)
LAP360 ((
(SETC SUBR 3)  (L M 8 (R0 R12))  (L R0 0 (R0 M))  (ST R0 TEMP)
     (ST NILR 0 (R0 M))  (*LINK SASSOC 3)  (L R0 TEMP)
     (ST R0 4 (R0 A))  (LR A R0)  (*RETURN NIL)  TEMP (DC 0X) ) NIL )
LAP360 ((
(GENSYM2 SUBR 1)
     (*LOAD A (QUOTE TEMP))
     (*LINK GENSYM1 1)
     (*RETURN NIL)
) NIL )
FLAG ((
$$$ $ A B C D E F G H I $$$¢$ $$$.$ $$$<$
$$$($ $$$+$  $$$!$ $$$&$ J K L M N O P Q R
$$$!$ $$A$A $$$*$ $$$)$ $$$;$ $$$¬$ $$$-$ $$$/$ S T U V W
X Y Z $$$,$ $$$%$ $$$_$  $$$>$ $$$?$ $$$0$ $$$1$ $$$2$ $$$3$
$$$4$ $$$5$ $$$6$ $$$7$ $$$8$ $$$9$ $$$:$ $$$#$ $$$@$ $$$'$ $$$=$ $$$"$
) CHAROB)
CSET  (LAMBSIGN   $$$#$)
CSET  (COLON      $$$:$)
CSET  (SEMICOLON  $$$;$)
CSET  (QUOTESIGN  $$$"$)
CSET  (COLONEQ    $$$:=$)
DEFINE ((
(DEFINESYN (LAMBDA (L) (DEFLIST (DFSYN1 L) (QUOTE ABSYN))))
(DFSYN1 (LAMBDA (L) (COND ((NULL L) NIL)
     (T (CONS (LIST (CADAR L)(CONS (CAAR L) (CDDAR L)))
          (DFSYN1 (CDR L))))))
))))))))))))))))))))))))))))
DEFINESYN ((
(UNION EXP CONSTANT IDENT FUNCTDES LAMBDAEXP CONDEXP CASEEXP BLOCK)
(CLASS CONSTANT (VALUE VALUEDEN))
(CLASS IDENT (STRING ATOMCLASS))
(CLASS FUNCTDES (FUNCTPART EXP) (ARGPART EXP))
(CLASS LAMBDAEXP (PARAMPART IDENT) (BODY EXP))
(CLASS CONDEXP (PREMISS EXP) (CONCLUSION EXP) (ALTERNATIVE EXP))
(CLASS CASEEXP (INDEX EXP) (BODY SEQ EXP))
(CLASS BLOCK (RDECLPART SEQ RDECL) (LDECLPART SEQ LDECL)
     (BODY SEQ EXP))
(CLASS RDECL (LEFT IDENT) (RIGHT LAMBDAEXP))
(CLASS LDECL (LEFT IDENT) (RIGHT SEQ EXP))
(UNION PFORM IDENT SEQPFORM)
(CLASS SEQPFORM (BODY SEQ PFORM))
(CLASS STATE (CONTROL SEQ INST) (STACK SEQ VALUEDEN)
     (ENVR SEQ ENVELEM) (DUMP SEQ DUMPELEM)
     (MEMORY SEQ VALUEDEN) (ATOMCOUNT INTCLASS))
(CLASS DUMPELEM (CONTROL SEQ INST) (STACK SEQ VALUEDEN)
     (ENVR SEQ ENVELEM))
(UNION VALUEDEN INTCLASS BOOLDEN CHARCLASS ATOMDEN FUNCTDEN
     REFDEN LABELDEN)
```

```
(UNION BOOLDEN TRUEDEN FALSEDEN)
(CLASS TRUEDEN)
(CLASS FALSEDEN)
(UNION ATOMDEN LLDEN ULDEN PROGATOMDEN)
(CLASS LLDEN)
(CLASS ULDEN)
(CLASS PROGATOMDEN (NAME INTCLASS))
(CLASS FUNCTDEN (CONTROL SEQ INST) (ENVR SEQ ENVELEM))
(UNION REFDEN EXPREFDEN IMPREFDEN)
(CLASS EXPREFDEN (ADDRESS INTCLASS))
(CLASS IMPREFDEN (SETF FUNCTDEN) (VALF FUNCTDEN))
(UNION LABELDEN ERRORDEN PROGLABELDEN)
(CLASS ERRORDEN)
(CLASS PROGLABELDEN (BODY SEQ EXP) (ENVR SEQ ENVELEM)
    (DUMP SEQ DUMPELEM))
(UNION ENVELEM ENVMARK ENVPAIR)
(CLASS ENVMARK)
(CLASS ENVPAIR (LEFT IDENT) (RIGHT ENVVALUEDEN))
(UNION ENVVALUEDEN VALUEDEN RECFUNCTDEN RECLABELDEN)
(CLASS RECFUNCTDEN (CONTROL SEQ INST))
(CLASS RECLABELDEN (BODY SEQ EXP) (DUMP SEQ DUMPELEM))
(UNION INST EXP RDECL LDECL EXEC BRANCH SELECT BIND APPLY
    MARKENV DELETE BASICFUNCTINST)
(CLASS EXEC (BODY SEQ EXP))
(CLASS BRANCH (CONCLUSION EXP) (ALTERNATIVE EXP))
(CLASS SELECT (BODY SEQ EXP))
(CLASS BIND (BODY IDENT))
(CLASS APPLY)
(CLASS MARKENV)
(CLASS DELETE)
(CLASS BASICFUNCTINST (STRING ATOMCLASS))
(CLASS QSTOKEN (STRING SEQ CHARCLASS))
))))))))))))))))))))))))))))
DEFINE ((
(ERRM (LAMBDA (N A) (PROG () (TERPRI) (ERROR (LIST N A)))))
))))))))))))))))))))))))))))
DEFINE((
(MACEVAL (LAMBDA (X) (COND
    ((ATOM X) (COND
        ((NUMBERP X) X)
        ((MEMBER X (QUOTE (NIL T F OBLIST ALIST))) X)
        ((MEMBER (QUOTE APVAL) (CDR X))
            (LIST (QUOTE QUOTE) (EVAL X NIL)))
        (T X)))
    ((ATOM (CAR X)) (COND
        ((EQ (CAR X) (QUOTE COND)) (CONS (CAR X) (MACEVCON (CDR X))))
        ((EQ (CAR X) (QUOTE FUNCTION))
            (LIST (CAR X) (MACAPPLY (CADR X))))
        ((EQ (CAR X) (QUOTE PROG))
            (CONS (CAR X) (CONS (CADR X) (MACEVPROG (CDDR X)))))
        ((OR (EQ (CAR X) (QUOTE SETQ)) (EQ (CAR X) (QUOTE CSETQ)))
            (LIST (CAR X) (CADR X) (MACEVAL (CADDR X))))
        ((OR (EQ (CAR X) (QUOTE QUOTE)) (EQ (CAR X) (QUOTE GO))) X)
        (T (PROG (Y) (SETQ Y (GET (CDAR X) (QUOTE MACRR)))
            (RETURN (COND
                ((NULL Y) (CONS (CAR X) (MACEVLIS (CDR X))))
                (T (MACEVAL (Y (CDR X))))))))))
    (T (CONS (MACAPPLY (CAR X)) (MACEVLIS (CDR X)))))))
(MACEVLIS (LAMBDA (X) (COND ((NULL X) NIL)
    (T (CONS (MACEVAL (CAR X)) (MACEVLIS (CDR X)))))))
(MACEVCON (LAMBDA (X) (COND ((NULL X) NIL)
    (T (CONS (MACEVLIS (CAR X)) (MACEVCON (CDR X))))))))
```

```
(MACEVPROG (LAMBDA (X) (COND ((NULL X) NIL)
    ((ATOM (CAR X)) (CONS (CAR X) (MACEVPROG (CDR X))))
    (T (CONS (MACEVAL (CAR X)) (MACEVPROG (CDR X)))))))
(MACAPPLY (LAMBDA (X) (COND
    ((ATOM X) X)
    ((EQ (CAR X) (QUOTE LAMBDA))
        (LIST (CAR X) (CADR X) (MACEVAL (CADDR X))))
    ((EQ (CAR X) (QUOTE LABEL))
        (LIST (CAR X) (CADR X) (MACAPPLY (CADDR X))))
    (T (MACEVAL X)))))
(MACEXPAND (LAMBDA (L) (COND ((NULL L) NIL) (T
    (PROG (E) (SETQ E (GET (CDAR L) (QUOTE EXPR)))
        (COND ((NULL E) (RETURN (MACEXPAND (CDR L))))
        (SETQ E (MACAPPLY E))
         (TERPRI)
        (PRINT (APPEND (QUOTE (MACRO EXPANSION OF)) (LIST (CAR L))))
        (TERPRI)
        (PRINT E)
        (DEFINE (LIST (LIST (CAR L) E)))
        (RETURN (CONS (CAR L) (MACEXPAND (CDR L))))))))))
(DEFMACRO (LAMBDA (L) (DEFLIST L (QUOTE MACRR))))
(SUBATOM (LAMBDA (X Y) (COND
    ((ATOM Y) (SUBATOM1 X Y))
    (T (CONS (SUBATOM X (CAR Y)) (SUBATOM  X (CDR Y)))))))
(SUBATOM1 (LAMBDA (X Y) (COND ((NULL X) Y)
    ((EQ (CAAR X) Y) (CDAR X)) (T (SUBATOM1 (CDR X) Y)))))
(CHECKM (LAMBDA (V D)
    (AND (NOT (NULL D)) (EQ (CAR D) (QUOTE CLASS))
        (EQUAL (LENGTH V) (LENGTH (CDR D))))))
(PFIND (LAMBDA (X G D) (COND
    ((AND (NOT (NULL D)) (EQ (CAR D) (QUOTE CLASS)))
        (PFIND1 X G (CDR D)))
    (T (ERRM 10 (LIST X G D))))))
(PFIND1 (LAMBDA (X G D) (COND
    ((NULL D) (ERRM 11 (LIST X G)))
    ((EQ (CAAR D) G) X)
    (T (LIST (QUOTE CDR) (PFIND1 X G (CDR D)))))))
(HMAC1 (LAMBDA (X S L) (COND
    ((EQ S (QUOTE INTCLASS))
        (ADDEL (SBQ (FIXP X) (X) (X)) L))
    ((EQ S (QUOTE ATOMCLASS))
        (ADDEL (SBQ (ATOM X) (X) (X)) L))
    ((EQ S (QUOTE CHARCLASS))
        (ADDEL (SBQ (AND (ATOM X) (MEMBER (QUOTE CHAROB) (CDR X)))
            (X) (X)) L))
    ((EQ S (QUOTE UNIVERSAL)) (ADDEL (QUOTE T) L))
    (T (HMAC2 X S L (GET (CDR S) (QUOTE ABSYN)))))))
(HMAC2 (LAMBDA (X S L D) (COND
    ((NULL D) (ERRM 7 S))
    ((EQ (CAR D) (QUOTE CLASS)) (ADDEL
        (SBQ (EQ (CAR X) (QUOTE S)) (X S) (X S)) L))
    ((EQ (CAR D) (QUOTE UNION)) (HMAC3 X (CDR D) L))
    (T (ERRM 8 S)))))
(HMAC3 (LAMBDA (X U L) (COND  ((NULL U) L)
    (T (HMAC1  X (CAR U) (HMAC3 X (CDR U) L))))))
(HMAC4 (LAMBDA (L) (COND
    ((NULL L) (QUOTE F)) ((NULL (CDR L)) (CAR L))
    (T (CONS (QUOTE OR) L)))))
(ADDEL (LAMBDA (X L) (COND
    ((MEMBER X L) L) (T (CONS X L)))))
(DEFANDEXP (LAMBDA (L) (PROG (X Y Z)
    (SETQ X (DEFINE L)) (SETQ Y X)
```

```
    LOOP
    (COND ((NULL Y) (RETURN X)))
    (SETQ Z (LIST (CAR Y))) (SETQ Y (CDR Y))
    (MACEXPAND Z)
    (COMPILE Z)
    (GO LOOP))))
))))))))))))))))))))))))))))
DEFLIST ((
(SBQ (LAMBDA (X A) (SUBATOM (PAIR (CADR X) (EVLIS (CADDR X) A))
    (CAR X))))
) FEXPR)
DEFMACRO ((
(EQQ (LAMBDA (L) (SBQ (EQ X (QUOTE Y)) (X Y) ((CAR L) (CADR L)))))
(MBOOL (LAMBDA (L) (SBQ (COND (X (M TRUEDEN)) (T (M FALSEDEN))) (X)
    ((CAR L)))))
(M (LAMBDA (L) (COND
    ((CHECKM (CDR L) (GET (CDAR L) (QUOTE ABSYN))) (COND
        ((NULL (CDR L)) (SBQ (QUOTE (C)) (C) ((CAR L))))
        (T (SBQ (LIST (QUOTE C).V) (C V) ((CAR L) (CDR L))))))
    (T (ERRM 9 L)))))
(P (LAMBDA (L) (LIST (QUOTE CADR) (PFIND (CAR L) (CADR L)
    (GET (CDR (CADDR L)) (QUOTE ABSYN))))))
(H (LAMBDA (L) (HMAC4 (HMAC1 (CAR L) (CADR L) NIL))))
))))))))))))))))))))))))
SPECIAL ((MR AC))
SPECIAL ((NXTK)) OVOFF ()
SPECIAL ((LIBSIGNAL))
DEFANDEXP((
(GTTK (LAMBDA () (PROG () (RETURN (SETQ NXTK (PROG (C S)
    START
    (SETQ C (READCH NIL))
    (COND ((EQ C BLANK) (GO START)) ((LETP C) (GO LID))
        ((DIGP C) (GO LIN)) ((EQ C QUOTESIGN) (GO LQS))
        ((EQ C COLON) (GO LST)) ((EQ C DOLLAR) (GO LBK)) (T (GO LPN)))
    LID
    (RLIT C)
    (SETQ C (READCH NIL))
    (COND ((OR (LETP C) (DIGP C)) (GO LID)))
    (READCH T)
    (SETQ S (MKATOM))
    (RETURN (COND ((MEMBER S
        (QUOTE (AND OR IF THEN ELSE CASE OF IS ISR))
        ) S) (T (M IDENT S))))
    LIN
    (RNUMB C)
    (SETQ C (READCH NIL))
    (COND ((DIGP C) (GO LIN)))
    (READCH T)
    (RETURN (M CONSTANT (MKATOM)))
    LQS1
    (SETQ S (APPEND1 S C))
    LQS
    (SETQ C (READCH NIL))
    (COND ((EQ C QUOTESIGN) NIL) ((EQ C DOLLAR) (READCH T))
        (T (GO LQS1)))
    (RETURN (M QSTOKEN S))
    LST
    (COND ((EQ (READCH NIL) EQSIGN) (RETURN COLONEQ)))
    LBK
    (READCH T)
    LPN
    (RETURN C)
```

```
    ))))))
(RCTOK (LAMBDA (X)
    (COND ((EQ NXTK X) (PROG () (GTTK) (RETURN T))) (T F))))
(CHK (LAMBDA (X) (COND (X X) (T (PARSERR)))))
(RCEXPZ (LAMBDA () (PROG (S1) (SETQ S1 NXTK)
    (COND ((OR (H S1 IDENT) (H S1 QSTOKEN) (H S1 CONSTANT)) (GO L1))
        ((RCTOK LPAR) (GO L2))) (RETURN F)
    L1 (GTTK) (RETURN S1)
    L2 (SETQ S1 (CHK (RCBLOCK2))) (CHK (RCTOK RPAR))
    (RETURN (LIST 1 S1)))))
(RCEXP1 (LAMBDA () (PROG (S1 S2) (SETQ S1 (RCEXPZ))
    (COND (S1 (GO L1))) (RETURN F)
    L1 (SETQ S2 (RCEXP1))
    (COND (S2 (RETURN (LIST 2 S1 S2))) (T (RETURN S1))))))
(RCEXP2 (LAMBDA () (PROG (S1) (SETQ S1 (RCEXP1))
    (COND (S1 (GO L1))) (RETURN F)
    L1 (COND ((RCTOK EQSIGN) (RETURN (LIST 3 S1 (CHK (RCEXP2))))))
    (RETURN S1))))
(RCEXP3 (LAMBDA () (PROG (S1) (SETQ S1 (RCEXP2))
    (COND (S1 (GO L1))) (RETURN F)
    L1 (COND ((RCTOK (QUOTE AND)) (RETURN (LIST 4 S1 (CHK (RCEXP3))))))
    (RETURN S1))))
(RCEXP4 (LAMBDA () (PROG (S1) (SETQ S1 (RCEXP3))
    (COND (S1 (GO L1))) (RETURN F)
    L1 (COND ((RCTOK (QUOTE OR)) (RETURN (LIST 5 S1 (CHK (RCEXP4))))))
    (RETURN S1))))
(RCEXP5 (LAMBDA () (PROG (S1 S2)
    (COND ((RCTOK (QUOTE IF)) (GO L2)) ((RCTOK LAMBSIGN) (GO L3)))
    (SETQ S1 (RCEXP4))
    (COND (S1 (GO L1))) (RETURN F)
    L1 (COND ((RCTOK COLONEQ) (RETURN (LIST 8 S1 (CHK (RCEXP5))))))
    (RETURN S1)
    L2 (SETQ S1 (CHK (RCEXP6))) (CHK (RCTOK (QUOTE THEN)))
    (SETQ S2 (CHK (RCEXP6))) (CHK (RCTOK (QUOTE ELSE)))
    (RETURN (LIST 6 S1 S2 (CHK (RCEXP5))))
    L3 (SETQ S1 (CHK (RCEXPZ)))
    (RETURN (LIST 7 S1 (CHK (RCEXP5)))))))
(RCEXP6 (LAMBDA () (PROG (S1)
    (COND ((RCTOK (QUOTE CASE)) (GO L5)))
    (SETQ S1 (RCEXP5))
    (COND (S1 (GO L1))) (RETURN (LIST 9))
    L1 (COND ((RCTOK COMMA) (GO L2))) (RETURN S1)
    L2 (SETQ S1 (LIST 9 S1 (CHK (RCEXP5))))
    L3 (COND ((RCTOK COMMA) (GO L4))) (RETURN S1)
    L4 (SETQ S1 (APPEND1 S1 (CHK (RCEXP5)))) (GO L3)
    L5 (SETQ S1 (CHK(RCEXP6))) (CHK (RCTOK (QUOTE OF)))
    (SETQ S1 (LIST 10 S1 (CHK (RCEXP5)))) (GO L3))))
(RCBLOCKZX (LAMBDA (S1) (PROG ()
    (COND ((RCTOK SEMICOLON) (GO L1)) ((RCTOK COLON) (GO L2)))
    (RETURN (LIST 11 S1))
    L1 (RETURN (LIST 12 S1 (CHK (RCBLOCKZX (CHK (RCEXP6))))))
    L2 (RETURN (LIST 13 S1 (CHK (RCBLOCKZX (CHK (RCEXP6)))))))))
(RCBLOCK1X (LAMBDA (S1) (PROG (S2 S3)
    (COND ((RCTOK (QUOTE ISR)) (GO L1))) (RETURN (RCBLOCKZX S1))
    L1 (CHK (RCTOK LAMBSIGN)) (SETQ S2 (CHK (RCEXPZ)))
    (SETQ S3 (CHK (RCEXP5))) (CHK (RCTOK SEMICOLON))
    (RETURN (LIST 14 S1 S2 S3 (CHK (RCBLOCK1X (CHK (RCEXP6)))))))))
(RCBLOCK2X (LAMBDA (S1) (PROG (S2)
    (COND ((RCTOK (QUOTE IS)) (GO L1)))
        (RETURN (LIST 16 (RCBLOCK1X S1))))
    L1 (SETQ S2 (CHK (RCEXP6))) (CHK (RCTOK SEMICOLON))
    (RETURN (LIST 15 S1 S2 (CHK (RCBLOCK2X (CHK (RCEXP6))))))))))
```

```
(RCBLOCK2 (LAMBDA ()(RCBLOCK2X (CHK (RCEXP6))))
(PARSE (LAMBDA () (PROG (NXTK S)
    (GTTK) (SETQ S (CHK (RCBLOCK2))) (CHK (RCTOK DOLLAR))
    (READCH NIL) (RETURN S))))
(PARSERR (LAMBDA () (PROG (C)
    (PRINT (QUOTE (ILL FORMED INPUT $$$,$ REMAINING STRING IS)))
    (TERPRI)
    LOOP (SETQ C (READCH NIL)) (PRIN1 C)
    (COND ((NOT (EQ C DOLLAR)) (GO LOOP)))
    (TERPRI)
    (ERRM 12 NIL))))
))))))))))))))))))))))))))
DEFANDEXP((
(GETVALPREDEF (LAMBDA (X) (PROG (R)
    (SETQ R (GET (CDR (P X STRING IDENT)) (QUOTE PREDEF)))
    (COND ((NULL R) (ERRM 16 X)))
    (RETURN (CAR R)))))
(DEFPREDEF (LAMBDA () (PROG (IDSTRS X C ONE TWO P1 P2 P3 P4 P5 P6 P7
    PLIST D R)                          (SETQ IDSTRS (QUOTE (
ISINTEGER ISBOOLEAN ISCHAR ISATOM ISFUNCTION
ISREF ISLABEL ATOM NCREF REF IMPREF NCSET SET VAL
COERCE GOTO NCEQUAL EQUAL GREATER CHARGREATER INC DEC
READCHAR WRITECHAR
NEG ADD SUBTRACT MULTIPLY DIVIDE REMAINDER
NOT
PRINT TERPRI
TRUE FALSE QUOTECHAR LL UL ERROR DOLLARCHAR)))
     (SETQ X (CRIDENT))  (SETQ C (M BASICFUNCTINST (QUOTE COERCE)))
    (SETQ ONE (M CONSTANT 1)) (SETQ TWO (M CONSTANT 2))
    (SETQ P1 (LIST (M DELETE))) (SETQ P2 NIL) (SETQ P3 (LIST C))
    (SETQ P4 (LIST (M BIND X) X C ONE (M APPLY) X C TWO (M APPLY)))
    (SETQ P5 (LIST (M BIND X) X C ONE (M APPLY) X C TWO (M APPLY) C))
    (SETQ P6 (LIST (M BIND X) X C ONE (M APPLY) C X C TWO (M APPLY) C))
    (SETQ P7 (LIST (M BIND X) X C ONE (M APPLY) C
        X C TWO (M APPLY) C X C (M CONSTANT 3) (M APPLY) C))
    (SETQ PLIST (LIST
P3 P3 P3 P3 P3 P2 P3 P1 P2 P3 P6 P4 P5 P2 P2 P3
P4 P6 P6 P6 P3 P3 P1 P3
P3 P6 P6 P6 P6 P6
P3
P2 P1
        ))
    (SETQ D (LIST (M TRUEDEN) (M FALSEDEN) QUOTESIGN (M LLDEN)
        (M ULDEN) (M ERRORDEN) DOLLAR))
    (SETQ R (DEFPD2 IDSTRS (APPEND (DEFPD1 PLIST IDSTRS) D)))
    (DEFLIST R (QUOTE PREDEF))
    (RETURN R))))
(DEFPD1 (LAMBDA (PL IL) (COND ((NULL PL) NIL)
    (T (CONS (M FUNCTDEN (APPEND (CAR PL) (LIST
        (M BASICFUNCTINST (CAR IL)))) NIL) (DEFPD1 (CDR PL) (CDR IL))
        )))))
(DEFPD2 (LAMBDA (IL DL) (COND ((NULL IL) NIL)
    (T (CONS (LIST (CAR IL) (LIST (CAR DL)))
    (DEFPD2 (CDR IL) (CDR DL)))))))
))))))))))))))))))))))))))
DEFANDEXP((
(TRN (LAMBDA (X) (COND
    ((H X IDENT) X)
    ((H X CONSTANT) X)
    ((H X QSTOKEN) (TRANSTRING (P X STRING QSTOKEN)))
    (T (PROG (Y Z) (SETQ Y (CAR X)) (SETQ Z (CDR X)) (RETURN (COND
    ((EQUAL Y 1) (TRN1 (TRN (CAR Z))))
```

```
      ((EQUAL Y 2) (TRN2 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 3) (TRN3 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 4) (TRN4 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 5) (TRN5 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 6) (TRN6 (TRN (CAR Z)) (TRN (CADR Z)) (TRN (CADDR Z))))
      ((EQUAL Y 7) (TRANSLAMBDA (TRNP (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 8) (TRN8 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 9) (TRANSEQEXP (TRNL Z)))
      ((EQUAL Y 10) (TRN10 (TRNL Z)))
      ((EQUAL Y 11) (TRN11 (TRN (CAR Z))))
      ((EQUAL Y 12) (TRN12 (TRN (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 13) (TRN13 (TRNI (CAR Z)) (TRN (CADR Z))))
      ((EQUAL Y 14) (TRN14 (TRNI (CAR Z)) (TRNP (CADR Z)) (TRN (CADDR Z))
          (TRN (CAR (CDDDR Z)))))
      ((EQUAL Y 15) (TRANSDECL (TRNP (CAR Z)) (TRN (CADR Z))
          (TRN (CADDR Z))))
      ((EQUAL Y 16) (TRN16 (TRN (CAR Z))))
      (T (ERRM 13 X)))))))))
(TRNL (LAMBDA (X) (COND  ((NULL X) NIL)
      (T (CONS (TRN (CAR X)) (TRNL (CDR X)))))))
(TRNI (LAMBDA (X) (COND ((H X IDENT) X) (T (ERRM 14 X)))))
(TRNP (LAMBDA (X) (COND ((H X IDENT) X)
      ((EQUAL (CAR X) 1) (TRNP (CADR X)))
      ((EQUAL (CAR X) 9) (M SEQPFORM (TRNPL (CDR X))))
      ((EQUAL (CAR X) 11) (TRNP (CADR X)))
      ((EQUAL (CAR X) 16) (TRNP (CADR X)))
      (T (ERRM 15 X)))))
(TRNPL (LAMBDA (X) (COND ((NULL X) NIL)
      (T (CONS (TRNP (CAR X)) (TRNPL (CDR X)))))))
(MKPDCON (LAMBDA (X) (M CONSTANT (GETVALPREDEF (M IDENT X)))))
(MKCOERCE (LAMBDA (X) (M FUNCTDES (MKPDCON (QUOTE COERCE)) X)))
(TRN1 (LAMBDA (X) X))
(TRN2 (LAMBDA (X Y) (M FUNCTDES (MKCOERCE X) Y)))
(TRN3 (LAMBDA (X Y) (M FUNCTDES (MKPDCON (QUOTE EQUAL))
      (TRANSEQEXP (LIST X Y)))))
(TRN4 (LAMBDA (X Y) (M CONDEXP (MKCOERCE X) (MKCOERCE Y)
      (MKPDCON (QUOTE FALSE)))))
(TRN5 (LAMBDA (X Y) (M CONDEXP (MKCOERCE X) (MKPDCON (QUOTE TRUE))
      (MKCOERCE Y))))
(TRN6 (LAMBDA (X Y Z) (M CONDEXP (MKCOERCE X) Y Z)))
(TRN8 (LAMBDA (X Y) (M FUNCTDES (MKPDCON (QUOTE SET))
      (TRANSEQEXP (LIST X Y)))))
(TRN10 (LAMBDA (X) (M CASEEXP (MKCOERCE (CAR X)) (CDR X))))
(TRN11 (LAMBDA (X) (M BLOCK NIL NIL (LIST X))))
(TRN12 (LAMBDA (X Y) (M BLOCK NIL (P Y LDECLPART BLOCK)
      (CONS X (P Y BODY BLOCK)))))
(TRN13 (LAMBDA (X Y) (M BLOCK NIL (CONS (M LDECL X (P Y BODY BLOCK))
      (P Y LDECLPART BLOCK)) (P Y BODY BLOCK))))
(TRN14 (LAMBDA (X Y Z W) (M BLOCK (CONS (M RDECL X (TRANSLAMBDA Y Z))
      (P W RDECLPART BLOCK)) (P W LDECLPART BLOCK) (P W BODY BLOCK))))
(TRN16 (LAMBDA (X) (COND
      ((AND (NULL (P X RDECLPART BLOCK)) (NULL (P X LDECLPART BLOCK))
          (EQUAL (LENGTH (P X BODY BLOCK)) 1))
          (CAR (P X BODY BLOCK)))
      (T X))))
(TRANSTRING (LAMBDA (X) (COND
      ((EQUAL (LENGTH X) 1) (M CONSTANT (CAR X))) (T (PROG (I)
      (SETQ I (CRIDENT))
      (RETURN (M LAMBDAEXP I (M CASEEXP (MKCOERCE I) (TRNSTR1 X)))))))))
(TRNSTR1 (LAMBDA (X) (COND ((NULL X) NIL)
      (T (CONS (M CONSTANT (CAR X)) (TRNSTR1 (CDR X)))))))
(CRIDENT (LAMBDA () (M IDENT (GENSYM1 (QUOTE $$$INT.$)))))
```

```
(TRANSDECL (LAMBDA (X E B) (M FUNCTDES (TRANSLAMBDA X B) E)))
(TRANSLAMBDA (LAMBDA (X B) (COND ((H X IDENT) (M LAMBDAEXP X B))
    (T (PROG (I) (SETQ I (CRIDENT)) (RETURN (M LAMBDAEXP I
        (TRLAMB1 1 I (P X BODY SEQPFORM) B)))))))))
(TRLAMB1 (LAMBDA (K I X B) (COND
    ((NULL X) B)
    (T (TRANSDECL (CAR X) (M FUNCTDES (MKCOERCE I)
        (M CONSTANT K)) (TRLAMB1 (ADD1 K) I (CDR X) B))))))
(TRANSEQEXP (LAMBDA (X) (PROG (S)
    (SETQ S (TRSE1 X)) (RETURN (TRSE2 X S S (CRIDENT))))))
(TRSE1 (LAMBDA (X) (COND ((NULL X) NIL)
    (T (CONS (CRIDENT) (TRSE1 (CDR X)))))))
(TRSE2 (LAMBDA (X S S1 I) (COND
    ((NULL X) (M LAMBDAEXP I (M CASEEXP (MKCOERCE I) S1)))
    (T (TRANSDECL (CAR S) (CAR X) (TRSE2 (CDR X) (CDR S) S1 I))))))
))))))))))))))))))))))))))
DEFANDEXP((
(GETVAL (LAMBDA (X E) (GETVAL1 X E NIL)))
(GETVAL1 (LAMBDA (X E EM) (COND
    ((NULL E) (COND ((EQ (P X STRING IDENT) (QUOTE LIBRARY))
        (PROG () (SETQ LIBSIGNAL T) (RETURN (M ERRORDEN))))
        (T (GETVALPREDEF X))))
    ((H (CAR E) ENVMARK) (GETVAL1 X (CDR E) E))
    ((IDEQUAL X (P (CAR E) LEFT ENVPAIR))
        ((LAMBDA (V) (COND
            ((H V RECFUNCTDEN)
                (M FUNCTDEN (P V CONTROL RECFUNCTDEN) EM))
            ((H V RECLABELDEN)
                (M PROGLABELDEN (P V BODY RECLABELDEN) EM
                    (P V DUMP RECLABELDEN)))
            (T V))) (P (CAR E) RIGHT ENVPAIR)))
    (T (GETVAL1 X (CDR E) EM)))))
(IDEQUAL (LAMBDA (X Y) (EQ (P X STRING IDENT) (P Y STRING IDENT))))
(INTERPRET (LAMBDA (PR IS) (INTERPRET2
    (CONS PR (P IS CONTROL STATE))
    (P IS STACK STATE)
    (P IS ENVR STATE)
    (P IS DUMP STATE)
    (P IS MEMORY STATE)
    (P IS ATOMCOUNT STATE)
    F
    (GETVALPREDEF (M IDENT (QUOTE COERCE)))
    )))
(INTERPRET2 (LAMBDA (CN ST EN DM MR AC LIBSIGNAL COERCECHK)
    (PROG (X S1 T1 S2 T2)
    LOOP
    (COND (LIBSIGNAL (RETURN (M STATE CN (CDR ST) EN DM MR AC))))
    (COND ((NOT (NULL CN)) (GO NEXTINST)))
    (COND ((NOT (NULL DM)) (GO REPLENISH)))
    (RETURN (CAR ST))
    REPLENISH
    (SETQ CN (P (CAR DM) CONTROL DUMPELEM))
    (SETQ ST (CONS (CAR ST) (P (CAR DM) STACK DUMPELEM)))
    (SETQ EN (P (CAR DM) ENVR DUMPELEM))
    (SETQ DM (CDR DM))
    (GO LOOP)
    NEXTINST
    (SETQ X (CAR CN)) (SETQ CN (CDR CN))
    (SETQ S1 (M ERRORDEN)) (SETQ T1 NIL) (SETQ S2 S1) (SETQ T2 NIL)
    (COND ((NULL ST) (GO SKIP)))
    (SETQ S1 (CAR ST)) (SETQ T1 (CDR ST))
    (COND ((NULL T1) (GO SKIP)))
```

```
      (SETQ S2 (CAR T1)) (SETQ T2 (CDR T1))
      SKIP
      (COND
  ((H X CONSTANT)
      (SETQ ST (CONS (P X VALUE CONSTANT) ST)))
  ((H X IDENT)
      (SETQ ST (CONS (GETVAL X EN) ST)))
  ((H X FUNCTDES)
      (SETQ CN (APPEND (LIST (P X FUNCTPART FUNCTDES)
          (P X ARGPART FUNCTDES) (M APPLY)) CN)))
  ((H X LAMBDAEXP)
      (SETQ ST (CONS (M FUNCTDEN (LIST (M BIND (P X PARAMPART LAMBDAEXP))
          (P X BODY LAMBDAEXP)) EN) ST)))
  ((H X CONDEXP)
      (SETQ CN (APPEND (LIST (P X PREMISS CONDEXP) (M BRANCH
           (P X CONCLUSION CONDEXP) (P X ALTERNATIVE CONDEXP))) CN)))
  ((H X CASEEXP)
      (SETQ CN (APPEND (LIST (P X INDEX CASEEXP) (M SELECT
          (P X BODY CASEEXP))) CN)))
  ((H X BLOCK)
      (PROG () (SETQ DM (CONS (M DUMPELEM CN ST EN) DM)) (SETQ ST NIL)
          (SETQ CN (APPEND (P X RDECLPART BLOCK) (APPEND
          (P X LDECLPART BLOCK) (LIST (M MARKENV) (M EXEC
          (P X BODY BLOCK)))))))))
  ((H X RDECL)
      (SETQ EN (CONS (M ENVPAIR (P X LEFT RDECL) (M RECFUNCTDEN (LIST
          (M BIND (P (P X RIGHT RDECL) PARAMPART LAMBDAEXP))
          (P (P X RIGHT RDECL) BODY LAMBDAEXP)))) EN)))
  ((H X LDECL)
      (SETQ EN (CONS (M ENVPAIR (P X LEFT LDECL) (M RECLABELDEN
          (P X RIGHT LDECL) DM)) EN)))
  ((H X EXEC)
      (PROG (B) (SETQ B (P X BODY EXEC))
          (SETQ CN (COND ((NULL (CDR B)) (CONS (CAR B) CN))
          (T (APPEND (LIST (CAR B) (M DELETE) (M EXEC (CDR B))) CN))))))
  ((H X BRANCH)
      (COND ((H S1 BOOLDEN) (PROG () (SETQ CN (CONS (COND ((H S1 TRUEDEN)
          (P X CONCLUSION BRANCH)) (T (P X ALTERNATIVE BRANCH))) CN))
          (SETQ ST T1))) (T (ERRM 1 S1)))))
  ((H X SELECT)
      (COND ((AND (H S1 INTCLASS) (NOT (GREATERP 1 S1))
          (NOT (GREATERP S1 (LENGTH (P X BODY SELECT)))))
          (PROG () (SETQ CN (CONS (ELEM (P X BODY SELECT) S1) CN))
          (SETQ ST T1)))
          ((H S1 LLDEN) (SETQ ST (CONS 1 T1)))
          ((H S1 ULDEN) (SETQ ST (CONS (LENGTH (P X BODY SELECT)) T1)))
          (T (ERRM 2 S1))))
  ((H X BIND)
      (PROG () (SETQ EN (CONS (M ENVPAIR (P X BODY BIND) S1) EN))
          (SETQ ST T1)))
  ((H X APPLY)
      (COND ((AND (EQ S2 COERCECHK) (NOT (H S1 REFDEN)))
          (SETQ ST (CONS S1 T2)))
          ((H S2 FUNCTDEN) (PROG () (SETQ DM (CONS
          (M DUMPELEM CN T2 EN) DM)) (SETQ CN (P S2 CONTROL FUNCTDEN))
          (SETQ EN (P S2 ENVR FUNCTDEN)) (SETQ ST (LIST S1))))
          (T (ERRM 3 S2))))
  ((H X MARKENV)
      (SETQ EN (CONS (M ENVMARK) EN)))
  ((H X DELETE)
      (SETQ ST T1))
  (T (PROG (Q) (SETQ Q (P X STRING BASICFUNCTINST)) (COND
```

```
((AND (OR (EQQ Q SET) (EQQ Q NCSET)) (H S2 REFDEN))
    (COND ((H S2 EXPREFDEN)
        (PROG () (REPLACE (P S2 ADDRESS EXPREFDEN) S1 MR)
        (SETQ ST (CONS S1 T2))))
    (T (PROG () (SETQ CN (APPEND (LIST (M APPLY) (M DELETE)) CN))
        (SETQ ST (APPEND (LIST S1 (P S2 SETF IMPREFDEN) S1) T2))))))
((AND (EQQ Q VAL) (H S1 REFDEN))
    (COND ((H S1 EXPREFDEN)
        (SETQ ST (CONS (ELEM MR (P S1 ADDRESS EXPREFDEN)) T1)))
    (T (PROG () (SETQ CN (CONS (M APPLY) CN)) (SETQ ST (APPEND
        (LIST (M ERRORDEN) (P S1 VALF IMPREFDEN)) T1))))))
((EQQ Q COERCE)
    (COND ((H S1 REFDEN)(SETQ CN (APPEND (LIST (M BASICFUNCTINST
        (QUOTE VAL)) (M BASICFUNCTINST (QUOTE COERCE))) CN))) (T ())))
((AND (EQQ Q GOTO) (H S1 LABELDEN))
    (COND ((H S1 PROGLABELDEN) (PROG () (SETQ CN (LIST (M EXEC
        (P S1 BODY PROGLABELDEN)))) (SETQ ST ()) (SETQ EN (P S1 ENVR
        PROGLABELDEN)) (SETQ DM (P S1 DUMP PROGLABELDEN))))
    (T (ERRM 4 S1))))
(T (SETQ ST (INTERPRET1 Q ST S1 T1 S2 T2)))))))
(GO LOOP))))
(INTERPRET1 (LAMBDA (Q ST S1 T1 S2 T2) (COND
((EQQ Q ATOM)
    (PROG () (SETQ AC (ADD1 AC))
        (RETURN (CONS (M PROGATOMDEN AC) ST))))
((OR (EQQ Q REF) (EQQ Q NCREF))
    (PROG () (SETQ MR (APPEND1 MR S1))
        (RETURN (CONS (M EXPREFDEN (LENGTH MR)) T1))))
((EQQ Q ISINTEGER)  (CONS (MBOOL (H S1 INTCLASS )) T1))
((EQQ Q ISBOOLEAN)  (CONS (MBOOL (H S1 BOOLDEN  )) T1))
((EQQ Q ISCHAR)     (CONS (MBOOL (H S1 CHARCLASS)) T1))
((EQQ Q ISATOM)     (CONS (MBOOL (H S1 ATOMDEN  )) T1))
((EQQ Q ISFUNCTION) (CONS (MBOOL (H S1 FUNCTDEN )) T1))
((EQQ Q ISREF)      (CONS (MBOOL (H S1 REFDEN   )) T1))
((EQQ Q ISLABEL)    (CONS (MBOOL (H S1 LABELDEN )) T1))
((AND (EQQ Q IMPREF) (H S2 FUNCTDEN) (H S1 FUNCTDEN))
    (CONS (M IMPREFDEN S2 S1) T2))
((OR (EQQ Q EQUAL) (EQQ Q NCEQUAL)) (CONS (MBOOL (OR
    (AND (H S2 INTCLASS) (H S1 INTCLASS) (EQUAL S2 S1))
    (AND (H S2 TRUEDEN) (H S1 TRUEDEN))
    (AND (H S2 FALSEDEN) (H S1 FALSEDEN))
    (AND (H S2 CHARCLASS) (H S1 CHARCLASS) (EQUAL S2 S1))
    (AND (H S2 LLDEN) (H S1 LLDEN))
    (AND (H S2 ULDEN) (H S1 ULDEN))
    (AND (H S2 PROGATOMDEN) (H S1 PROGATOMDEN)
        (EQUAL (P S2 NAME PROGATOMDEN) (P S1 NAME PROGATOMDEN)))
    (AND (H S2 EXPREFDEN) (H S1 EXPREFDEN)
        (EQUAL (P S2 ADDRESS EXPREFDEN) (P S1 ADDRESS EXPREFDEN)))
    )) T2 ))
((AND (EQQ Q CHARGREATER) (H S2 CHARCLASS) (H S1 CHARCLASS))
    (CONS (MBOOL (NOT (ORDERP S1 S2))) T2))
((EQQ Q READCHAR) (CONS (READCH NIL) ST))
((AND (EQQ Q WRITECHAR) (H S1 CHARCLASS))
    (PROG () (PRIN1 S1) (RETURN ST)))
((AND (EQQ Q NOT) (H S1 BOOLDEN))
    (CONS (MBOOL (H S1 FALSEDEN)) T1))
((EQQ Q PRINT) (PROG () (PRINT S1) (RETURN ST)))
((EQQ Q TERPRI) (PROG () (TERPRI) (RETURN (CONS (M ERRORDEN) ST))))
(T (PROG (ZZ) (OVON) (SETQ ZZ (COND
((AND (EQQ Q GREATER) (H S2 INTCLASS) (H S1 INTCLASS))
    (CONS (MBOOL (GREATERP S2 S1)) T2))
((AND (EQQ Q INC) (H S1 INTCLASS)) (CONS (ADD1 S1) T1))
```

```
((AND (EQQ Q DEC) (H S1 INTCLASS)) (CONS (SUB1 S1) T1))
((AND (EQQ Q NEG)       (H S1 INTCLASS))
    (CONS (MINUS S1) T1))
((AND (EQQ Q ADD)       (H S2 INTCLASS) (H S1 INTCLASS))
    (CONS (PLUS S2 S1) T2))
((AND (EQQ Q SUBTRACT)  (H S2 INTCLASS) (H S1 INTCLASS))
    (CONS (DIFFERENCE S2 S1) T2))
((AND (EQQ Q MULTIPLY)  (H S2 INTCLASS) (H S1 INTCLASS))
    (CONS (TIMES S2 S1) T2))
((AND (EQQ Q DIVIDE)    (H S2 INTCLASS) (H S1 INTCLASS)
    (NOT (ZEROP S1)))
    (CONS (QUOTIENT S2 S1) T2))
((AND (EQQ Q REMAINDER) (H S2 INTCLASS) (H S1 INTCLASS)
    (NOT (ZEROP S1)))
    (CONS (REMAINDER S2 S1) T2))
(T (ERRM 5 (LIST Q S2 S1)))
)) (OVOFF) (RETURN ZZ))))))
(ELEM (LAMBDA (L I) (COND ((NULL L) (ERRM 6 I))
    ((EQUAL I 1) (CAR L)) (T (ELEM (CDR L) (SUB1 I))))))
(REPLACE (LAMBDA (I V L) (COND ((NULL L) (ERRM 7 I))
    ((EQUAL I 1) (RPLACA L V)) (T (REPLACE (SUB1 I) V (CDR L))))))
))))))))))))))))))))))))))))
DEFPREDEF ()
DEFINE((
(GEDANKEN (LAMBDA () (PROG (X) (OVOFF)
    (SETQ X (PARSE))
    (COND (PARSEPRINTSW (PRINTGED (QUOTE (RESULT OF PARSE)) X)))
    (SETQ X (TRN X))
    (COND (ABTRNPRINTSW (PRINTGED (QUOTE (ABSTRACT TRANSLATION)) X)))
    (TERPRI)
    (SETQ X (INTERPRET X INITIALSTATE))
    (TERPRI)
    (COND ((NOT (EQ (CAR X) (QUOTE STATE))) (RETURN X)))
    (CSETQ INITIALSTATE X)
    (RETURN (QUOTE (INITIALSTATE HAS BEEN UPDATED))))))
(PRINTGED (LAMBDA (X Y) (PROG ()
    (TERPRI) (PRINT X) (TERPRI) (PRINT Y))))
))))))))))))))))))))))))))))
CSET (INITIALSTATE (STATE NIL NIL NIL NIL NIL 0))
CSET (PARSEPRINTSW NIL)
CSET (ABTRNPRINTSW NIL)
VERBOS (NIL)
PRBUFFER (T)
GEDANKEN ()
UNITSEQ IS #X #I (CASE I OF X);
INTTODIGIT IS #X
    (CASE INC X OF "0", "1", "2", "3", "4", "5", "6", "7", "8", "9");
DIGITTOINT IS #X (X IS COERCE X;
IF X = "0" THEN 0 ELSE
IF X = "1" THEN 1 ELSE
IF X = "2" THEN 2 ELSE
IF X = "3" THEN 3 ELSE
IF X = "4" THEN 4 ELSE
IF X = "5" THEN 5 ELSE
IF X = "6" THEN 6 ELSE
IF X = "7" THEN 7 ELSE
IF X = "8" THEN 8 ELSE
IF X = "9" THEN 9 ELSE
GOTO ERROR);
VECTOR ISR #(L,U,F) (L IS COERCE L; U IS COERCE U; F IS COERCE F;
    IF GREATER(L,U) THEN #I (I IS COERCE I;
        IF I = LL THEN L ELSE IF I = UL THEN DEC L ELSE GOTO ERROR)
```

```
    ELSE (V IS VECTOR(L, DEC U, F); T IS F U; #I (I IS COERCE I;
        IF I = UL THEN U ELSE IF I = U THEN T ELSE V I)));
LIBRARY $
(LAMBDA () INITIALSTATE) ()
OPEN (GEDANK SYSFILE OUTPUT)
CHKPOINT (GEDANK)
CLOSE (GEDANK)
GEDANKEN ()
X IS 100; ALP1 IS 128; LIBRARY $
DEFINE (((TS GEDANKEN)))
TS() 3$
TS() 512$
TS() X$
TS() ALP1$
TS() INC 1$
TS() NEG INC 1$
TS() #X NEG INC X$
TS() (# X NEG INC X)1$
TS() (# P IF P THEN X ELSE ALP1) TRUE$
TS() (# P IF P THEN X ELSE ALP1)FALSE$
TS() (CASE 1 OF 4,5,6)$
TS() (CASE 2OF 4,5,6)$
TS() (CASE 3 OF 4,5,6)$
TS() (CASE LL OF 4,5,6)$
TS() (CASE UL OF 4,5,6)$
TS() (F ISR # X F;F)$
TS() (F ISR # X F; F 1)$
TS() (F ISR#X F;(F 1)1)$
TS() ""$
TS() "A"$
TS() "AB"$
TS() "#,=:(); |"$
TS() TRUE$
TS() FALSE$
TS() UL$
TS() LL$
TS() ERROR$
TS() QUOTECHAR$
TS() DOLLARCHAR$
TS() ATOM$
TS() VAL$
TS() DEC$
TS() NCEQUAL$
TS() SET$
TS() EQUAL$
TS() ATOM()$
TS() 1 = 1$
TS() TRUE AND FALSE$
TS() TRUE OR FALSE$
TS() $
TS() 1,2$
TS() 1,2,X,2$
TS() (X IS 3; X IS INC X; NEG X)$
TS() (X,Y IS 3,4; X,Y IS Y,X; ADD(X,ADD(Y,X)))$
TS() #(X,Y) SUBTRACT(X,Y)$
TS() #()MULTIPLY(7,10)$
TS() IS 7; DIVIDE(9,2)$
TS() ((R) IS 7; REMAINDER(R,3))$
TS() (X IS ((9,2),(),(7,3)); ((X,Y),(),(Z,W)) IS X;
        ADD(DIVIDE(X,Y),REMAINDER(Z,W)))$
TS() (((3)))$
TS() (X IS REF 3; IF X = 3 THEN X:=4 ELSE SET(X,5); VAL X)$
```

```
TS() (X,Y IS NCREF REF 3, REF REF 4; IF ISREF VAL X AND NOT ISREF VAL Y
        THEN VAL X := Y ELSE (); PRINT VAL X; PRINT VAL Y;
        NCSET(X,REF 10);
        PRINT COERCE X; PRINT COERCE Y; PRINT X; COERCE 17)$
TS() (X IS REF 3; Y IS REF 3; NCEQUAL(X,Y))$
TS() (X IS REF 3; Y IS X; NCEQUAL(X,Y))$
TS() CASE 1 OF 6$
TS()  GREATER(6,5)$
TS()  GREATER(6,6)$
TS()  GREATER(6,7)$
TS() (S IS REF 0;
      SM IS IMPREF(#X S := ADD(S,X), #() S := DEC S);
      PRINT VAL SM;
      PRINT (SM := 3);
      PRINT VAL SM;
      PRINT VAL SM)$
TS() (WRITECHAR "A"; WRITECHAR "B"; TERPRI(); WRITECHAR "C")$
CSET (PARSEPRINTSW T)
CSET (ABTRNPRINTSW T)
TS() READCHAR()$X
TS() )))))))$
TS() SILLYS
TS()  CHARGREATER("A","A")$
TS()  CHARGREATER("A","B")$
TS()  CHARGREATER("B","A")$
CSET (PARSEPRINTSW NIL)
CSET (ABTRNPRINTSW NIL)
TS()  (N IS REF 1; S IS REF 0;
      L: IF GREATER(N,2) THEN GOTO D ELSE ();
      S := ADD(S,MULTIPLY(N,N)); N:= INC N; GOTO L;
      D: VAL S)$
/*
```