235 LISP INTERPRETER

THERE NØW EXISTS A VERSIØN ØF LISP 1.5 ØN THE DARTMØUTH TIME-SHARING SYSTEM. IT RESIDES UNDER SYSTEM NAME "EVAL." EVAL IS A MØDIFICATIØN ØF LISP 1.5 DEVELØPED FØR THE IBM 7090 AT M.I.T. IT IS SIMILAR ENØUGH TØ THE M.I.T. VERSIØN THAT AN ADEQUATE REFERENCE MANUAL IS:

LISP 1.5 PRØGRAMMERS MANUAL
BY J. MCCARTHY ET AL.
M.I.T. PRESS

HØWEVER THE USER ØF EVAL MUST KEEP IN MIND THE FØLLØWING DIFFERENCES BETWEEN THE SYSTEM AT DARTMØUTH AND THE SYSTEM DESCRIBED IN THE MANUAL.

(1)THE LISP INTERPRETER AT DARTMØUTH IS EVAL INSTEAD ØF THE MØRE USUAL EVALQUØTE. THIS MEANS THE USER MUST MAKE HIS PRØGRAM CØNSIST ØF FØRMS FØR EVAL INSTEAD ØF DØUBLETS FØR EVALQUØTE.
FØR EXAMPLE:
CAR((A B C)) WØULD BECØME (CAR(QUØTE(A B C)))
CØNS(A B) WØULD BECØME (CØNS(QUØTE A)(QUØTE B))

HØWEVER THE EXECUTIØN ØF (INTERPRETQ T) WILL CAUSE THE INTERPRETER TØ BECØME EVALQUØTE. INTERPRETQ(()) WILL BRING BACK EVAL.

(2) THE RESTRICTIØN AGAINST HAVING AN ATØM WITH A PRINT NAME ØF MØRE THAN 30 CHARACTERS HAS BEEN REMØVED. THE ØNLY PRESENT RESTRICTIØN IS HØW MUCH STØRAGE THE USER IS WILLING TØ GIVE FØR THE SAVING ØF THE PRINT NAME.

(3) THE SYMBØLS FØR TRUE AND FALSE ARE T AND NIL RESPECTIVELY. THERE IS NØ *T* ØR F IN THE DARTMØUTH SYSTEM. BØTH T AND NIL ARE CØNSTANTS THAT EVALUATE INTØ THEMSELVES.

(4) THE FØRM ØF NUMBERS ØN INPUT AND ØUTPUT HAS BEEN CHANGED SØME. ØN INPUT A STRING ØF DIGITS PRECEEDED BY A MINUS SIGN ØR AN ØPTIØNAL PLUS SIGN THAT CØNTAINS NØ DECIMAL PØINT (.) IS READ AS AN INTEGER IN THE CURRENT INPUT BASE. (FØR EXAMPLE 123 -456). A SIMILAR STRING ØF DIGITS BUT FØLLØWED IMEDIATELY BY ØNLY A DECIMAL PØINT IS TAKEN TØ BE AN INTEGER IN BASE TEN. (FØR EXAMPLE 356. -794.). FLØATING PØINT NUMBERS TAKE THE

SAME FØRM AS DESCRIBED IN THE MANUAL. IT IS STILL INCØRRECT TØ HAVE A NUMBER WITH A DECIMAL PØINT AS THE FIRST CHARACTER AS THE DECIMAL PØINT WILL BE INTERPRETED AS PART ØF LISP DØT NØTATIØN. IF THERE IS AN "E" IN A FLØATING PØINT NUMBER THE DECIMAL PØINT MAY BE DRØPPED. IF IT IS IT IS ASSUMBED TØ BE BEFØRE THE "E". TØ SET THE INPUT AND ØUTPUT BASE ØF NUMBERS TWØ NEW FUNCTIØNS WERE ADDED--SETIBASE AND SETØBASE. SETIBASE ØNLY EFFECTS THE INPUT ØF NUMBERS THAT DØ NØT CØNTAIN A DECIMAL PØINT ØR AN "E". SETØBASE DØES NØT EFFECT THE ØUTPUT ØF FLØATING PØINTING PØINT NUMBERS.

CØNTINUED IN LISP-2***

(5) CØND HAS BEEN REDEFINED TØ ACCEPT DIFFERENT FØRMS ØF ARGUMENTS THAN THE PREVIØUS DEFINITIØN. THE EXPANDED DEFINITIØN DØES STILL HØWEVER ACCEPT EXPRESSIØNS CØDED IN THE ØLD MANNER EXCEPT THAT IT RETURNS NIL INSTEAD ØF GIVING AN ERRØR IF NØNE ØF THE PRØPØSITIØNS ARE TRUE. FØR A DESCRIPTIØN ØF THE NEW ADDITØNS TØ CØND SEE CØND--***.

(6) THE RESTRICTIØN THAT GØ AND RETURN MAY ØNLY APPEAR ØN THE TØP LEVEL ØF A PRØG HAS BEEN REMØVED. INSTEAD THEY MAY BE USED ANYTIME AFTER A PRØG HAS BEEN ENTERED. WHEN THEY ARE EVALUATED GØ HAS AS ITS VALUE THE PØINT TØ BE TRANSFERED TØ AND RETURN HAS AS ITS VALUE THE RETURN VALUE ØF THE PRØG. THE EXECUTIØN ØF GØ AND RETURN IS INHIBITED UNTIL THE INTERPRETER RETURNS TØ THE TØP LEVEL ØF THE CURRENT PRØG AT WHICH TIME PRØG THEN TRANSFERS TØ THE PØINT SPECIFIED BY GØ ØR RETURN. GØ AND RETURN MAY ØNLY REFER TØ THE LAST PRØG ENTERED BY THE INTERPRETER; THEY CANNØT REFER TØ PRØGS ENTERED PREVIØUSLY TØ THE CURRENT ØNE.

(7) THE SECTIØN ØF THE MANUAL DESCRIBING ØVERLØRD SHØULD BE IGNØRED. INSTEAD ØNE SETS UP HIS PRØGRAM AS A CØLLECTIØN ØF FØRMS. EACH ØF THESE FØRMS WILL BE EVALUATED AND ITS RESULT PRINTED. REACHING THE END ØF A PRØGRAM WILL CAUSE AN END ØF FILE ERRØR (ERRØR R4). THE READ RØUTINE CØMPLETELY IGNØRES CARRIAGE RETURNS AND LINE NUMBERS SØ THAT A FØRM MAY RUN FØR SEVERAL LINES. THERE EXISTS A SPECIAL FUNCTIØN CALLED LISTEN THAT CAUSES EVAL TØ INTERPRET FRØM THE TELETYPE INSTEAD ØF THE CURRENT PRØGRAM.

(8) LISP'S ERRØR MESSAGE'S ARE DESCRIBED IN LSPERR***.

(9) LIST STRUCTURE IS STØRED IN THE 235 DIFFERENTLY THAN ØN THE 7090. EACH WØRD ØF LIST STRUCTURE TAKES TWØ WØRDS ØF CØRE MEMØRY IN THE 235. THE EVEN LØCATIØN WØRD CØNTAINS THE CAR HALF AND THE NEXT HIGHER ØDD LØCATIØN WØRD CØNTAINS THE CDR HALF. ALL LIST PØINTERS PØINT TØ THE EVEN HALF ØF THE WØRD. FØR THIS REASØN. THE ATØM HEAD (THE CAR ØF AN ATØM) IN THE 235 IS AN ØDD PØINTER INSTEAD ØF THE -1 THAT IS USED ØN THE 7090. BECAUSE THERE IS NØ TAG PØRTIØN IN THE 235 NUMBERS ARE REPRESENTED AS ATØMS WITH THE PRØPERTY FIXNUM ØR FLØNUM INSTEAD ØF PNAME AS WITH REGULAR ATØMS. THE PRØPERTY AFTER THE INDICATØR FIXNUM IS A PØINTER TØ AN INTEGER

AND THE PRØPERTY AFTER THE INDICATØR FLØNUM IS A PØINTER TØ A FLØATING PØINT NUMBER. TØ ACCØMODATE PNAMES AND NUMBERS FULL-WØRD SPACE ALSØ CØNSISTS ØF DØUBLE WØRDS--AN EVEN LØCATIØN AND THE NEXT HIGHER ØDD LØCATIØN.

(10) THERE EXISTS IN THE LIBRARY FØR THØSE WHØ ARE INTERESTED A CØPY ØF THE WANG ALGØRITHM FØR PRØPØSITIØNAL CALCULUS UNDER WANG--***.

(11) WHEN THE PRØGRAMMER MAKES A CALL TØ READ, THE FUNCTIØN DØES NØT READ THE FIRST LIST AFTER STØP BUT INSTEAD READS THE NEXT LIST THAT HAS NØT BEEN READ BY THE INTERPRETER.

(12) A LIST AND DESCRIPTIØN ØF THØSE FUNCTIØNS IN THE DARTMØUTH LISP SYSTEM CAN BE FØUND IN LSPFNC***.

(13) THE INTERPRETER IS ACTUALLY AS DESCRIBED IN APPENDIX B ØF THE MANUAL EXCEPT THAT EVA LQUØTE MUST BE ASKED FØR SPECIALLY AND EVAL IS THE NØRMAL INTERPRETER. ALSØ THERE IS NØ SPECIAL CHECK FØR NIL IN EVAL ØR APPLY. THIS MEANS THAT (APPLY ()ARGS ALIST) CAUSES AN ERRØR MESSAGE SAYING NIL IS AN UNDEFINED FUNCTIØN.

(14) THERE ARE NØ PRØVISIØNS FØR LAP ØR A LISP CØMPILER IN DARTMØUTH LISP.

(15) THE LISP READ RØUTINE DØES NØT THE RECØGNIZE THE $$ CØNVENTIØN ØF THE 7090 FØR QUØTING CHARACTERS IN PRINT NAMES. INSTEAD ØNE USES THE SLASH (/). THE NEXT CHARACTER FØLLØWING A SLASH IS PLACED IN THE PNAME REGARDLESS ØF WHAT IT IS. CHARACTERS THAT MUST BE SLASHED ARE <SPACE>,<CØMMA>,<DØT>,<SLASH>, <RIGHT PARENTHESIS>,<LEFT PARENTHESIS>,<CARRIAGE RETURN>, AND LEADING <PLUS SIGN>,<MINUS SIGN>, AND <DIGITS> TØ PREVENT CØNFUSIØN BETWEEN ATØMS AND NUMBERS.

(16) SINCE LISP IS A RAPIDLY CHANGING SYSTEM EACH CHANGE WILL CARRY WITH IT A NUMBER WHICH INDICATES WHICH VERSIØN IN CURRENTLY IN USE. THIS NUMBER IS PRINTED AT THE BEGINNING ØF THE RUN IN THE FØRM ØF " LISP XX" WHERE XX IS THE UPDATE NUMBER. IN ADDITIØN IF THERE IS AN E FØLLØWING THE NUMBER THIS INDICATES THE CURRENT VERSIØN IS EXPERIMENTAL AND MAY CØNTAIN PRØGRAMMING ERRØRS. WITH EACH CHANGE IN VERSIØN A NØTATIØN WILL BE MADE IN THE BEGINNING ØF LSPUPD*** INDICATING THE ADDITIØNS ØVER PREVIØUS VERSIØNS. IN ØRDER TØ HELP IN MAKING THESE CHANGES AND ADDITIØNS THE USER IS ENCØURAGED TØ APPEND QUESTIØNS AND CØMMENTS TØ FILE " LISP??" UNDER USER NUMBER 407886.

THIS IS A LIST ØF THE ATØMS THAT ARE INITIALLY DEFINED IN THE DARTMØUTH VERSIØN ØF LISP. A $ <DØLLAR SIGN> INDICATES A CHANGE FRØM THE "LISP 1.5 PRØGRAMMERS MANUAL."

LAMBDA

QUØTE [FSUBR]

CØNS [SUBR]

EQ [SUBR]

LIST [FSUBR]

CØND [FSUBR] $
CØND HAS BEEN MØDIFIED TØ ACCEPT AN EXPANDED FØRM ØF PRØPØSITIØNS. SEE CØND--*** FØR DETAILS.

CDR; CAR; CDDR; CDAR; CADR; CAAR; CDDDR; CDDAR; CDADR; CDAAR; CADDR; CADAR; CAADR; CAAAR; [SUBR]

RPLACA; RPLACD; [SUBR] $
A CHECK IS MADE TØ SEE IF THE USER IS MØDIFYING A LØCATIØN THAT IS NØT IN FREE STØRAGE ØR FULL WØRD SPACE. IF SØ, THE ØPERATIØN IS NØT DØNE AND AN ERRØR IS GIVEN. SINCE INITIAL DEFINITIØNS ARE NØT IN THESE AREAS THE USER MAY NØT MØDIFY THEM.

READ [SUBR] $
READ NØW READS THE NEXT UNREAD LIST FRØM THE DISK. IT IS INITIALLY SET TØ READ FRØM YØUR CURRENT PRØGRAM AREA.

PRINT [SUBR]

PRIN1 [SUBR] $
PRIN1 MAY NØW BE USED TØ PRINT LISTS AS WELL AS ATØMS. IF GIVEN A LIST IT PRINTS IT NØRMALLY BUT DØES NØT TERMINATE THE PRINT LINE.

TERPRI [SUBR]

PRØG [FSUBR]

RETURN [SUBR] $
RETURN MAY NØW BE GIVEN ANYTIME AFTER A PRØG HAS BEEN ENTERED. IF IT IS NØT GIVEN ØN THE TØP LEVEL ØF A PRØG ITS ACTIØN IS INHIBITED UNTIL CØNTRØL RETURNS TØ THE PRØG.

GØ [FSUBR] $
GØ MAY BE USED ANYTIME AFTER A PRØG HAS BEEN ENTERED. IF IT IS NØT GIVEN ØN THE TØP LEVEL ØF A PRØG ITS ACTIØN IS INHIBITED UNTIL CØNTRØL RETURNS TØ THE PRØG.

GØ* [SUBR] $
GØ* IS IDENTICAL TØ GØ EXCEPT THAT IT EVALUATES ITS ARGUMENT. IT MAY BE USED FØR A SWITCH-TYPE TRANSFER.

ADVANCE [SUBR] $
ADVANCE READS THE NEXT CHARACTER FRØM THE DISK. IT IGNØRES LINE NUMBERS. ADVANCE SHØULD BE USED CAREFULLY AS CHARACTERS IT READS WILL NØT BE READ BY THE READ FUNCTIØN.

ADVANCE* [SUBR] $
ADVANCE* IS IDENTICAL TØ ADVANCE EXCEPT THAT IT DØES NØT IGNØRE LINE NUMBERS.

READTTY [SUBR] $
READTTY IS A FUNCTIØN ØF NØ ARGUMENTS. ITS VALUE IS ØNE LIST TYPED IN FRØM THE USERS TELETYPE. IT CAUSES A CALL FØR INPUT AND CØNTINUES TØ GIVE ØNE UNTIL A CØMPLETE LIST IS TYPED. THIS MEANS THAT IF PARENTHESES DØ NØT BALANCE ØN ØNE LINE THE CØMPUTER WILL CØNTINUE TØ ASK FØR INPUT UNTIL THEY DØ. AS WITH READ, READTTY ALSØ IGNØRES <CARRIAGE RETURN>S. THIS MEANS THAT A SINGLE ATØM MUST HAVE A SPACE TYPED AFTER IT. ØN ANY CALL FØR INPUT THERE MAY BE GIVEN MØRE THAN ØNE LIST; IF SØ THE EXTRA LISTS (ØR PART ØF A LIST) WILL BE USED ØN THE NEXT USE ØF READTTY.

ADVANCETTY [SUBR] $
ADVANCETTY READS THE NEXT CHARACTER FRØM THE USERS TELETYPE. IF THE LAST CHARACTER READ WAS A <CARRIAGE RETURN> THEN A CALL ØF ADVANCETTY CAUSES A CALL FØR INPUT AND IT RETURNS THE FIRST CHARACTER TYPED. THE VERY FIRST CHARACTER READ BY ADVANCETTY WILL BE A <CARRIAGE RETURN>.

FIXNUM $
THE PRØPERTY ØF A FIXED PØINT NUMBER.

FLØNUM $
THE PRØPERTY ØF A FLØATING PØINT NUMBER.

PACK [SUBR]

CLEARBUFF [SUBR]

MAKNUM [SUBR] $
MAKNUM IS A FUNCTIØN ØF TWØ ARGUMENTS. IT CREATES AN ATØM WITH A PRØPERTY LIST ØF MAKNUM'S SECØND ARGUMENT FØLLØWED BY MAKNUM'S FIRST ARGUMENT. MAKNUM IS ØFTEN USED AS AN EASY WAY TØ CREATE NUMBERS FRØM FULL WØRDS.

NUMØB [SUBR]

PLUS [FSUBR]

TIMES [FSUBR]

CONTINUED IN LSPFND***

FIX [SUBR]

FIXP [SUBR]

FLØATP [SUBR]

NUMBERP [SUBR]

DIFFERENCE [SUBR]

QUØTIENT [SUBR]

REMAINDER [SUBR]

DIVIDE [SUBR]

MINUSP [SUBR]

LESSP [SUBR].

GREATERP [SUBR]

SETIBASE [SUBR] $
SETIBASE IS A FUNCTIØN ØF ØNE ARGUMENT. ITS EFFECT IS TØ CHANGE THE BASE ØF INTEGERS ØN INPUT. SETIBASE ØNLY EFFECTS THØSE INTEGERS THAT DØ NØT HAVE A DECIMAL PØINT AS THE LAST CHARACTER. SETIBASE HAS NØ EFFECT ØN FLØATING PØINT NUMBERS. THE VALUE ØF SETIBASE IS THE PREVIØUS INPUT BASE. THIS ALLØWS RØUTINES TØ CHANGE THE BASE AND RESTØRE IT. THE INPUT BASE IS INITIALLY SET TØ TØ TEN.

SETØBASE [SUBR] $
SETØBASE IS A FUNCTIØN ØF ØNE ARGUMENT. ITS EFFECT IS TØ CHANGE THE ØUTPUT BASE ØF INTEGERS. SETØBASE RETURNS THE PREVIØUS ØUTPUT BASE. THE ØUTPUT BASE IS INITIALLY SET TØ TEN.

PRØG2 [SUBR]

SET;SETQ; [FSUBR]

GENSYM [SUBR]

ERSETQ [FSUBR] $
ERSETQ IS A FUNCTIØN ØF ØNE ARGUEMNT. IT'S VALUE IS THE EVALUATIØN ØF ITS ARGUMENT CØNSED WITH NIL IF NØ ERRØR ØCCURS. IF AN ERRØR ØCCURS THEN ERSETQ RETURNS NIL. ERSETQ ALLØWS THE USER TØ TRY ØNE APPRØACH TØ A SØLUTIØN AND IF IT

DØES NØT WØRK, TØ BACK ØFF AND TRY ANØTHER PATH.

ERRØR [SUBR]

EQUAL [SUBR]

ØR; AND [FSUBR]

NØT [SUBR]

PUTPRØP [SUBR] $
PUTPRØP IS DARTMØUTH'S VERSIØN ØF LISP'S WAY ØF DEFINING THINGS. IT IS A FUNCTIØN ØF THREE ARGUMENTS. THE FIRST ØF THESE MUST BE AN ATØM AND IT IS THIS ATØM THAT IS BEING DEFINED. THE SECØND ARGUMENT IS THE DEFINITIØN AND THE THIRD ARGUMENT IS THE DEFINITIØN TYPE (USUALLY EXPR ØF FEXPR). ANY ØTHER ØCCURENCE ØF THE PRØPERTY TYPE ØN THE DEFINED ATØM IS REMØVED.

DEFPRØP [FSUBR] $
DEFPRØP IS IDENTICAL TØ PUTPRØP EXCEPT THAT DEFPRØP QUØTES ITS THREE ARGUMENTS. THIS ALLØWS THE USER TØ GIVE ATØMS PRØPERTIES WITHØUT TYPING QUØTE ALL THE TIME.

CSET [SUBR]

CSETQ [FSUBR]

PAIR [SUBR]

FUNARG

LABEL

T [APVAL] $
T REPLACES *T* IN THE LISP MANUAL. T IS A CØNSTANT THAT EVALUATES INTØ ITSELF. PREDICATES IN LISP RETURN T ØR NIL.

NIL [APVAL]

GET [SUBR]

SASSØC [SUBR]

NCØNC [SUBR]

EVAL [SUBR]

APPLY [SUBR]
APPLY NØ LØNGER MAKES A SPECIAL CHECK FØR NIL AS A FUNCTIØN NAME. INSTEAD (APPLY NIL ARGS ALIST) WILL CAUSE AN ERRØR STATING THAT NIL IS AN UNDEFINED FUNCTIØN.

EVLIS [SUBR]

INTERN [SUBR]

LISTEN [SUBR] $
LISTEN IS A FUNCTIØN ØF USE FØR TIME-SHARING LISP. ØN ENTERING A LISTEN THE INTERPRETER STØPS EVALUATING FRØM THE DISK AND INSTEAD EVALUATES STATEMENTS FRØM THE TELETYPE. LISTEN IS TERMINATED BY TYPING THE ATØM "STØP". CARE MUST BE TAKEN TØ INSURE THAT STØP IS FØLLØWED BY A SPACE TØ PREVENT THE TIME-SHARING SYSTEM FRØM THINKING THE INPUT IS A CØMMAND TØ STØP LISP. WHEN LISTEN IS TERMINATED IT RETURNS NIL. THE TYPICAL LISP PRØGRAM CØNSISTS ØF A SET ØF FUNCTIØN DEFINITIØNS FØLLØWED BY A LISTEN.

STØP [SUBR] $
STØP AS AN ATØM IS USED TØ TERMINATE A LISTEN. STØP AS A FUNCTIØN IS USED TØ GIVE A TERMINAL EXIT FØR LISP.

RECLAIM [SUBR]

PACKØ [SUBR] $
PACKØ IS A FUNCTIØN ØF ØNE ARGUMENT WHICH IS A PØINTER TØ A CHARACTER. ITS EFFECT IS TØ ADD THIS CHARACTER TØ THE TELETYPE ØUTPUT BUFFER. PACKØ RETURNS NIL.

BELL [APVAL] $
BELL IS FØR THØSE WHØ LIKE A LITTLE AUDIØ ØUTPUT FRØM THEIR PRØGRAM. BELL IS A CØNSTANT THAT EVALUATES INTØ AN ATØM WITH THE PRINT NAME <BELL>.

EXPR; FEXPR; SUBR; FSUBR; APVAL; PNAME;

FWCØNS [SUBR] $
FWCØNS IS A FUNCTIØN ØF ØNE ARGUMENT. IT'S ARGUMENT IS A PØINTER TØ A FULL WØRD. FWCØNS'S VALUE IS A PØINTER TØ A NEW FULL WØRD WHICH IS A CØPY ØF THE ARGUMENT.

ØBLIST [APVAL] 4
AS ØN THE 7090 THE ØBLIST EVALUATES INTØ ALL ATØMIC ØBJECTS THAT HAVE BEEN INPUTED INTØ THE SYSTEM. THE ØBLIST ØN THE 235 HAS ØNLY 32 BUCKETS INSTEAD ØF THE USUAL 64 BUT IS ØTHERWISE SIMILAR.

ØPEN [SUBR] $
ØPEN IS A FUNCTIØN ØF ØNE ØR TWØ ARGUMENTS. ITS PURPØSE IS TØ TØ DETERMINE WHICH FILE READ, ADVANCE, AND ADVANCE* PULL THEIR DATA FRØM. INITIALLY THE FILE THAT IS ØPEN FØR READING IS THE USERS CURRENT PRØGRAM. HØWEVER IF THE USER WISHES TØ READ ELSEWHERE ØN THE DISK HE JUST SPECIFIES THE USER NUMBER AND THE FILE NAME TØ ØPEN. THE FIRST ARGUMENT ØF ØPEN IS THE USER NUMBER AND THE SECØND IS THE FILE NAME. FØR EXAMPLE: TØ READ IN TRACE- FRØM THE LIBRARY, (ØPEN(QUØTE LIBEVA)(QUØTE TRACE-)). IT SHØULD BE

NOTED THAT THE USER NUMBER AND THE FILE NAME ARE BØTH ATØMS AND NEVER NUMBERS. THIS MEANS THAT AN USER NUMBER WITH ALL NUMERIC CHARACTERS MUST HAVE THE FIRST DIGIT QUØTED WITH A <SLASH> IN ØRDER TØ MAKE THE NUMBER LØØK LIKE AN ATØM. ALSØ ANY FILE NAME WITH LESS THAN SIX CHARACTERS MUST BE FØLLØWED BY TRAILING BLANKS.

CØNTINUED IN LSPFNE***

LISP

LSPFNE 21:22 SEPT. 7,1966

```
100(DEFPRØP SQR(LAMBDA(N)(PRØG(K X FLAG)(SETQ N(QUØTIENT(ADD1
110(SETQ X(PLUS N(SETQ FLAG 0.0))))2))LØØP(CØND((GREATERP FLAG 12)
120(RETURN N)))(SETQ N(QUØTIENT(PLUS N(QUØTIENT X N))2))(SETQ FLAG(ADD1
130 FLAG))(GØ LØØP)))EXPR)
140(LISTEN)
```

ØPEN [SUBR] $ (CØNTINUED)
FØR EXAMPLE: TØ ØPEN FILE "ABCD" UNDER USER NUMBER 123456, (ØPEN(QUØTE /123456)(QUØTE ABCD/ /)). ØNCE A FILE IS ØPENED READING CØMMENCES WITH THE BEGINNING ØF THE FILE WITH THE USE ØF A FILE READING CØMMAND (READ, ADVANCE, ADVANCE*). THE FIRST CHARACTER READ BY ADVANCE ØR ADVANCE* WILL BE A <CARRIAGE RETURN>. IF INSTEAD ØF GIVING ØPEN A USER NUMBER AND A FILE NAME THE USER GIVES ØNLY THE SINGLE ARGUMENT NIL, THEN READING CØMMENCES WITH THE BEGINNING ØF THE USERS CURRENT PRØGRAM. THUS (ØPEN()) ACTS AS A RESTØRE.

INTERPRETQ [SUBR] $
INTERPRETQ IS A FUNCTIØN DESIGNED TØ GIVE EVAL BETTER CØMPATIBILITY WITH EVALQUØTE SYSTEMS. INTERPRETQ TAKES ØNE ARGUMENT. IF THIS ARGUMENT IS NIL THEN THE INTERPRETIVE FUNCTIØN IS EVAL. HØWEVER IF THE ARGUMENT IS NØT NIL THEN THE INTERPRETIVE FUNCTIØN BECØMES EVALQUØTE. THUS (INTERPRETQ T) WILL CALL EVALQUØTE AND INTERPRETQ(NIL) WILL BRING BACK EVAL. INTERPRETQ AFFECTS BØTH THE NØRMAL INTERPRETING FRØM THE PRØGRAM AND THE INTERPRETING DØNE ØN STATEMENTS TYPED IN FRØM A TELETYPE BY MEANS ØF A LISTEN. THE VALUE ØF INTERPRETQ IS ITS PREVIØUS ARGUMENT. THIS IS INITIALY NIL SØ THAT INTERPRETING STARTS WITH EVAL.

FUNCTIØN [FSUBR]

MEMBER [SUBR]

APPEND [SUBR]

SUBST [SUBR]

MAPLIST [SUBR] $
MAPLIST'S DEFNITIØN HAS BEEN CHANGED TØ:
```
(MAPLIST(LAMBDA(F,L)(CØND
   ((NULL L)NIL)
   (T(CØNS(F L)(MAPLIST F(CDR L))))  ))EXPR)
```
THIS MEANS THAT THE ØRDER ØF MAPLISTS ARGUMENTS HAS BEEN CHANGED FRØM THE ØRDER DESCRIBED IN THE LISP 1.5 PRØGRAMMERS MANUAL.

MAPCAR; MAP; MAPCØN [SUBR] $
LIKE MAPLIST, MAPCAR AND MAP AND MAPCØN HAVE HAD THEIR ØRDER ØF ARGUMENTS REVERSED FRØM THEIR ØRIGINAL DEFINITIØNS.

CØPY [SUBR]

REVERSE [SUBR]

LENGTH [SUBR]

CLØCK [SUBR] $
CLØCK IS A FUNCTIØN ØF NØ ARGUMENTS. ITS VALUE IS THE NUMERICAL TIME ØF DAY IN SIXTHS ØF A SECØND.

LØGØR; LØGAND; LØGXØR [FSUBR] $
THESE FUNCTIØNS WØRK NØRMALLY EXCEPT THAT THEY ACCEPT FLØATING PØINT ARGUMENTS WITHØUT ERRØR. IF GIVEN A FLØATING PØINT ARGUMENT THEY DØ NØT FIX IT BUT USE ITS PATTERN ØF BITS DIRECTLY TØ FØRM THEIR RESULT. THE VALUE ØF THESE FUNCTIØNS IS ALWAYS A FIXED PØINT NUMBER.

ADD1; SUB1 [SUBR]

ZERØP; ØNEP [SUBR]

LEFTSHIFT; RIGHTSHIFT [SUBR] $
THE SECØND ARGUMENT ØF LEFTSHIFT ØR RIGHTSHIFT (WHICH IS THE SHIFT CØUNT) MUST BE A FIXED PØINT NUMBER. THESE FUNCTIØNS SHIFT THE NUMBER SPECIFIED AS THE FIRST ARGUMENT (FLØATING PØINT ARGUMENT IS HANDLED THE SAME AS IN LØGØR) THE SPECIFIED NUMBER ØF BITS. THE EFFECT ØN FIXED PØINT NUMBERS IS MULTIPLYING ØR DIVIDING BY THE APPRØPIATE PØWER ØF TWØ. THE FØRMS (LEFTSHIFT A B) AND (RIGHTSHIFT A(MINUS B)) ARE IDENTICAL EXCEPT IN THE CASE WHERE B IS ZERØ. IN THIS CASE DUE TØ THE NATURE ØF THE GE 235 A LEFTSHIFT 0 CAUSES A SHIFT ØF THE SIGN BIT ØF THE LØWER HALF INTØ THE UPPER HALF WHILE A RIGHTSHIFT 0 CAUSES A SHIFT ØF THE SIGN BIT FRØM THE UPPER HALF INTØ THE LØWER HALF. IN FIXED PØINT ARITHMETIC THE SIGN IN THE LØWER HALF IS IGNØRED AND IS FØRCED TØ AGREE WITH THE SIGN ØF THE UPPER HALF. IN FLØATING PØINT ARITHMENTIC THE SIGN ØF THE LØWER HALF IS THE SIGN ØF THE NUMBER AND SIGN ØF THE UPPER HALF IS THE SIGN ØF THE EXPØNENT.

MAX; MIN [FSUBR]

RECIP [SUBR]

ATTRIB [SUBR]

REMPRØP [SUBR]

PRØP [SUBR]

CØNC [SUBR]

DEFINE

```
100 (DEFPRØP DEFLIST
110          (LAMBDA(L A)(MAPLIST
120             (FUNCTIØN(LAMBDA(X)(PUTPRØP(CAAR X)(CADAR X)(CADR L))))
130             (CAR L)  ))FEXPR)
140 (DEFLIST(
150   (DEFINE(LAMBDA(L A)(EVAL
160         (LIST(QUØTE DEFLIST)(CAR L)(QUØTE EXPR))
170         A  )))   )FEXPR)
180 (LISTEN)
```

CØND--

```
90 .USE EDIT RUNØFF TØ LIST
100  CØND HAS BEEN MØDIFIED TØ ACCEPT EXPRESSIØNS ØF THE FØRM:
110
115 .RIG 66
120  (CØND(P11,P12,...,PIK)(P21,P22,...,P2L)...(PN1,PN2,...,PNM))
125 .RIG 60
130
140 CØND EVALUATES ALL PI1 UNTIL IT FINDS THE FIRST THAT IS NØT NIL.
150 CØND THEN EVALUATES THE REMAINDER ØF PIJ, FØR ALL J, RETURNING
160 THE LAST PIJ FØR ITS VALUE.  IF ALL PI1 ARE NIL, THEN THE VALUE ØF
170 CØND IS NIL.  UNDER THIS DEFNITIØN IT IS PØSSIBLE TØ HAVE A
180 CØNDITIØNAL ØF ØNE EXPRESSIØN IN THE CASE ØF THERE BEING ØNLY
190 ØNE PIJ.  IN THIS CASE IF PI1 IS NØT NIL, THEN THE VALUE ØF PI1 IS
200 THE VALUE ØF CØND.  A LISP DEFNITIØN ØF CØND (HERE CALLED CØND*) IS:
210
220   (DEFPRØP CØND*
230     (LAMBDA(L A)(CØND
240       ((NULL L)NIL)
250       (T(CØND1(EVAL(CAAR L)A)L A))
260     ))      FEXPR)
270
230   (DEFPRØP CØND1
290     (LAMBDA(E L A)(CØND
300       ((NULL E)(CØND
310          ((NULL(CDR L))NIL)
320          (T(CØND1(EVAL(CAADR L)A)(CDR L)A))
330       ))
340       (T(CØND2 E(CDAR L)A))
350     ))      EXPR)
360
370   (DEFPRØP CØND2
```

COND-- CONTINUED

```
380     (LAMBDA(E L A)(COND
390       ((NULL L)E)
400       (T(COND2(EVAL(CAR L)A)(CDR L)A))
410     ))      EXPR)
```

BOOLE-

```
100 (DEFPROP NOT NULL EXPR)
110 (DEFPROP AND
120           (LAMBDA(L A)(COND((NULL L)T)
130                           ((EVAL(CAR L)A)
140                            (EVAL(CONS(QUOTE AND)(CDR L))A))
150                           (T())   ))
160           FEXPR)
170 (DEFPROP OR
180           (LAMBDA(L A)(COND((NULL L)())
190                           ((EVAL(CAR L)A)T)
200                           (T(EVAL(CONS(QUOTE OR)(CDR L))A))   ))
210           FEXPR)
```

SETQ--

```
100 (DEFPROP SET
110           (LAMBDA(L A)(CDR(RPLACD(SASSOC(EVAL(CAR L)A)
120                                          A
130                                          (FUNCTION(LAMBDA()
140                                             (ERROR(QUOTE SET))   ))  )
150                                    (EVAL(CADR L)A)  )))
160           FEXPR)
170 (DEFPROP SETQ
180           (LAMBDA(L A)(CDR(RPLACD(SASSOC(CAR L)
190                                          A
200                                          (FUNCTION(LAMBDA()
210                                             (ERROR(QUOTE SETQ))   ))  )
220                                    (EVAL(CADR L)A)   )))
230           FEXPR)
```

WANG--

```
100 (DEFPRØP THEØREM(LAMBDA(S)(TH1 NIL NIL(CADR S)(CADDR S)))EXPR)
110 (DEFPRØP TH1(LAMBDA(A1 A2 A C)(CØND((NULL A)(TH2 A1 A2 NIL NIL C))
120 (T(ØR(MEMBER(CAR A)C)(CØND((ATØM(CAR A))(TH1(CØND((MEMBER(CAR A)
130 A1)A1)(T(CØNS(CAR A)A1)))A2(CDR A)C))(T(TH1 A1(CØND((MEMBER(CAR A)
140 A2)A2)(T(CØNS(CAR A)A2)))(CDR A)C)))))))EXPR)
150 (DEFPRØP TH2(LAMBDA(A1 A2 C1 C2 C)(CØND((NULL C)(TH A1 A2 C1 C2))
160 ((ATØM(CAR C))(TH2 A1 A2(CØND((MEMBER(CAR C)C1)C1)(T(CØNS(CAR C)
170 C1)))C2(CDR C)))(T(TH2 A1 A2 C1(CØND((MEMBER(CAR C)C2)C2)(T(CØNS
180 (CAR C)C2)))(CDR C)))))EXPR)
190 (DEFPRØP TH(LAMBDA(A1 A2 C1 C2)(CØND((NULL A2)(AND(NØT(NULL C2))
200 (THR(CAR C2)A1 A2 C1(CDR C2))))(T(THL(CAR A2)A1(CDR A2)C1 C2))))EXPR)
210 (DEFPRØP THL(LAMBDA(U A1 A2 C1 C2)(CØND((EQ(CAR U)(QUØTE NØT))
220 (TH1R(CADR U)A1 A2 C1 C2))((EQ(CAR U)(QUØTE AND))(TH2L(CDR U)
230 A1 A2 C1 C2))((EQ(CAR U)(QUØTE ØR))(AND(TH1L(CADR U)A1 A2 C1 C2)
240 (TH1L(CADDR U)A1 A2 C1 C2)))((EQ(CAR U)(QUØTE IMPLIES))(AND(TH1L
250 (CADDR U)A1 A2 C1 C2)(TH1R(CADR U)A1 A2 C1 C2)))((EQ(CAR U)
260 (QUØTE EQUIV))(AND(TH2L(CDR U)A1 A2 C1 C2)(TH2R(CDR U)A1 A2 C1 C2)))
270 (T(ERRØR(LIST(QUØTE THL)U A1 A2 C1 C2)))))EXPR)
280 (DEFPRØP THR(LAMBDA(U A1 A2 C1 C2)(CØND((EQ(CAR U)(QUØTE NØT))
290 (TH1L(CADR U)A1 A2 C1 C2))((EQ(CAR U)(QUØTE AND))(AND(TH1R(CADR U)
300 A1 A2 C1 C2)(TH1R(CADDR U)A1 A2 C1 C2)))((EQ(CAR U)(QUØTE ØR))
310 (TH2R(CDR U)A1 A2 C1 C2))((EQ(CAR U)(QUØTE IMPLIES))(TH11(CADR U)
320 (CADDR U)A1 A2 C1 C2))((EQ(CAR U)(QUØTE EQUIV))(AND(TH11(CADR U)
330 (CADDR U)A1 A2 C1 C2)(TH11(CADDR U)(CADR U)A1 A2 C1 C2)))(T(ERRØR
340 (LIST(QUØTE THR)U A1 A2 C1 C2)))))EXPR)
350 (DEFPRØP TH1L(LAMBDA(V A1 A2 C1 C2)(CØND((ATØM V)(ØR(MEMBER V C1)
360 (TH(CØNS V A1)A2 C1 C2)))(T(ØR(MEMBER V C2)(TH A1(CØNS V A2)C1 C2)))
370 ))EXPR)
380 (DEFPRØP TH1R(LAMBDA(V A1 A2 C1 C2)(CØND((ATØM V)(ØR(MEMBER V A1)
390 (TH A1 A2(CØNS V C1)C2)))(T(ØR(MEMBER V A2)(TH A1 A2 C1(CØNS V C2))))
400 ))EXPR)
410 (DEFPRØP TH2L(LAMBDA(V A1 A2 C1 C2)(CØND((ATØM(CAR V))(ØR(MEMBER
420 (CAR V)C1)(TH1L(CADR V)(CØNS(CAR V)A1)A2 C1 C2)))(T(ØR(MEMBER
430 (CAR V)C2)(TH1L(CADR V)A1(CØNS(CAR V)A2)C1 C2)))))EXPR)
440 (DEFPRØP TH2R(LAMBDA(V A1 A2 C1 C2)(CØND((ATØM(CAR V))(ØR(MEMBER
450 (CAR V)A1)(TH1R(CADR V)A1 A2(CØNS(CAR V)C1)C2)))(T(ØR(MEMBER(CAR V)
460 A2)(TH1R(CADR V)A1 A2 C1(CØNS(CAR V)C2)))))EXPR)
470 (DEFPRØP TH11(LAMBDA(V1 V2 A1 A2 C1 C2)(CØND((ATØM V1)(ØR(MEMBER
480 V1 C1)(TH1R V2(CØNS V1 A1)A2 C1 C2)))(T(ØR(MEMBER V1 C2)(TH1R
490 V2 A1(CØNS V1 A2)C1 C2)))))EXPR)
500 (PRØG (A)A(SETQ A(ERSETQ(READTTY)))(CØND((NULL A)(GØ A)))
510 (SETQ A(ERSETQ(THEØREM(CAR A))))(CØND((NULL A)(GØ A)))
520 (CØND((CAR A)(PRINT(QUØTE VALID)))(T(PRINT(QUØTE INVALID))))
530 (GØ A))
```

BREAK-

```
100 (DEFPRØP BREAK(LAMBDA(FN WHEN WHAT)(PRØG(TYPE DEF)
110  (CØND((SETQ DEF(GET FN(QUØTE EXPR)))(SETQ TYPE(QUØTE EXPR)))
120    ((SETQ DEF(GET FN(QUØTE FEXPR)))(SETQ TYPE(QUØTE FEXPR)))
130    ((PUTPRØP FN(LIST(QUØTE LAMBDA)(QUØTE(L A))(LIST(QUØTE BREAK1)
140 NIL T(SETQ DEF(LIST FN(QUØTE UNDEFINED)))WHAT))(QUØTE FEXPR))
150     (RETURN DEF)))
160  (CØND((EQ(CAR(CADDR DEF))(QUØTE BREAK1))(RETURN(CØNS FN(QUØTE
170 (ALREADY BRØKEN))))))
180  (PUTPRØP FN(LIST(QUØTE LAMBDA)(CADR DEF)(LIST(QUØTE BREAK1)
190 (CADDR DEF)WHEN(LIST FN)WHAT))TYPE)
200  (RETURN FN) ))EXPR)
210 (DEFPRØP BREAK1(LAMBDA(L A)(PRØG(*X)
220  (CØND((NULL(SETQ *X(EVAL(CADR L)A)))(RETURN(EVAL(CAR L)A)))
230   ((NULL(EQUAL *X(QUØTE(NIL))))(GØ BO)))
240  (PRINT(APPEND(QUØTE(CRACK IN))(CADDR L)))
250  (CØND((NULL(CAR(CDDDR L)))NIL)
260   (T(PRINT(EVAL(CAR(CDDDR L))A))))
270  (GØ B3)
280  BO(PRINT(APPEND(QUØTE(BREAK IN))(CADDR L)))
290  (CØND((NULL(CAR(CDDDR L)))NIL)
300  (T(PRINT(EVAL(CAR(CADDDR L))A))))
310  B1(CØND((NULL(SETQ *X(ERSETQ(READTTY))))(GØ BO))
320   ((EQ(SETQ *X(CAR *X))(QUØTE QUIT))(ERRØR(CADDR L)))
330   ((EQ *X(QUØTE STØP))(GØ B3))
340   ((EQ *X(QUØTE RETURN))(GØ B2))
350   ((EQ *X(QUØTE EVAL)))
360   ((AND(SETQ *X(ERSETQ(EVAL *X A)))
370    (ERSETQ(PRINT(CAR *X))) )(GØ B1))
380   (T(GØ BO)) )
390  (CØND((NULL(SETQ *X(ERSETQ(EVAL(CAR L)A))))(GØ BO)))
400  (PRINT(CØNS(CAR(CADDR L))(QUØTE(EVALUATED))))
410  (SETQ A(CØNS(CØNS(CAR(CADDR L))(CAR *X))A))
420  (GØ B1)
430  B2(CØND((ØR(NULL(SETQ *X(ERSETQ(READTTY))))
440                (NULL(SETQ *X(ERSETQ(EVAL(CAR *X)A)) )))(GØ BO)))
450  (GØ B4)
460  B3(CØND((EQ(CAAR A)(CAR(CADDR L)))(SETQ *X(LIST(CDAR A))))
470   ((NULL(SETQ *X(ERSETQ(EVAL(CAR L)A))))(GØ BO)))
480  B4(PRINT(APPEND(QUØTE(VALUE ØF))(CADDR L)))
490  (CØND((NULL(ERSETQ(PRINT(CAR *X))))(PRINT(QUØTE ØK))))
500  (RETURN(CAR *X)) ))FEXPR)
510 (DEFPRØP UNBREAK(LAMBDA(FN)(PRØG(TYPE DEF)
520  (CØND((SETQ DEF(GET FN(QUØTE EXPR)))(SETQ TYPE(QUØTE EXPR)))
530   ((SETQ DEF(GET FN(QUØTE FEXPR)))(SETQ TYPE(QUØTE FEXPR)))
540   (T(RETURN(CØNS FN(QUØTE(NØT BRØKEN))))) )
550  (CØND((EQ(CAR(CADDR DEF))(QUØTE BREAK1))(RETURN
560    (PUTPRØP FN(LIST(QUØTE LAMBDA)(CADR DEF)(CADR(CADDR DEF)))TYPE))))
570  (RETURN(CØNS FN(QUØTE(NØT BRØKEN)))) ))EXPR)
580 (DEFPRØP BREAKLIST(LAMBDA(L A)(MAPCAR
590  (QUØTE(LAMBDA(X)(BREAK X T NIL)))L))FEXPR)
```

BREAK- CONTINUED

```
600 (DEFPROP UNBREAKLIST(LAMBDA(L A)(MAPCAR(QUOTE UNBREAK)L))FEXPR)
999 (LISTEN)
```

LISP02

```
100 (DEFPROP UNION
110          (LAMBDA(X Y)(COND((NULL X)Y)
120                           ((MEMBER(CAR X)Y)(UNION(CDR X)Y))
130                           (T(CONS(CAR X)(UNION(CDR X)Y)))   ))
140          EXPR)
150 (DEFPROP INTERSECTION
160          (LAMBDA(X Y)(COND((NULL X)())
170                           ((MEMBER(CAR X)Y)
180                            (CONS(CAR X)(INTERSECTION(CDR X)Y)))
190                           (T(INTERSECTION(CDR X)Y))   ))
200          EXPR)
210 (LISTEN)
```

REMOB -

```
100 (DEFPROP REMOB(LAMBDA(L A)(PROG(B C)A(COND((NULL L)(RETURN())))
110 (SETQ B(CAR L))(SETQ C OBLIST)B(COND((NULL C)())((MEMBER B(CAR C))
120 (PROG()(COND((EQ(CAAR C)B)(RETURN(RPLACA C(CDAR C))))(SETQ C(CAR C))
130 A(COND((EQ(CADR C)B)(RPLACD C(CDDR C)))((SETQ C(CDR C))(GO A)))))
140 ((SETQ C(CDR C))(GO B)))(SETQ L(CDR L))(GO A)))FEXPR)
150 (DEFPROP RENAME(LAMBDA(L A)(PROG(B)(EVAL(LIST(QUOTE REMOB)(CAR L))NIL
160 (SETQ B OBLIST)A(COND((NULL B)(RETURN()))((MEMBER(CADR L)(CAR B))
170 (GO B)))(SETQ B(CDR B))(GO A)B(SETQ B(REDUCE(CADR L)(CAR B)))
180 (RPLACA B(CAR L))(RPLACA(CDR(REDUCE(QUOTE PNAME)(CDAR L)))(GET
190 (CADR L)(QUOTE PNAME)))(RETURN(CADR L))))FEXPR)
200 (DEFPROP REDUCE(LAMBDA(A L)(PROG()B(COND((NULL)(RETURN())))
210 ((EQUAL(CAR L)A)(RETURN L)))(SETQ L(CDR L))(GO B)))EXPR)
220 (LISTEN)
```

MKNAM-

```
50 (PRØG()(CSETQ BRK(LIST
60 (CDDR 12.)(CDDR 31.)(CDDR 48.)(CDDR 60.)
70 ))
90 (CSETQ FILLS 274877644799.)
100 (DEFPRØP MKNAM(LAMBDA()(PRØG(PNAME WØRD CHAR CNT FLG)
110 A(SETQ WØRD 0)  (SETQ CNT 12.)  (SETQ FLG T)
120 B(CØND((SETQ CHAR(CHAR)))
130  (T(RETURN(CØND((EQUAL WØRD 0)(INTERN PNAME))
140    (T(INTERN(NCØNC PNAME(LIST(CDDR(LØGXØR FILLS WØRD))))))) ))) )
145 (SETQ CHAR(LØGXØR CHAR 63.))
150 (CØND(FLG(SETQ CHAR(LEFTSHIFT CHAR 19.))))
160 (SETQ WØRD(LØGØR WØRD(LEFTSHIFT CHAR CNT)))
170 (CØND((MINUSP(SETQ CNT(DIFFERENCE CNT 6.)))
180  (CØND(FLG(SETQ FLG NIL)(SETQ CNT 12))
190    (T(SETQ PNAME(NCØNC PNAME(LIST(CDDR(LØGXØR FILLS WØRD)))))(GØ A)))
200 (GØ B) ))EXPR)
210 (DEFPRØP CHAR(LAMBDA()(CØND
220  ((MEM(CSETQ LCHAR(ADVANCETTY))BRK)NIL)
230  (T(MAKNUM LCHAR(QUØTE FIXNUM))))))EXPR)
240 (DEFPRØP MEM(LAMBDA(A B)(CØND
250 ((NULL B)NIL)
260 ((EQ(CDR A)(CDAR B)))(T(MEM A(CDR B)))))EXPR)
9999)(LISTEN)
```

LSPUPD

```
10  THE FØLLØWING IS A LIST ØF CHANGES ADDED TØ NEW LISP UPDATES.
20 THE CHANGES ARE LISTED IN REVERSE ØRDER.
30  IN ØRDER TØ DIRECT NEW ADDITIØNS AND CHANGES THE USER IS
40 ENCØURAGED TØ APPEND QUESTIØNS AND CØMMENTS ØNTØ FILE "LISP??"
50 UNDER USER NUMBER 407886.
60
70 CURRENT VERSIØN:  LISP 47
80  1) NUMØB NØW RØUNDS ØN FLØATING PØINT CØNVERSIØN SØ THAT (FIX 1.0)
90 NØW RETURNS 1 INSTEAD ØF 0.
100  2) ØN ØUTPUT USING DØT NØTATIØN, THE DØT IS SURØUNDED BY A PAIR
110 ØF BLANKS.
120  3) BECAUSE ØF ABUSE ØF ITS INTENT ØPEN HAS BEEN UNDEFINED.
130  4) NEW FUNCTIØNS:  ATTRIB, CØNC, PRØP, REMPRØP, MAPCØN (PLEASE
140 NØTE THAT LIKE MAPLIST AND MAPCAR THE ARGUMENTS ØF MAPCØN HAVE BEEN
150 REVERSED.)
160  5) THE NAME ØF THE GARBAGE CØLLECTØR FUNCTIØN HAS BEEN CHANGED TØ
170 RECLAIM AS STATED IN THE "LISP 1.5 PRØGRAMMERS MANUAL."
```

LSPUPD CONTINUED

180 6) THE SIZE OF THE OUTPUT BUFFER HAS BEEN DECREASED CAUSING MORE
190 SWAPS WITH PROGRAMS THAT HAVE ALOT OF OUTPUT BUT GIVING MORE FREE
200 STORAGE.
210
220 LISP 46
230 1) A PREVIOUS BUG IN NUMBERP HAS BEEN FIXED
240 2) THE FORM OF NUMBERS HAS BEEN REDEFINED SO THAT NUMBERS LOOK
250 LIKE ATOMS WHOSE FIRST PROPERTY IS FIXNUM OR FLONUM BUT INSTEAD
260 OF HAVING A FURTHER PROPERTY LIST AFTER THE NUMBER PROPERTY
270 IT POINTS DIRECTLY TO THE NUMBER. IN OTHER WORDS (CDDR NUMBER) IS
280 A POINTER INTO FULL-WORD SPACE TO A CELL THAT CONTAINS THE VALUE
290 OF THE NUMBER.
300 3) STORAGE HAS BEEN REALLOCATED SO THAT THERE IS LESS FULL-WORD
310 SPACE AND LESS PUSH-DOWN LIST BUT MORE FREE-STORAGE.
320 4) NEW PROTECTION FEATURES HAVE BEEN ADDED SO THAT IT IS MORE
330 DIFFICULT (MAYBE EVEN IMPOSSIBLE) FOR THE USER TO DAMAGE TIME-
340 SHARING.
350 5) THE REPRESENTATION OF CONSTANTS ON PROPERTY LISTS HAS BEEN
360 CHANGED. NO LONGER ARE CONSTANTS DEPRESSED A LEVEL IN THE LIST
370 STRUCTURE AFTER THE APVAL PROPERTY FLAG. THIS MEANS THAT CSET
380 IS DEFINED AS (PROG2(PUTPROP ARG1 ARG2(QUOTE APVAL))ARG2) INSTEAD
390 OF (PROG2(PUTPROP ARG1(LIST ARG2)(QUOTE APVAL))ARG2) AS PRESIOUSLY.
400 6) THE SIZE OF BUFFO HAS BE CHANGED FROM 81 CHARACTERS TO
410 ONLY 40 CHARACTERS. THE REASON FOR THIS IS THAT BUFFO CAN ONLY
420 BE USED FOR NUMBERS AND NUMBERS MORE THAN 12 CHARACTERS ARE ILLEGAL.
430
440 LISP 45
450 1) NEW FUNCTIONS: CLOCK, LOGOR, LOGAND, LOGXOR, LEFTSHIFT,
460 RIGHTSHIFT, ADD1, SUB1, ONEP, ZEROP, MAX, MIN, RECIP, LENGTH.
470 2) THE FLOAT PHASE OF THE ARITHMETIC FUNCTIONS HAS BEEN FIXED SO
480 THAT IT FLOATS A FIXED POINT ZERO CORRECTLY.
490 3) NEW ERROR--ERROR I4. ERROR I4 IS GIVEN BY AN ARITHMETIC
500 FUNCTION THAT FINDS A FLOATING POINT ARGUMENT WHEN IT WAS EXPECTING
510 A FIXED POINT ARGUMENT. LEFTSHIFT AND RIGHTSHIFT GIVE THIS ERROR
520 IF A FLOATING POINT NUMBER IS GIVEN AS THEIR SHIFT COUNT.
530
540 LISP 44
550 1) A BUCKET SORTED OBLIST HAS BEEN ADDED. THE OBLIST CONSISTS
560 OF 32 SUBLIST (NOT THE USUAL 64) IN ORDER THAT ON READING AN ATOM
570 ONLY A SMALL FRACTION OF THE OBLIST NEED BE SCANNED. THE RESULT
580 IS THAT READ AND READTTY HAVE BEEN SPEEDED UP BY ALMOST A FACTOR
590 OF FIVE OVER PREVIOUS VERSIONS.
600 2) COND HAS BEEN REDEFINED TO TAKE A GREATER VARIETY OF FORMS
610 OF ARGUMENTS. SEE COND--*** FOR DETAILS.
620 3) THE FUNCTIONS MAPLIST, MAPCAR, AND MAP HAVE BEEN ADDED. THEIR
630 DEFINITIONS HAVE BEEN CHANGED SLIGHTLY SO THAT THEIR FIRST
640 ARGUMENT IS A FUNCTION AND THEIR SECOND ARGUMENT IS A LIST TO APPLY
650 THIS FUNCTION TO. NORMAL ORDER IS TO HAVE THE FIRST ARGUMENT
660 BE THE LIST AND THE FUNCTION SECOND BUT THIS SEEMS AWKWARD CONSIDERIO
670 THE MATHEMATICAL NOTATION OF F(X) FOR FUNCTIONS.

LSPUPD CØNTINUED

```
680  4) EVAL AND APPLY HAVE BEEN MØDIFIED SØ THAT THEY DØ NØT MAKE A
690 SPECIAL CHECK FØR NIL ANY LØNGER.  THE RESULT IS THAT
700 (APPLY NIL ARGS ALIST) GIVES UNDEFINED FUNCTIØN--NIL.
710 HØWEVER, (EVAL NIL ALIST) STILL RESULTS IN NIL, SINCE NIL IS
720 A CØNSTANT THAT EVALUATES INTØ ITSELF.
730  5)  ØTHER CHANGES WERE MADE TØ IMPRØVE THE SPEED ØF LISP
740 (LIKE ALL INTERPRETIVE SYSTEMS LISP IS INTØLERABLY SLØW).
750 NØNE ØF THESE CHANGES CAN BE NØTICED BY THE USER.
760  6) LISP HAS BEEN GIVEN A NEW ERRØR--ERRØR TRA.  THIS ERRØR IS
770 GIVEN IF THE USER ATTEMPTS TØ MAKE A TRANSFER INTØ MEMØRY USING
780 SUBR ØR FSUBR FLAGS.  ØNLY PREDEFINED FUNCTIØNS ARE ALLØWED TØ
790 BE SUBRS ØR FSUBRS.
```

LIBEVA

```
100   THE FØLLØWING IS A CATLØG ØF LISP PRØGRAMS.  THE LISP PRØGRAMS IN
110 THE LIBRARY MAY NØT HAVE ALL THE NEEDED FUNCTIØNS INCLUDED WITHIN
120 THEM BUT USUALLY ØNE WILL FIND THE NECESSARY FUNCTIØNS ELSEWHERE
130 IN THE LIBRARY.
140
150   NØTE:  THØSE FILES THAT CØNTAIN SAMPLE DEFNITIØNS WILL NØT RUN
160 ØN THE LISP SYSTEM.  SINCE THESE FUNCTIØNS ARE PREDEFINED
170 A MEMØRY ERRØR WILL ØCCUR IF THEY ARE REDEFINED.  THESE DEFINITIØNS
180 ARE ØNLY USE TØ GIVE THE USER AN IDEA ØN HØW THE SYSTEM WØRKS.
185
190   ANYØNE HAVING PRØGRAMS ØF INTEREST TØ LISP USERS AND THINKS THEY
200 SHØULD BE IN THE LIBRARY SHØULD CØNTACT STEVE HØBBS AT THE
210 CØMPUTATIØN CENTER.
220
230 LISP02:   UNIØN; INTERSECTIØN;
240
250 DEFINE:  DEFLIST; DEFINE
260
270 SETQ--:  CØNTAINS SAMPLE DEFINITIØNS ØF SET AND SETQ.
280
290 TRACE-:  CØNTAINS A SIMULATØR FØR THE TRACE FEATURE IN 7090 LISP.
300 BØTH TRACE AND UNTRACE ARE FUNCTIØNS THAT TAKE AN UNLIMITED NUMBER
310 ØF ARGUMENTS WHICH ARE AUTØMATICALLY QUØTED.  FØR EXAMPLE:  TØ TRACE
320 ALPHA, BETA AND GAMMA  (TRACE ALPHA BETA GAMMA).  TRACE AND UNTRACE
330 WILL NØT WØRK ØN PREDEFINED FUNCTIØNS.
340
350 WANG--:   CØNTAINS THE WANG ALGØRITHM FØR PRØPØSITIØNAL CALCULUS
360 AS IN THE BACK ØF THE "LISP 1.5 PRØGRAMMERS MANUAL."  IT HAS AN
370 INPUT DRIVER SØ ØNE JUST TYPES (ARRØW(A1 A2 A3...)(B1 B2 B3...))
380 WHERE THE A'S ARE THE ELEMENTS ØF A LIST ØF PREMISES AND THE B'S
390 ARE THE ELEMENTS ØF A LIST ØF CØNCLUSIØNS.  WANG WILL RESPØND WITH
400 THE VALIDITY ØR INVALIDITY ØF THE STATEMENT.
410
420 BØØLE-:  CØNTAINS A SAMPLE DEFINITIØN ØF LISP BØØLEAN FUNCTIØNS:
430 AND; ØR; NØT;
440
450 LISP-1 AND LISP-2:  CØNTAINS A DISCUSSIØN ØF DIFFERENCES BETWEEN
460 THE DARTMØUTH VERSIØN ØF LISP AND THE LISP DESCRIBED IN THE "LISP
470 1.5 PRØGRAMMERS MANUAL."
480
490 LSPFNC, LSPFND AND LSPFNE:  CØNTAINS A LIST ØF PREDEFINED ATØMS
500 AND HØW THEY DIFFER FRØM THE "LISP 1.5 PRØGRAMMERS MANUAL."
510
520 LSPERR:  DESCRIBES THE PØSSIBLE LISP ERRØR MESSAGES.
530
540 TRCSET:  UNTRACESET; TRACESET;  TRACES SETQ IN FUNCTIØNS.  (DØ NØT
550 USE TRACESET AFTER A TRACE AS ØNLY THE TRACE RØUTINE
560 WILL BE TRACESET.  IF ØNE WANTS TØ USE BØTH TRACE AND TRACESET ØN THE
570 SAME FUNCTIØN THEN TRACESET IT FIRST FØLLØWED BY TRACE.)
580
```

LIBEVA CØNTINUED

590 LSPUPD: CØNTAINS A LIST CHANGES ØR ADDITIØNS THAT ARE MADE
600 IN EACH CHANGE IN LISP.
610
620 REMØB-: REMØB; RENAME
630
640 MKNAM-: CØNTAINS A SPECIAL VERSIØN ØF MKNAM. IT READS CØNSECUTIVE
650 CHARACTERS FRØM THE THE TTY UNTIL IT FINDS A BREAK CHARACTER (DEFINED
660 AS VARIABLE BRK) AND THEN RETURNS THE ATØM WITH THIS PRINT NAME.
670 THE BREAK CHARACTER THAT MARKED THE END ØF THE ATØM CAN BE FØUND
680 AS VARIABLE LCHAR. IT SHØULD BE NØTED THAT THIS PRØGRAM DØES
690 NØT HANDLE NUMBERS BUT ØNE MUST INSTEAD USE PACK AND NUMØB.

LSPERR***-----OCTOBER 30, 1968

THIS FILE IS A LIST OF POSSIBLE ERROR MESSAGES UNDER THE DARTMOUTH VERSION OF LISP. ALL MESSAGES GIVEN BY LISP THAT AGREE WITH THE "LISP 1.5 PROGRAMMERS MANUAL" HAVE IDENTICAL MEANING; HOWEVER, SOME ERRORS HAVE BEEN ADDED AND SOME HAVE BEEN REMOVED.

ERROR A1: APPLIED FUNCTION CALLED ERROR. THIS IS AN ERROR FORCED BY THE PROGRAMMER BY THE EVALUATION OF THE FUNCTION ERROR. ERROR IS A FUNCTION OF ONE ARGUMENT. ON THE LINE AFTER THE ERROR TYPE THE EVALUATION OF ERROR'S ARGUMENT ISPRINTED. ERROR IS USEFUL AS A DEBUGGING AID.

ERROR A2: FUNCTION OBJECT HAS NO DEFINITION--APPLY. APPLY HAS BEEN ASKED TO EVALUATE AN UNDEFINED FUNCTION. AFTER THE ERROR TYPE THE FUNCTION THAT WAS UNDEFINED IS PRINTED.

ERROR A4: SETQ GIVEN ON A NONEXISTENT PROGRAM VARIABLE. THE NONEXISTENT VARIABLE IS PRINTED AFTER THE ERROR TYPE.

ERROR A5: SET GIVEN ON A NONEXISTENT PROGRAM VARIABLE. THE NONEXISTENT VARIABLE IS PRINTED AFTER THE ERROR TYPE.

ERROR A6: GO (OR GO*) REFERS TO AN UNALBELLED POINT. THE UNLABELLED POINT IS PRINTED AFTER THE ERROR TYPE.

ERROR A8: EVAL ASKED TO EVALUATE AN UNBOUND VARIABLE. THE VARIABLE THAT IS NOT DEFINED (BY LAMBDA, PROG, OR CSET) IS PRINTED. THIS ERROR OFTEN OCCURS WHEN THERE IS A PARENTHESES MISCOUNT.

ERROR A9: FUNCTION OBJECT HAS NO DEFINITION--EVAL. THE UNDEFINED FUNCTION IS PRINTED. THIS ERROR OFTEN OCCURS WHEN THERE IS A PARENTHESES MISCOUNT.

ERROR A10: PUTPROP'S FIRST ARGUMENT IS NOT ATOMIC. THIS ERRO:TPFSO OCCUR WITH DEFPROP; CSET AND CSETQ

ERROR CH1: TOO MANY CHARACTERS--PACK. PACK HAS BEEN CALLED TO PACK MORE 81 CHARACTERS INTO BUFFO WITHOUT A CLEARBUFF.

ERROR CH2: FLOATING POINT NUMBER OUT RANGE--NUMOB. IT IS POSSIBLE ALSO TO GET THIS ERROR ON DECIMAL INTEGERS OF MORE THAN 12 CHARACTERS SINCE AT THIS POINT NUMBO CANNOT HAVE DECIDED WHETHER OR NOT THE NUMBER IS FLOATING POINT.

ERROR CH4: BAD CHARACTER--NUMOB

ERROR DSK: THERE HAS BEEN A DISK ERROR. IF THIS ERROR OCCURS DURING AN OPEN THEN READ, ADVANCE AND ADVANCE* BECOME UNDEFINED.

ERROR F2: PAIR'S FIRST ARGUMENT LIST IS TOO SHORT. THE EXTRA ELEMENTS ARE PRINTED AFTER THE ERROR TYPE. THIS MOST OFTEN HAPPENS WHEN TOO MANY ARGUMENTS ARE GIVEN TO A LAMBDA BOUND FUNCTION.

ERROR F3: PAIR'S SECOND ARGUMENT LIST IS TOO SHORT. THE EXTRA ELEMENTS ARE PRINTED AFTER THE ERROR TYPE. THIS MOST OFTEN HAPPENS WHEN TOO FEW ARGUMENTS ARE GIVEN TO A LAMBDA BOUND FUNCTION.
ERROR G1: ARITHMETIC OVERFLOW OR DIVIDE CHECK. ON THE GE 235 BOTH BIXED AND FLOATING POINT OPERATIONS CAN BE TRAPPED.
ERROR G2: OUT OF PUSH-DOWN LIST. THIS HAPPENS WHEN RECURSION IS TOO DEEP. IT OFTEN INDICATES INFINITE RECURSION.
ERROR GC2: NOW WORDS COLLECTED--GARBAGE COLLECTER. THIS MEANS THE EVALUATION OF PRESENT FUNCTIONS REQUIRE MORE THAN THE MACHINE'S CAPACITY FOR LIST STRUCTURE. THE 235 HAS A VERY SMALL CAPACITY.
ERROR I3: ARGUMENT OF A NUMERIC FUNCTION IS NOT A NUMBER.
ERROR I4: ARGUMENT OF A FUNCTION EXPECTING A FIXED POINT NUMBER IS A FLOATING POINT NUMBER.
ERROR MEM: THERE HAS BEEN AN ATTEMPT TO MODIFY A CELL THAT IS NOT IN FREE-STORAGE OR FULL-WORD-SPACE. THIS MAY HAPPEN IF THE USER TRYS TO MODIFY A PREDEFINED ATOM.
ERROR P2: ATOM HAS NO PRINT NAME--PRIN1. THIS HAPPENS WHEN AN ATOM IS ATTEMPTED TO BE PRINTED WITHOUT A PNAME, FIXNUM OR FLONUM ON ITS PROPERTY LIST. THIS MOST OFTEN HAPPENS WHEN ONE TRYS TO PRINT CARS OR CDRS BEYOND THE ATOMIC LEVEL.
ERROR P3: BAD BASE--SETOBASE OF SETIBASE. ONLY NUMBER BASES BETWEEN TWO AND TEN CAN BE USED FOR INTEGER INPUT-OUTPUT.
ERROR R1: FIRST OBJECT ON THE INPUT LIST IS ILLEGAL--READ. THIS MOST OFTEN HAPPENS WHEN THERE IS A MISPLACED <DOT> OR <RIGHT PARENTHESIS>.
ERROR R2: CONTEXT ERROR WITH DOT NOTATION--READ.
ERROR R3: ILLEGAL CHARACTER--READ. THE INPUT OF BAD END OF FILE. THIS IS CAUSED WHEN READ, ADVANCE OR ADVANCE* ATTEMPT TO READ BEYOND THE END OF A FILE.
ERROR R7: FILE NOT FOUND--OPEN. AN ATTEMPT WAS MADE TO OPEN A NONEXISTENT FILE. IF OPEN IS UNSUCESSFUL THEN READ, ADVANCE AND ADVANCE* ARE UNDEFINED.
ERROR TRA: THIS ERROR OCCURS WHEN THE INTERPRETER DISCOVERS XTW, THIS PREVENTS THE USER FROM TRANSFERING TO TO ILLEGAL LOCATIONS IN MEMORY.
END