



DECUS

PROGRAM LIBRARY

DECUS NO.	8-628
TITLE	LISP 1.5 INTERPRETER FOR PDP-8 WITH OS/8 (PS/8)
AUTHOR	Larry Davis, Washington University and Torbjorn Alm, Autocode AB
COMPANY	Submitted by: Robert Hassinger Liberty Mutual Research Center Hopkinton, Massachusetts
DATE	June 1973
SOURCE LANGUAGE	PAL-8

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

INSIDE THE SYSTEM A LISP CELL OCCUPIES TWO LOCATIONS IN CORE. THE FIRST LOCATION IS USED FOR THE 'CDR' PART AND THE NEXT LOCATION IS USED FOR THE 'CAR' PART. AS ALL CELLS START IN AN EVEN LOCATION, POINTERS ARE ALWAYS EVEN. THEREFORE, THE RIGHTMOST BIT HAS BEEN USED FOR OTHER PURPOSES, E.G. THE RIGHTMOST BIT IN THE 'CAR' PART IS THE ATOM MARK AND THE RIGHTMOST BIT IN THE 'CDR' PART IS USED IN THE GARBAGE COLLECTOR. IT IS NORMALLY ZERO. AS THE 'CDR' OPERATION IS MORE FREQUENT THAN THE 'CAR' OPERATION, THIS ARRANGEMENT ALLOWS THE 'CDR' TO BE SIMPLY PERFORMED BY THE INDIRECT ACCESS OF A LOCATION.

ACCESS TO MACHINE CODE PROGRAMING IS THROUGH THE FUNCTIONS 'EXPR' AND 'ZEXPR'. 'EXPR' IS USED TO ACCESS CODE IN FIELD 1 WERE THE BODY OF THE INTERPRETER IS LOCATED. 'ZEXPR' IS USED TO ACCESS CODE IN FIELD 0 WERE PS/8 INTERFACE IS LOCATED AND WERE CONSIDERABLE SPACE IS AVAILABLE FOR EXTENDED FUNCTIONS. REFER TO THE LIST OF MACHINE CODE FUNCTIONS FOR DETAILS.

THE SYSTEM IS LOCATED AT 10000-13765, 00034-00046, 00100-02377 AND 05400-07577, 07600-07777 AND 17600-17777 ARE NORMALLY RESERVED FOR THE PS/8 MONITOR HEAD. THE REST OF FIELD 1 IS USED AS THE LIST SPACE. BY CHANGING A SINGLE CONSTANT THE LIST SPACE CAN BE LENGTHENED OR SHORTENED. FOR EXAMPLE, MACHINE CODE PROGRAMS CAN BE PLACED ABOVE THE LIST SPACE. THESE CHANGES CAN BE EXPRESSED IN THE LISP LANGUAGE ITSELF.

THE AVAILABLE FUNCTIONS AND OBJECTS

THE FOLLOWING LIST OF OBJECTS AND FUNCTIONS ARE BUILT-IN FOR PS/8 LISP. WHEN THEY ARE IDENTICAL TO THE SAME FUNCTIONS IN 7090 LISP, THEY ARE DESCRIBED VERY SUMMARILY, BUT SEVERAL FUNCTIONS DEVIATE IN SOME RESPECTS AND ARE MORE FULLY DESCRIBED. IN ADDITION, A LISP DEFINITION OF THE FUNCTION IS GIVEN WHERE POSSIBLE. THE LISP DESCRIPTION IS NOT ALWAYS ACCURATE, E.G. NO ERROR TESTING WILL BE SHOWN, NOR WILL MACHINE MODE FUNCTION BE EXPRESSED ADEQUATELY.

BUILT-IN FUNCTIONS ARE RECOGNIZED BY THE VALUE OF THEIR POINTER. THEY DO NOT HAVE INDICATORS 'SUBR' OR 'FSUBR' ON THEIR PROPERTY LISTS. IN FACT THERE ARE FIVE KINDS OF BUILT-IN FUNCTIONS AND OBJECTS:

1. SPECIAL FUNCTIONS HAVING AN ARBITRARY NUMBER OF ARGUMENTS OR A VERY SPECIAL TREATMENT OF THEIR ARGUMENTS. THEY ARE:
'LAMBDA', 'NIL', 'FUNARG', 'T', 'APVAL', 'COND', 'FEXPR', 'FUNCTI', 'GO', 'LIST', 'MINUS', 'PLUS', 'TIMES', 'PROG', 'QUOTE', 'RETURN', 'SETQ', 'STOP', 'CLEAR', 'EXIT'.
2. FUNCTIONS WITH NO ARGUMENTS:
'GENSYM', 'READ', 'TERPRI', 'ICLOSE', 'OCLOSE'.
3. FUNCTIONS WITH ONE ARGUMENT:
'ATOM', 'CAR', 'CDR', 'DEFINE', 'NULL', 'NUMBER', 'PRINT'.
4. FUNCTIONS WITH TWO ARGUMENTS:
'ASSOC', 'CONS', 'DEFLIS', 'EQ', 'EQUAL', 'EVAL', 'GET', 'LESSP', 'RPLACA', 'RPLACD', 'SET'.
5. FUNCTIONS WITH THREE ARGUMENTS:
'APPLY', 'EXPR', 'ZEXPR', 'IOPEN', 'OOPEN'.

SUMMARY

THE LISP INTERPRETER, DECUS NO. 8-120A, HAS BEEN MODIFIED TO RUN UNDER PS/8; PS/8 FILE INPUT AND OUTPUT IS ALLOWED WHICH, AMONG OTHER THINGS, ALLOWS THE USER TO PREPARE LISP PROGRAMS USING THE PS/8 EDITOR. THE FUNCTIONS USED FOR FILE I/O ARE 'IOPEN', 'ICLOSE', 'OOPEN' AND 'OCLOSE'. ALSO LINE AT A TIME EDITING IS NOW AVAILABLE DIRECTLY WITHIN THE LISP INTERPRETER. 'TIMES', 'CLEAR', AND 'EXIT' FUNCTIONS HAVE BEEN ADDED. CCITT2 TELEPRINTER CODE IS NO LONGER SUPPORTED.

WRITE UP PRINT CONVENTIONS

DOUBLE QUOTES (") ARE USED AROUND NAMES OF CHARACTERS AS THEY APPEAR ON THE TERMINAL KEYBOARD. I.E. "\$", "(", "RETURN".

SINGLE QUOTES (') ARE USED AROUND THE PRINT NAMES OF LISP OBJECTS AND FUNCTIONS WHEN THEY APPEAR IN TEXT. I.E. 'ATOM', 'EVALQUOTE' IN EXAMPLE LISP EXPRESSIONS THE PRINT NAMES ARE USED WITHOUT THE QUOTES.

ANGLE BRACKETS (<,>) ARE USED AROUND DISSCRIPTIVE NAMES OF STRING ELEMENTS

INTRODUCTION

THE READER IS ASSUMED TO BE FAMILIAR WITH LISP, A PROGRAMMING LANGUAGE FOR LIST MANIPULATION. DETAILED DESCRIPTIONS CAN BE FOUND IN:

1. "LISP 1.5 PROGRAMMER'S MANUAL", J MCCARTHY ET AL, M.I.T PRESS, 1965
2. "THE PROGRAMMING LANGUAGE LISP - ITS OPERATION AND APPLICATION" E. C. BERKELEY AND D. G. BOBROW, EDS., INFORMATION INTERNATIONAL, INC.
3. "LISP 1.5 PRIMER", CLARK WEISSMAN, DICKINSON PUBLISHING COMPANY

THE LISP SYSTEM IS AN INTERPRETER-BASED SIMPLIFICATION OF LISP 1.5 AS IMPLEMENTED ON THE IBM 7090. THE RESTRICTIONS ON SIZE IN THE PDP-8 HAVE CAUSED THE OMISSION OF MANY ADVANCED FUNCTIONS, WHICH CAN BE DEFINED IN MORE BASIC FUNCTIONS. NEVERTHELESS, THE SYSTEM HAS BEEN DESIGNED IN A VERY GENERAL WAY, E.G. NO RESTRICTIONS ON THE LENGTH OF NAMES AND ON THE SIZE OF THE PUSH-DOWN LIST, ALL FUNCTIONS C...R WITH AS MANY AS 11 A'S AND D'S IN BETWEEN ARE BUILT IN, DECIMAL INPUT/OUTPUT (CAN BE SWITCHED TO OCTAL INDEPENDENTLY FOR INPUT AND OUTPUT). THE STORAGE IS USED BOTH FOR PUSH-DOWN SPACE AND LIST SPACE. ONLY IF THE TOTAL CAPACITY OF BOTH TOGETHER IS TOO BIG, THE SYSTEM STOPS. THE PUSH-DOWN IS ORGANIZED ITSELF AS A LIST IN THE NORMAL LIST SPACE. A VERY FULL ASSORTMENT OF ERROR STOPS IS INCLUDED. JUST AS IN 7090 LISP, THE PROGRAMMER NEED NOT QUOTE HIS ARGUMENTS ON THE OUTSIDE LEVEL; THE SYSTEM HAS THE FUNCTION 'EVALQUOTE'. INPUT IS IN ASCII CODE, CONTROL OVER THE CHOICE OF LOW OR HIGH SPEED READER CAN BE EXERCISED BY A MODE NUMBER INSTEAD OF A FUNCTION IN 'EVALQUOTE'. (SEE SYSTEM OPERATION)

CAR RETURNS WITH THE FIRST OF ITS ARGUMENT, WHICH MUST BE A LIST. IF ARGUMENT IS ATOMIC, THEN ERROR PRINT FOLLOWS. IF A 'CAR' FUNCTION IS NEEDED WHICH CAN LOOK INTO THE ATOM IT CAN BE SIMULATED WITH:

```
DEFINE(((CAR(LAMBDA(X)(CDR(EXPR 3172(PLUS(CONS NIL X)1)-1)) ) )))
```

CDR RETURNS WITH THE REST OF ITS ARGUMENT WHEN THE FIRST IS REMOVED. APPLICATION TO AN ATOM IS ALLOWED TO LOOK INTO PROPERTY LISTS; HOWEVER THIS IS NOT ADVISABLE FOR BUILT-IN FUNCTIONS AS THEY HAVE NO PROPERTY LIST EXCEPT FOR 'OBLIST' AND 'NIL'. INSTEAD THEY HAVE A POINTER INTO MACHINE CODE PROGRAM WHICH IS RECOGNIZED BECAUSE IT IS BELOW A CERTAIN LIMIT.

```
CDR(NIL) = NIL
```

C...R ANY FUNCTION WITH AS MANY AS ELEVEN A'S AND D'S MAY BE GIVEN. FOR EXAMPLE, (CAADDAR X) = (CAR(CAR(CDR(CDR(CAR X))))). ON FIRST APPEARANCE OF SUCH A FUNCTION, IT WILL BE PLACED ON THE 'OBLIST' WITH NO PROPERTIES. NEVERTHELESS, IT WILL BE RECOGNIZED AS A FUNCTION UNLESS IT HAS BEEN REDEFINED AS SOMETHING ELSE.

CLEAR CLEARS THE WHOLE SYSTEM; 'OBLIST' IS EMPTIED, 'GENSYM' COUNT IS ZEROED.

COND PSEUDO-FUNCTION HAVING AN INDETERMINATE NUMBER OF ARGUMENTS, EACH BEING A PAIR. IF THE FIRST OF A PAIR DOES NOT HAVE THE VALUE 'NIL', THE VALUE OF THE COMPLETE 'COND' FORM HAS THE VALUE OF THE SECOND OF THE PAIR. OTHERWISE THE NEXT PAIR IS TRIED. AN ERROR PRINT WILL FOLLOW IF 'COND' "DROPS OUT THE BOTTOM" EXCEPT IN 'PROG'.

CONS FUNCTION OF TWO ARGUMENTS. CONSTRUCTS A DOTTED PAIR FROM ITS ARGUMENTS.

DEFINE DEFINES A LIST OF FUNCTIONS BY ATTACHING A FORM TO THE NAME ON THE PROPERTY LIST WITH THE INDICATOR 'EXPR'. A FUNCTION, ALSO A BUILT-IN FUNCTION MAY BE REDEFINED. THE ORIGINAL SIGNIFICANCE IS THEN INACCESSIBLE UNLESS THE NAME IS REMOVED FROM THE 'OBLIST' WITH THE HELP OF THIS FEATURE, A STANDARD FUNCTION CAN BE REDEFINED TO PRINT RELEVANT INFORMATION ABOUT ITS ARGUMENTS AND VALUES, OR IT CAN COUNT ITS NUMBER OF CALLS. IT MAKES 'TRACE' SUPERFLUOUS.

```
DEFINE(((DEFINE(LAMBDA(X)(DEFLIS X EXPR)))))
```

APPLY THIS IS A FUNCTION OF THREE ARGUMENTS. IT APPLYS THE FUNCTION MENTIONED IN THE FIRST ARGUMENT TO THE LIST OF ARGUMENTS MENTIONED AS THE SECOND ARGUMENT. THE ASSOCIATION LIST TO BE USED IS GIVEN IN THE THIRD ARGUMENT (OFTEN 'NIL').

```

DEFINE((
  (APPLY(LAMBDA(FN ARGS ALIST)(PROG(K)
    ((NULL FN)(RETURN NIL))
    ((FN-POINTER<CERTAIN ADDRESS)(DO MACHINE CODE FUNCTION))
    ((ATOM FN)(GO A))
    B((EQ(CAR FN)FUNARG)(RETURN(APPLY(CADR FN)ARGS(CADDR FN))))
    ((EQ(CAR FN)LAMBDA)
      (RETURN(EVAL(CADDR FN)(PAIRLIS(CADR FN)ARGS ALIST))) )
    A(SETQ K(ASSOC FN ALIST))
    (COND(K(RETURN(APPLY(CDR K)ARGS ALIST))))
    SETQ K(GET FN EXPR))
    COND(K(RETURN(APPLY K ARGS ALIST))))
    (GO B) )))
  (PAIRLIS(LAMBDA(X Y A)(COND((NULL X)A)
    (T(CONS(CONS(CAR X)(CAR Y))(PAIRLIS(CDR X)(CDR Y)A))))))
  ))

```

REMARKS: THE GIVEN DEFINITIONS WILL TAKE ADVANTAGE OF THE FEATURES OF PS/8 LISP, E.G. 'COND' HAS BEEN OMITTED WHERE POSSIBLE (SEE UNDER 'COND') AND STANDARD OBJECT NAMES NEED NOT BE QUOTED. 'PAIRLIS' HAS BEEN DEFINED TOGETHER WITH 'APPLY' BUT IS NOT A STANDARD FUNCTION OF THE SYSTEM.

APVAL THIS IS USED AS AN INDICATOR FOR OBJECTS HAVING A VALUE ON THE PROPERTY LIST. BUT IN PS/8 LISP IT CAN ALSO BE USED AS A FUNCTION FOR SENSING BLANK TAPE. IT RETURNS 'T' WHEN A SYMBOL BLANK IS TYPED IN OR READ IN BY THE TAPE READER ; THE INPUT STEPS TO THE NEXT SYMBOL. WHEN THERE IS NO BLANK UNDER THE TAPE READER, IT RETURNS WITH 'NIL' AND THE INPUT DOES NOT MOVE TO THE NEXT SYMBOL. (FOR USE SEE ALSO 'T')

ASSOC LOOKS UP THE FIRST ARGUMENT ON THE ASSOCIATION LIST GIVEN IN THE SECOND ARGUMENT. RETURNS WITH THE PAIR FOUND. IF NOT FOUND THEN 'NIL'.

```

DEFINE(((ASSOC(LAMBDA(X A)(COND
  ((NULL A)NIL)
  ((EQ(CAR A)X)(CAR A))
  (T(ASSOC X(CDR A))))))

```

ATOM FUNCTION OF ONE ARGUMENT. RETURNS WITH 'T' IF ARGUMENT IS ATOMIC, OTHERWISE WITH 'NIL'.

EVAL RETURNS WITH THE EVALUATED FIRST ARGUMENT, WITH THE ASSOCIATION LIST GIVEN IN THE SECOND ARGUMENT.

```

DEFINE((
  (EVAL(LAMBDA(FORM ALIST)(PROG(K L)
    ((NULL FORM)(RETURN NIL))
    ((NUMBER FORM)(RETURN FORM))
    ((ATOM FORM)(GO A))
    B ((SETQ K(CAR FORM))
      ((EQ K QUOTE)(RETURN(CADR FORM)))
      ((EQ K FUNCTI)(RETURN(LIST FUNARG(CADR FORM)ALIST)))
      ((EQ K COND)(RETURN(EVCON(CDR FORM)ALIST)))
      ((EQ K PROG)(RETURN(EVPROG(CDR FORM)ALIST)))
      ((ATOM K)(GO C))
    D (RETURN(APPLY K(EVLIS(CDR FORM)ALIST)ALIST))
    A (SETQ K(ASSOC FORM ALIST))
      (COND(K(RETURN(CDR K)))
      (RETURN(GET FORM APVAL))
    C (SETQ L(GET K EXPR))
      (COND(L(RETURN(APPLY L(EVLIS(CDR FORM)ALIST)ALIST))))
      (SETQ L(GET K FEXPR))
      (COND(L(RETURN(APPLY L(LIST(CDR FORM)ALIST)ALIST))))
      (RETURN(EVAL(CONS(CDR(ASSOC K ALIST))(CDR FORM))ALIST)) )))
  (EVCON(LAMBDA(C A)(COND
    ((EVAL(CAAR C)A)(EVAL(CADAR C)A))
    (T(EVCON(CDR C)A)) )))
  (EVLIS(LAMBDA(L A)(COND
    ((NULL L)NIL)
    (T(CONS(EVAL(CAR L)A)(EVLIS(CDR L)A)))) )))

```

REMARK: THE FUNCTIONS "EVCON" AND "EVLIS" ARE NOT AVAILABLE AS STANDARD FUNCTIONS UNDER THESE NAMES.

EXIT FUNCTION OF NO ARGUMENTS; RETURNS TO THE PS/8 MONITOR AT 07600.

EXPR USED AS AN INDICATOR ON PROPERTY LISTS FOR FUNCTIONS WHICH HAVE THEIR ARGUMENTS EVALUATED. IN PS/8 LISP, "EXPR" ALSO SERVES AS A FUNCTION OF THREE ARGUMENTS. THE ARGUMENTS ARE SUPPOSED TO EVALUATE TO NUMERICAL VALUES (UNLESS NOT USED). IT JUMPS TO THE FIELD 1 MACHINE ADDRESS INDICATED IN THE FIRST ARGUMENT WITH THE NUMERICAL VALUE OF THE SECOND ARGUMENT IN THE ACCUMULATOR. THE POINTER TO THE SECOND ARGUMENT CAN BE FOUND IN 10037, AND THE POINTER TO THE THIRD ARGUMENT IN 10041. THE MACHINE CODE PROGRAM CAN RETURN TO THE LISP SYSTEM WITH THE INSTRUCTION 5171(JMP 171) IF NO VALUE NEEDS TO BE RETURNED. IF A NUMERICAL VALUE MUST BE RETURNED, THIS CAN BE DONE BY JUMPING TO 13175 WITH THE VALUE IN THE AC. BACK IN LISP ONE THEN HAS A POINTER TO THAT VALUE AS VALUE. IN THE SYSTEM, SEVERAL MACHINE CODE FUNCTIONS ARE PROVIDED FOR THE USER. SEE THE LIST FOR DETAILS. (NOTE: REFER TO "ZEXPR" FOR A SIMILAR FUNCTION WHICH REFERS TO LOCATIONS IN FIELD 0)

DEFLIS DEFINES A LIST OF FUNCTIONS BY ATTACHING A FORM TO THE NAME ON THE PROPERTY LIST WITH THE INDICATOR GIVEN IN THE SECOND ARGUMENT. ALL BUILT-IN FUNCTIONS HAVE NAMES OF NO MORE THAN SIX LETTERS, MAINLY FOR ECONOMY REASONS. ITS EFFECT IS EXACTLY THE SAME AS 'DEFLIST' IN THE 7090.

```

DEFINE((
  (DEFLIS(LAMBDA(L PRO)(MAPLIST L(FUNCTI(LAMBDA(J)
    (DEF1(CAAR J)(CADAR J)) )))))
  (DEF1(LAMBDA(OB L)(PROG NIL
    (RPLACA(PROP OB PRO)(FUNCTI(LAMBDA NIL
      (CDDR(RPLACD OB(CONS PRO(CONS NIL(CDR OB)))))) ))L)
    (RETURN OB) )))
  (PROP(LAMBDA(X Y FN)(PROG NIL
    A ((NULL X)(RETURN(FN)))
      ((EQ(CAR X)Y)(RETURN(CDR X)))
      (SETQ X(CDR X))
      (GO A) )))
  (MAPLIST(LAMBDA(X FN)(COND((NULL X)NIL)
    (T(CONS(FN X)(MAPLIST(CDR X)FN)) )))) ))

```

EQ RETURNS WITH 'T' WHEN ITS TWO ARGUMENTS ARE IDENTICAL POINTERS, OR WHEN THEY POINT TO NUMBERS WHICH HAVE THE SAME VALUE. OTHERWISE, ITS VALUE IS 'NIL'.

EQUAL RETURNS WITH 'T' IF ITS ARGUMENTS ARE EQUAL; 'NIL' IF THEY ARE UNEQUAL.

```

DEFINE(((EQUAL(LAMBDA(X Y)(COND
  ((ATOM X)(COND((ATOM Y)(EQ X Y))
    (T NIL) ))
  ((ATOM Y)NIL)
  ((EQUAL(CAR X)(CAR Y))(EQUAL(CDR X)(CDR Y))
    (T NIL) )))))

```

IOPEN FUNCTION OF THREE ARGUMENTS WHICH OPENS ANY FILE ON ANY PS/8 DEVICE FOR INPUT. ALL FURTHER INPUT WILL BE FROM THIS FILE UNTIL AN 'ICLOSE' IS ENCOUNTERED. IF AN END-OF-FILE MARKER IS FOUND (CONTROL-Z) AN ERROR MESSAGE IS PRINTED AND INPUT REVERTS TO THE STANDARD INPUT DEVICE (TELETYPE OR HIGH SPEED READER AS SELECTED BY MODE BIT 4). THE FIRST ARGUMENT IS THE PS/8 DEVICE NAME, THE SECOND ARGUMENT IS THE FILE NAME, AND THE THIRD ARGUMENT IS THE FILE EXTENSION. IF THE FIRST ARGUMENT IS ZERO, 'DSK' IS ASSUMED; IF THE SECOND ARGUMENT IS ZERO, NO FILE NAME IS ASSUMED, I.E. WHEN THE DEVICE IS NOT FILE STRUCTURED; IF THE THIRD ARGUMENT IS ZERO, NO EXTENSION IS ASSUMED. THE THREE ARGUMENTS CAN BE UP TO FOUR, SIX, AND TWO CHARACTERS RESPECTIVELY. ANY ADDITIONAL CHARACTERS ARE IGNORED. IF ANY ARGUMENT BEGINS WITH A DIGIT, THE FIRST DIGIT MUST BE PRECEDED BY A QUOTE ('), I.E.

IOPEN(DSK FILE2 '32)

SINCE THE THIRD ARGUMENT BEGINS WITH A DIGIT, THAT DIGIT MUST BE PRECEDED BY "'".

LAMBDA PSEUDO-FUNCTION OF TWO ARGUMENTS. BINDS THE FORMAL PARAMETERS GIVEN IN THE FIRST ARGUMENT (WHICH MUST BE A LIST) TO THE ACTUAL PARAMETERS. THE NUMBER OF ACTUAL AND FORMAL PARAMETERS MUST FIT IN THE CASE OF 'EXPR' AND 'ZEXPR'. THEN THE FORM GIVEN IN THE SECOND ARGUMENT IS EVALUATED. IT IS POSSIBLE TO GIVE MORE THAN TWO ARGUMENTS TO 'LAMBDA'. IN THAT CASE, ALL FORMS GIVEN WILL BE EVALUATED EXACTLY AS IN 'PROG'. THERE MAY BE LABELS AND 'GO' STATEMENTS ETC. THE CHECKING ON THE NUMBER OF FORMALS IS SUPPRESSED; INSTEAD ALL EXTRA FORMALS PLAY THE SAME ROLE AS 'PROG' VARIABLES AND ARE INITIALIZED TO 'NIL'.

LESSP FUNCTION OF TWO PARAMETERS. RETURN WITH 'T' IF FIRST ARGUMENT IS LESS THAN SECOND.

LIST EVALUATES AN ARBITRARY NUMBER OF ARGUMENTS AND RETURNS WITH A LIST OF THE VALUES.

DEFLIS(((LIST(LAMBDA(X A)(EVLIS X A))))FEXPR)

MINUS FUNCTION WITH AN ARBITRARY NUMBER OF ARGUMENTS. RETURNS WITH
 $\dots = X(N-2) + X(N-1) - X(N)$. IN PARTICULAR, (MINUS X) = -X
 AND (MINUS X Y) = X-Y.

NIL SERVES AS THE EMPTY LIST AND AT THE SAME TIME IS AN ATOM. CDR(NIL) = NIL. AS A FUNCTION, IT HAS AN INDETERMINATE NUMBER OF ARGUMENTS AND RETURNS 'NIL' AS VALUE. THIS CAN BE USEFUL AS A COMMENT, BUT ALL NEWLY READ WORDS ARE PLACED ON THE 'OBLIST'! FALSE IS ALWAYS REPRESENTED BY 'NIL'; 'F' DOES NOT HAVE THIS SIGNIFICANCE UNLESS IT HAS BEEN DEFINED SO EXPLICITLY.

- FEXPR INDICATOR ON PROPERTY LISTS TO INDICATE SPECIAL FORMS WHOSE ARGUMENTS ARE NOT EVALUATED.
- FUNARG ATOM USED ON THE ASSOCIATION LIST TO BE ABLE TO RECOVER PREVIOUS ASSOCIATION LISTS. ITS OPERATION IS AUTOMATICALLY INTRODUCED BY THE SYSTEM WHENEVER 'FUNCTI' IS USED. THE USER ALMOST NEVER SEES THE ATOM 'FUNARG' PRINTED. (SEE 'APPLY' AND 'EVAL')
- FUNCTI SAME AS 'FUNCTION' IN 7090 LISP. USED TO QUOTE LITERAL FUNCTIONS PRESENTED AS FUNCTIONAL ARGUMENTS.
- GENSYM FUNCTION WITH NO ARGUMENTS. IT RETURNS WITH A UNIQUELY CREATED NEW ATOM, WHICH WILL NOT BE ATTACHED TO THE 'OBLIST'. IT APPEARS IN PRINT AS: 'GGGG', 'GGIG', ..., 'GGVG', 'GGGH', 'GGHH', ..., 'GGVV', ..., 'GHGG', ECT.
- GET FUNCTION OF TWO ARGUMENTS. LOOKS ON THE PROPERTY LIST OF THE OBJECT GIVEN AS THE FIRST ARGUMENT TO FIND THE INDICATOR GIVEN IN THE SECOND ARGUMENT. WHEN THE SECOND ARGUMENT IS 'EXPR', 'FEXPR' OR 'APVAL', IT NEED NOT BE QUOTED AS BUILT-IN NAMES IN PS/8 LISP. A 'GET' APPLIED TO A BUILT-IN FUNCTION WILL ALWAYS RETURN WITH 'NIL'.
- ```

DEFINE(((GET(LAMBDA(OBJ IND)(PROG(K)
 (SETQ K(CDR OBJ))
 A ((NULL K)(RETURN NIL))
 ((EQ(CAR K)IND)(RETURN(CADR X)))
 (SETQ K(CDR K))
 (GO A))))))

```
- GO PSEUDO-FUNCTION OF A SINGLE ARGUMENT. GOES TO THAT ARGUMENT WHEN USED AS A LABEL WITHIN 'PROG'. OTHERWISE, AN ERROR PRINT FOLLOWS.
- ICLOSE FUNCTION OF NO ARGUMENTS WHICH CAUSES FURTHER INPUT TO BE FROM THE STANDARD SYSTEM DEVICE (TELETYPE OR HIGH SPEED READER AS SELECTED BY MODE BIT 4).



- PRINT** FUNCTION OF ONE ARGUMENT. WILL PRINT THE VALUE OF THE ARGUMENT AND ALSO RETURNS WITH THE VALUE. IF THE VALUE IS ATOMIC, IT WILL ONLY PRINT THE ATOM, NO SPACES WILL FOLLOW. NUMBERS ARE PRINTED WITH THE LEAST NUMBER OF DIGITS, NON-SIGNIFICANT ZEROS ARE OMITTED AND THE SIGN WHEN IT IS PLUS. IF THE ELEMENT TO BE PRINTED EXCEEDS THE CAPACITY OF THE LINE (64 CHARACTERS), A CARRIAGE-RETURN LINE-FEED WILL AUTOMATICALLY BE GIVEN. HENCE, PRINT(X) PRINT(Y) WILL CAUSE "XY" TO BE PRINTED. ANY SYMBOL CAN SERVE AS A LETTER IN A PRINT NAME EVEN SPACE, CARRIAGE RETURN, DOT, LEFT PARENTHESIS, ETC. FOR THAT PURPOSE, THEY MUST BE READ IN PRECEDED BY "'". THIS WILL QUOTE THE NEXT FOLLOWING SINGLE CHARACTER. EVEN "" ITSELF MAY SERVE AS A CHARACTER WHEN READ AS "'". HENCE, SPACE MAY BE PRINTED WITH PRINT(' '), OR ON THE INSIDE LEVEL WITH (PRINT(QUOTE ')). WHEN PRINTING BACK SUCH A STATEMENT FROM A PROPERTY LIST, IT CANNOT BE READ IN AGAIN AS THE "" HAS DISAPPEARED IN PRINT. IN THAT CASE, (EXPR 2160 @ NIL) IS PREFERABLE. A PRINT NAME WHICH IS LONGER THAN 64 CHARACTERS CANNOT BE PRINTED BACK; A NEW LINE WILL BE GIVEN BECAUSE THE NAME DID NOT FIT INTO THE PREVIOUS LINE, AND THIS SITUATION WILL CONTINUE TO BE SO. IF AN OUTPUT FILE (SEE 'OOPEN') IS OPEN, THE RESULT OF 'PRINT' GOES TO THE OPEN FILE.
- PROG** HAS THE SAME MEANING AS IN 7090 LISP. THE FIRST ARGUMENT IS A LIST OF PROGRAM VARIABLES, ALL SET TO 'NIL' ON ENTRY; THEN FOLLOW STATEMENTS AND LABELS. LABELS ARE ATOMS, STATEMENTS ARE NOT. IN A 'PROG', A 'COND' STATEMENT IS ALLOWED TO FALL OUT OF THE BOTTOM. 'PROG' IS LEFT ON A 'RETURN' STATEMENT, AND THE VALUE RETURNED IS THE VALUE OF THE ARGUMENT OF 'RETURN'.
- QUOTE** PREVENTS ITS ARGUMENT FROM BEING EVALUATED. THE VALUE IS THE UNEVALUATED ARGUMENT. STANDARD BUILT-IN OBJECTS NEED NOT BE QUOTED IN PS/8 LISP.
- DEFLIS(((QUOTE(LAMBDA(X A)(CAR X))))FEXPR)
- READ** FUNCTION OF NO ARGUMENTS. READS A SINGLE S-EXPRESSION FROM TAPE OR KEYBOARD OR THE OPEN INPUT FILE (SEE 'IOPEN'). ALL IDENTIFIERS READ FOR THE FIRST TIME ARE PUT ON THE 'OBLIST'. IDENTIFIERS MAY CONSIST OF ANY NUMBER OF CHARACTERS AND ANY CHARACTER EXCEPT LEFT PARENTHESIS, RIGHT PARENTHESIS, DOT, SPACE, CARRIAGE-RETURN, LINE-FEED, BLANK, AND APOSTROPHE; HOWEVER, THESE CHARACTERS CAN BE "QUOTED" BY PRECEDING THEM WITH "'". THEN, THEY MAY AGAIN BE A CHARACTER OF A NAME. A NAME MUST START WITH A LETTER. AN OBJECT STARTING WITH A DIGIT OR A PLUS SIGN OR A MINUS SIGN IS REGARDED AS A NUMBER (EXCEPT WHEN PRECEDED BY "'"). AN ISOLATED PLUS SIGN AND MINUS SIGN IS ALSO A LEGAL NAME, E.G. (LEFTHANDVARIABLE := A \* TEMPERATURE) IS A PERFECTLY CORRECT S-EXPRESSION AND WILL BE PRINTED EXACTLY THE SAME; HOWEVER, WHEN ('234567 + 7777 A)B) IS READ IT WILL PRINT AS (234567 + -1 A)B).

NULL FUNCTION OF ONE ARGUMENT. RETURNS WITH 'T' IF ARGUMENT IS 'NIL'. 'NULL' HAS THE SAME ACTION AS 'NOT', WHICH IS NOT INCLUDED IN THE SYSTEM.

```
DEFINE(((NULL(LAMBDA(X)(EQ X NIL))))))
```

NUMBER SAME AS 'NUMBERP' IN 7090 LISP. RETURNS WITH 'T' IF ARGUMENT IS A NUMBER.

OBLIST LIST OF ALL PROGRAMMER-DEFINED OBJECTS. THE OBJECT 'OBLIST' ITSELF ALWAYS APPEARS AT THE BOTTOM OF THE 'OBLIST'; FOR TECHNICAL REASONS THERE IS AN EXTRA 'NIL' AT THE TOP. HENCE: CDR(OBLIST) = (APVAL(NIL ... ALL DEFINED OBJECTS ... OBLIST)) THE 'OBLIST' MAY BE MANIPULATED BY THE PROGRAMMER INCLUDING THE ELEMENT 'NIL'. CAUTION: GREAT CARE MUST BE TAKEN NOT TO REMOVE 'OBLIST' FROM THE 'OBLIST'.

OCLOSE FUNCTION OF NO ARGUMENTS. WHEN USING THE SPECIAL FORM OF 'OOPEN' TO BYPASS THE LINE PRINTER DEVICE HANDLER, 'OCLOSE' SIMPLY RESTORES THE STANDARD OUTPUT DEVICE (TELETYPE). WHEN A FILE HAS BEEN OPENED, THE DIRECTORY IS UPDATED, THE FILE IS CLOSED, AND A CONTROL-Z IS APPENDED TO THE FILE.

OOPEN FUNCTION OF THREE ARGUMENTS. OPENS ANY FILE ON ANY PS/8 DEVICE FOR OUTPUT. ALL FURTHER OUTPUT (EXCEPT FOR ECHO, IF SPECIFIED) GOES TO THIS FILE UNTIL AN 'OCLOSE' IS ENCOUNTERED. IF THERE IS NOT ENOUGH ROOM ON THE DEVICE FOR THE ENTIRE FILE, AN ERROR MESSAGE IS PRINTED AND OUTPUT REVERTS TO THE SYSTEM OUTPUT DEVICE (TELETYPE), AND CONTROL IS RETURNED TO 'EVALQUOTE'. THE FILE IS NOT CLOSED AND NO ENTRY WILL BE MADE IN THE DIRECTORY. SEE 'IOPEN' FOR THE FORM OF THE ARGUMENTS. IF THE FIRST ARGUMENT IS A NON-ZERO NUMBER, THE OUTPUT DEVICE IS ASSUMED TO BE A DEVICE FOR WHICH A SUBROUTINE HAS BEEN WRITTEN IN ORDER TO BYPASS THE PS/8 DEVICE HANDLER. THE SUBROUTINE IS INSERTED AT LPTOUT (00600). THE CHARACTER TO BE PRINTED IS IN THE AC IN 8 BIT ASCII AT ENTRY TO THE SUBROUTINE. THE AC MUST BE RESTORED UPON RETURN. A ROUTINE FOR THE SCM KLEINSHMIDT M-311 PRINTER IS SUPPLIED WITH THE SYSTEM. EXAMPLES OF OTHER ROUTINES ARE INCLUDED ON THE SOURCE TAPE AS OVERLAYS.

PLUS FUNCTION OF AN INDETERMINATE NUMBER OF ARGUMENTS. WILL PERFORM A SIGNED ADDITION ON INTEGERS ONLY. NO TEST ON THE EXCEEDING OF CAPACITY IS PERFORMED. NO TEST IS MADE ON WHETHER THE ARGUMENT(S) IS(ARE) NUMERICAL.



- TERPRI** THIS WILL PRINT CARRIAGE-RETURN, LINE-FEED AND WILL SET THE LINE COUNT TO ZERO. ITS VALUE IS 'NIL'. PRINT("CARRIAGE-RETURN") AND EXPR(2160 6 NIL) EXPR(2160 3 NIL) WOULD NOT HAVE EXACTLY THE SAME EFFECT, AS THESE DO NOT RESET THE LINE COUNT.
- TIMES** FUNCTION OF AN INDETERMINATE NUMBER OF ARGUMENTS. WILL PERFORM A SIGNED MULTIPLICATION ON INTEGERS ONLY. NO TEST FOR OVERFLOW IS MADE. NO CHECK IS MADE TO INSURE THAT THE ARGUMENTS ARE NUMERIC.
- ZEXPR** FUNCTION OF THREE ARGUMENTS. SIMILAR TO 'EXPR', EXCEPT THE LOCATION REFERENCED BY THE FIRST ARGUMENT IS IN FIELD 0. THE ARGUMENTS ARE SUPPOSED TO EVALUATE TO NUMERICAL VALUES (UNLESS THEY ARE NOT USED). JUMPS TO THE FIELD 0 ADDRESS INDICATED BY THE FIRST ARGUMENT WITH THE NUMERICAL VALUE OF THE SECOND ARGUMENT IN THE AC. A POINTER TO THE SECOND ARGUMENT IS AT 00037, AND A POINTER TO THE THIRD ARGUMENT IS IN 00041. THE MACHINE CODE CAN RETURN TO THE LISP SYSTEM WITH THE INSTRUCTION JMP 42 IF NO VALUE NEEDS TO BE RETURNED (CLEAR THE AC BEFORE RETURNING IN THIS CASE). IF A NUMERICAL VALUE IS TO BE RETURNED, THIS IS DONE BY HAVING THE VALUE IN THE AC, AND DOING A JMP 44. BACK IN LISP, ONE THEN HAS A POINTER TO THAT VALUE AS A VALUE. SEE THE LIST OF MACHINE CODE FUNCTIONS FOR USEFUL ROUTINES THAT ARE BUILT INTO THE SYSTEM.



RETURN FUNCTION OF ONE ARGUMENT. IF USED TO RETURN FROM A 'PROG' IT HAS THE VALUE OF THE ARGUMENT.

RPLACA REPLACE THE 'CAR' LINK OF THE FIRST ARGUMENT BY THE SECOND ARGUMENT.

RPLACD REPLACE THE 'CDR' LINK OF THE FIRST ARGUMENT BY THE SECOND ARGUMENT. BOTH THESE FUNCTIONS MUST BE USED WITH GREAT CARE AS THEY CAN DESTROY EXISTING LIST STRUCTURE. IN PARTICULAR ONE MUST NEVER TRY TO REPLACE THE PROPERTY LIST OF A STANDARD ATOM UNLESS THAT ATOM IS REDEFINED WITH 'DEFINE' OR 'DEFLIS'. 'RPLACA' WILL LEAVE THE ATOM MARK INTACT.

SET FUNCTION OF TWO ARGUMENTS. SETS THE VALUE OF THE FIRST ARGUMENT, BOUND ON THE ASSOCIATION LIST EQUAL TO THE VALUE OF THE SECOND ARGUMENT. IT RETURNS WITH THE VALUE OF THE SECOND ARGUMENT.

```
DEFLIS((
 (SET(LAMBDA(X A)(RPLACD(ASSOC(EVAL(CAR X)A)A)
 (EVAL(CADR X)A)))))FEXPR)
```

SETQ SAME AS 'SET', BUT AUTOMATICALLY QUOTES ITS FIRST ARGUMENT. 'SET' AND 'SETQ' CAN ONLY EFFECTIVELY BE USED WITHIN 'LAMBDA' OR 'PROG'.

```
DEFLIS(((SETQ(LAMBDA(X A)(RPLACD(ASSOC(CAR X)A)
 (EVAL(CADR X)A)))))FEXPR)
```

STOP NO ARGUMENTS. RETURNS WITH 'NIL'. PRESS "CONTINUE" TO RESTART.

T HAS VALUE 'T' FOR TRUE. IN PS/8 LISP, 'T' ALSO IS A FUNCTION WHICH WILL RETURN ITS ARGUMENT AS A VALUE. THIS IS USED ESPECIALLY IN 'PROG'. 'COND' FORMS CAN BE ABBREVIATED TO THE CONDITION PAIRS ONLY. WHEN THE FIRST ELEMENT EVALUATES TO 'NIL', THE FUNCTION WILL RETURN WITH 'NIL'; BUT WHEN THE FIRST ELEMENT EVALUATES TO 'T', THE VALUE OF THE SECOND ARGUMENT IS TAKEN, E.G. (COND((NULL L)(GO A))) CAN BE ABBREVIATED TO ((NULL L)(GO A)). WITH 'COND' EVERY BOOLEAN NOT 'NIL' IS REGARDED AS 'T'. THIS IS NOT THE CASE IN ABBREVIATED PAIRS. THE FIRST ELEMENT MUST EVALUATE TO 'T' OR 'NIL'. ANOTHER EXAMPLE IS SKIPPING BLANK TAPE BEFORE 'READ'. ALTHOUGH 'READ' WILL DO THIS AUTOMATICALLY, ONE HAS NO WAY OF KNOWING THAT THERE HAS BEEN BLANK OTHER THAN: ... (PROG(K L)(GO B) A (SETQ K T) B ((APVAL)(GO A)) (SETQ L(READ))...) 'K' WILL BE 'NIL' UNLESS BLANK HAS BEEN SKIPPED BY THE FUNCTION 'APVAL'. WHEN IT IS DESIRABLE TO HAVE 'T' AVAILABLE AS A NORMAL VARIABLE ONE CAN GIVE:

```
DEFINE(((T NIL))) RPLACD(T NIL)
```

THIS REDEFINES 'T' AND THEN REMOVES ALL ITS PROPERTIES.

```
DEFINE(((T(LAMBDA(X)X))) DEFLIS(((T T))APVAL)
```

## OPERATION OF THE SYSTEM

-----  
LOADING FROM PS/8:

.RUN DEV LISP  
OR  
.R LISP

(THE STARTING ADDRESS OF LISP IS 13000)

AFTER THE SYSTEM IS LOADED, IT CAN BE RESTARTED AT:

13000 THIS WILL CLEAR THE WHOLE SYSTEM. THE 'OBLIST' IS  
EMPTIED AND THE 'GENSYM' COUNT IS ZEROED. (SEE ALSO 'CLEAR')

13001 THIS WILL INITIALIZE THE SYSTEM BUT WILL KEEP THE  
'OBLIST' AND ALL PROPERTIES OF THE OBJECTS. IN BOTH CASES  
'EVALQUOTE' WILL BE ENTERED.

07600 THIS RESTARTS PS/8. (SEE ALSO 'EXIT')

## MODE CONTROL:

CONTROL OVER WHAT THE INPUT AND OUTPUT DEVICES ARE IS BY  
USE OF A MODE NUMBER. THE MODE NUMBER IS INPUT IN THE PLACE OF THE  
FIRST ELEMENT OF AN 'EVALQUOTE' PAIR. THIS MODE WILL NOW BE SET AND THE  
SYSTEM WILL AGAIN EXPECT A PAIR (OR CAN ACCEPT ANOTHER MODE NUMBER).  
THERE ARE FOUR BITS IN THE MODE NUMBER WHICH HAVE SIGNIFICANCE.

WHEN 1 IS ABSENT, ALL TELETYPE INPUT WILL BE ECHOED ON THE TELETYPE.  
THERE IS NO ECHO FOR THE HIGH SPEED READER OR AN OPEN FILE  
(SEE 'IOPEN'). ALSO THE FUNCTION 'READ' WILL ECHO.

WHEN 1 IS PRESENT, THERE WILL BE NO ECHO BUT EACH S-EXPRESSION OF  
THE 'EVALQUOTE' PAIR WILL BE REPRODUCED AFTER IT HAS BEEN  
COMPLETELY READ IN.

WHEN 2 IS ABSENT, THE VALUE OF EVALQUOTE WILL NOT BE PRINTED BUT ANY  
EXPLICIT PRINT FUNCTION IN THE PROGRAM WILL PRINT.

WHEN 2 IS PRESENT, THE VALUE OF EVALQUOTE WILL BE PRINTED.

WHEN 4 IS ABSENT, READING TAKES PLACE FROM THE KEYBOARD (OR LOW SPEED  
READER) EXCEPT WHEN AN INPUT FILE IS OPEN (SEE 'IOPEN').

WHEN 4 IS PRESENT, READING TAKES PLACE FROM THE HIGH SPEED READER  
EXCEPT WHEN AN INPUT FILE IS OPEN (SEE 'IOPEN').

WHEN 40 (32 DECIMAL) IS PRESENT, ONLY THE ECHO IS SUPPRESSED.

MODE 2 IS INITIALLY INSTALLED. EACH CHANGE IN MODE WILL PERSIST UNTIL  
EXPLICITLY CHANGED AGAIN, BUT ANY ERROR WILL REINSTALL MODE 2.

IMPORTANT LOCATIONS IN THE SYSTEM  
-----

## FIELD 0

THESE LOCATIONS CAN BE INSPECTED WITH ZEXPR(1710 X 0)

00037 POINTER TO SECOND ARGUMENT OF 'ZEXPR' (ARGUMENT IS IN FIELD 1)

00041 POINTER TO THIRD ARGUMENT OF 'ZEXPR' (ARGUMENT IS IN FIELD 1)

00042 JMP 42 WITH AC=0 TO RETURN WITH NO VALUE FROM 'ZEXPR'

00044 JMP 44 WITH AC=VALUE TO RETURN WITH A VALUE FROM 'ZEXPR'

## FIELD 1

THESE LOCATIONS CAN BE INSPECTED WITH EXPR(3172 X -1)

10006 CONTAINS THE CUMULATIVE NUMBER OF TIMES THAT THE GARBAGE  
COLLECTOR HAS BEEN CALLED.

10014 CONTAINS THE LAST CHARACTER READ.

10015 CONTAINS QUOTIENT AFTER DIVIDE, USED BY MULTIPLE  
PRECISION ARITHMETIC ALSO.

10016 CONTAINS THE LINE COUNT, COUNTING FROM -77 TO 0.

10017 CONTAINS THE 'GENSYM' COUNT FROM WHICH 'GENSYM' NAMES ARE DERIVED.

10025 CONTAINS THE POINTER TO THE ASSOCIATION LIST.

10027 CONTAINS THE POINTER TO THE STACK.

10030 CONTAINS THE POINTER TO THE FREE LIST.

10035 CONTAINS THE POINTER TO THE FIRST ARGUMENT.

10037 CONTAINS THE POINTER TO THE SECOND ARGUMENT.

10041 CONTAINS THE POINTER TO THE THIRD ARGUMENT.

10045 CONTAINS THE BEGINNING POINT OF THE LIST SPACE MINUS THE END  
POINT. THIS MUST BE AN EVEN DIFFERENCE.

10100 CONTAINS THE ADDRESS OF THE BASIC 'IN' ROUTINE.

12270 CONTAINS THE ADDRESS OF THE BASIC 'OUT' ROUTINE.



THE OPERATION OF THE SYSTEM CAN BE CONCISELY DESCRIBED BY THE FOLLOWING PROGRAM FOR 'EVALQUOTE':

```

DEFINE(((EVALQUOTE(LAMBDA()(PROG(FN ARGS VALUE MODE ECHO HI)
START13000 (CLEAR)
 (CLEARGENSYM)
ERROR (SETQ MODE 2)
START13001 (TERPRI)
 (SETQ ECHO (EQ(LOGAND MODE 45)0))
 (SETQ HI (EQ(LOGAND MODE 4)4))
 (SETQ FN (READ))
 ((EQ(LOGAND MODE 1)0)(GO NOFN))
 (PRINT FN)
NOFN ((NUMBER FN)(GO NEWMODE))
 (SETQ ARGS (READ))
 ((EQ(LOGAND MODE 1)0)(GO NOARGS))
 (PRINT ARGS)
 (TERPRI)
NOARGS (SETQ VALUE (APPLY FN ARGS NIL))
 ((EQ(LOGAND MODE 2)0)(GO START13001))
 (PRINT VALUE)
 (GO START13001)
NEWMODE (SETQ MODE FN)
 (GO START13001))))

(CLEAR(LAMBDA()(RPLACD OBLIST(QUOTE(OBLIST))))))
(CLEARGENSYM(LAMBDA()(EXPR 3202 17 0)))
(LOGAND(LAMBDA(X Y)(EXPR 3174 X Y)))))) EVALQUOTE NIL

```

SOME FEATURES HAVE NOT BEEN SHOWN ADEQUATELY. THE FUNCTION 'READ' WILL PRINT AN IMMEDIATE ECHO IF 'ECHO' = 'T'. THE HIGH SPEED READER WILL BE USED IF 'HI' = 'T'. AN ERROR WILL GIVE AN ERROR PRINT AND WILL GO TO 'ERROR'. IF ANYWHERE INSIDE AN 'EVALQUOTE' PAIR A BLANK OR TRAILER CODE IS READ THIS IS AN ERROR. BLANK WILL ONLY BE SKIPPED BEFORE A PAIR. THIS FEATURE CAN BE USED TO STOP A TAPE ON THE HIGH SPEED READER BY ENDING THE PREVIOUS PAIR WITH A CARRIAGE RETURN, LINE FEED, BLANK. THE FUNCTION READ WILL THEN REACT AS IF THE NEW PAIR HAD BEGUN BECAUSE OF THE CARRIAGE-RETURN AND GIVES AN ERROR STOP. THE ERROR STOP THEN AUTOMATICALLY RETURNS THE MODE TO 2 AND HENCE SWITCHES TO KEYBOARD. SOME ACTIONS WILL BE DIFFERENT IF INPUT OR OUTPUT FILES ARE OPEN (SEE 'IOPEN', 'OOPEN').

AN ERROR WILL RESULT IN AN ERROR PRINTOUT THAT CANNOT BE SUPPRESSED BY THE MODE NUMBER AND WHICH HAS THE FORMAT:

STOP N FOLLOWED BY SOME INDICATION OF WHAT HAS GONE WRONG, E.G. THE NAME OF THE FUNCTION.

AFTER AN ERROR THE SYSTEM WILL IMMEDIATELY GO ON READING THE NEXT PAIR FOR EVALQUOTE. THIS CAN GIVE RISE TO FURTHER ERRORS IF, FOR EXAMPLE, THE NUMBER OF PARENTHESIS IN THE INPUT IS WRONG.

THERE ARE TWO ERRORS OF A SPECIAL KIND:

- ? WILL BE PRINTED IF ALL THE WORKING SPACE HAS BEEN USED, WHETHER FOR STACK OR LIST. IN THIS CASE PUSHDOWN LIST, ASSOCIATION LIST, ETC. ARE CLEARED AND ANOTHER GARBAGE COLLECTION IS MADE. THE SYSTEM THEN COMES TO A STOP. IF "CONTINUE" IS PRESSED, THE NEXT PAIR FOR EVALQUOTE WILL BE READ. IN EXTREME CASES, EVEN BY CLEARING THE STACK AND A-LIST, NEW SPACE CANNOT BE FOUND AND THE GARBAGE COLLECTOR WILL CONTINUE TO PRINT "?".
- ⊙ WILL BE PRINTED IF THE SPECIAL COLLECTOR STACK OVERFLOWS. THIS WILL RARELY BE THE CASE AS THE STACK IS ONLY USED IN THE 'CAR' DIRECTION. A DEPTH OF 42 IS PROVIDED. ALSO, A SECOND COLLECTOR RUN IS TRIED, AND THE SYSTEM STOPS AND MUST BE RESTARTED WITH "CONTINUE".

CHANGING THE SIZE OF THE WORKING SPACE:

THE SYSTEM CAN CHANGE THE SIZE OF THE WORKING SPACE BY EXECUTING:

```
EVAL((EXPR 3202 45(MINUS 3735 <LAST USABLE ADDRESS(ODD)>))NIL)
```

THIS WILL PUT IN 10045 THE DIFFERENCE BETWEEN THE FIRST USABLE ADDRESS (=3725) AND THE LAST USABLE ADDRESS (NORMALLY 7577). THE CHANGE IN 10045 WILL HAVE NO IMMEDIATE EFFECT BUT WILL COME INTO PLAY ON THE NEXT RESTARTING (OR ON A COLLECTOR RUN). THEREFORE IT IS ADVISABLE TO HAVE THE CHANGE IN SIZE FOLLOWED IMMEDIATELY BY EXPR(3000 NIL NIL) OR CLEAR TO REINITIALIZE THE SYSTEM AND TO CREATE A NEW FREE SPACE. CHANGING THE SIZE GIVES THE POSSIBILITY TO ADD MACHINE CODE FUNCTIONS IN FIELD 1 OR TO RUN A SMALL LISP COMPILER TO COMPILE MACHINE CODE FUNCTIONS BEHIND THE LIST SPACE.

IF WORKING SPACE IS AT A PREMIUM, ONE CAN ENLARGE THE WORKING SPACE TO THE MAXIMUM AMOUNT POSSIBLE, THEREBY OVERWRITING THE PS/8 MONITOR HEAD IN FIELD 1 BY EXPR(3202 45 3756) OR EXPR(1666 37 2030) IN DECIMAL. NOTE THAT THE PS/8 SYSTEM WILL HAVE TO BE BOOTSTRAPED AFTER DOING THIS.

PS/8 PROGRAMMERS WILL NOTE THAT THE AREA FROM 17600 TO 17646 IS USED ONLY TO HOLD THE OUTPUT OF THE COMMAND DECODER. AS THIS IS NORMALLY NOT USED WITH LISP THIS SPACE CAN BE ADDED TO THE WORKING SPACE WITHOUT DESTROYING THE SYSTEM HEAD.

## EDITING CAPABILITIES IN THE INTERPRETER

---

WHEN INPUT IS FROM THE TELETYPE, CHARACTERS ARE PLACED IN A BUFFER. NO TRANSMISSION OF CHARACTERS TO THE LISP SYSTEM OCCURS UNTIL A CARRIAGE RETURN OR ALTMODE IS STRUCK. WHEN EITHER OF THESE IS STRUCK, NO MORE INFORMATION IS READ FROM THE TELETYPE UNTIL ALL OF THE CHARACTERS IN THE BUFFER HAVE BEEN USED.

THIS METHOD OF BUFFERED INPUT WAS CHOSEN IN ORDER THAT EDITING MAY BE DONE ON A TEXT LINE BEFORE TRANSMITTING THE LINE TO LISP. EDITING MAY ONLY BE DONE ON THE CURRENT LINE. CHARACTERS WITH SPECIAL SIGNIFICANCE ARE:

1. RUBOUT            ECHOS (IF ECHO IS SPECIFIED) AS "\". THIS DELETES THE LAST CHARACTER IN THE BUFFER.
2. CONTROL-U        ECHOS AS "U". THIS DELETES THE ENTIRE BUFFER.
3. CONTROL-C        ECHOS AS "C". THIS CLEARS THE LISP SYSTEM (SAME AS CLEAR).
4. CONTROL-R        ECHOS AS "R". RESTARTS THE LISP SYSTEM (SAME AS START 13001)
5. RETURN.           ECHOS AS A CARRIAGE-RETURN AND LINE-FEED. A CARRIAGE-RETURN IS INSERTED AS THE LAST CHARACTER IN THE BUFFER AND TRANSMISSION OF THE CHARACTERS IN THE BUFFER BEGINS.
6. ALTMODE.         SAME AS "RETURN", BUT NO ECHO OCCURS.

THE BUFFER IS 126 CHARACTERS LONG. MORE CHARACTERS (OTHER THAN RETURN OR ALTMODE) ARE IGNORED UNTIL RETURN OR ALTMODE IS STRUCK.

## ASSEMBLING AND LOADING LISP

---

```
.R PAL8
*LISP/LS
.SAVE SYS LISP;13000
```

OR:

```
.R PAL8
*LISP,LISP<LISP
.R ARSLDR
*LISPS
.SAVE DSK LISP;13000
```



## LIST OF ERROR CODES

-----  
 DEPENDING ON WHETHER THE SYSTEM IS PRINTING IN DECIMAL OR IN OCTAL, THE  
 FOLLOWING ERROR CODES WILL BE PRINTED:

| OCTAL | DECIMAL | KIND OF ERROR                                                                              |
|-------|---------|--------------------------------------------------------------------------------------------|
| ----- | -----   | -----                                                                                      |
| 164   | 116     | ERROR IN OPENING/CLOSING FILE; ARGUMENTS NOT ATOMS                                         |
| 171   | 121     | ERROR IN OUTPUTTING A CHARACTER ON CURRENTLY OPEN FILE                                     |
| 213   | 139     | VALUE OF THIS VARIABLE IS NOT DEFINED                                                      |
| 243   | 163     | A NUMBER IS STANDING IN THE PLACE OF A FUNCTION                                            |
| 331   | 217     | BUILT-IN FUNCTION HAS TOO FEW ARGUMENTS                                                    |
| 346   | 230     | BUILT-IN FUNCTION HAS TOO MANY ARGUMENTS                                                   |
| 422   | 274     | THIS FUNCTIONAL ARGUMENT IS NOT A FUNCTION                                                 |
| 501   | 321     | 'LAMBDA' FORM HAS TOO FEW ARGUMENTS                                                        |
| 522   | 338     | 'LAMBDA' FORM HAS TOO MANY ARGUMENTS                                                       |
| 554   | 364     | 'GO', 'RETURN', OR 'COND' WITH UNDEFINED VALUE HAS BEEN<br>ENCOUNTERED OUTSIDE OF A 'PROG' |
| 567   | 375     | 'GO' HAS AN UNKNOWN LABEL                                                                  |
| 744   | 484     | FIRST ARGUMENT OF 'SET' OR 'SETQ' IS NOT ATOMIC                                            |
| 1022  | 530     | WRONG NUMBER OF ARGUMENTS IN THIS FUNCTION                                                 |
| 1231  | 665     | FIRST ELEMENT OF A PAIR IN 'DEFINE' OR 'DEFLIS' IS NOT A NAME                              |
| 1260  | 688     | ERROR IN OPENING AN INPUT FILE                                                             |
| 1335  | 733     | ERROR READING AN INPUT FILE                                                                |
| 1345  | 741     | NAME IN POSITION OF A FUNCTION WHICH IS NOT A FUNCTION                                     |
| 1501  | 833     | THE 'CAR' OF AN ATOM HAS BEEN TAKEN                                                        |
| 2125  | 1109    | DIVIDE CHECK IN ZEXPR FOR DIVISION                                                         |
| 2433  | 1306    | BLANK IN A LISP EXPRESSION CANNOT OCCUR                                                    |
| 2505  | 1348    | CLOSING PARENTHESIS CANNOT OCCUR HERE                                                      |
| 2526  | 1365    | BLANK AFTER A NUMBER HAS BEEN FOUND                                                        |
| 3252  | 1706    | BLANK AFTER " " HAS BEEN FOUND                                                             |
| ?     | ?       | WORKING SPACE FULL                                                                         |
| •     | •       | COLLECTOR STACK FULL                                                                       |

SOMETIMES IT CAN BE USEFUL TO KNOW NUMERICALLY THE LOCATION OF OBJECTS. THERE IS NO 'EXPR' FOR THIS BUT IT CAN BE DONE WITH (PLUS(CONS NIL X)) AS 'PLUS' DOES NOT CHECK WHETHER THE ARGUMENT REALLY IS A NUMBER. CONVERSELY (CDR <NUMBER>) WILL CONVERT A NUMBER INTO A POINTER.

ZEXPR(1703 X Y) WILL STORE THE NUMBER Y IN LOCATION X IN FIELD 0.

ZEXPR(1713 X 0) WILL PRINT THE ASCII CHARACTER REPRESENTED IN THE LOW ORDER 8 BITS OF THE AC ON THE LINE PRINTER (I.E. THE DEVICE WHOSE HANDLER IS LOCATED AT 00600). IT RETURNS WITH THE AC CLEAR. USING THIS, ONE CAN PRINT ON THE LINE PRINTER EVEN IF ANOTHER OUTPUT FILE IS OPEN.

ZEXPR(1710 X 0) WILL RETURN THE CONTENTS OF LOCATION X IN FIELD 0 AS ITS VALUE.

ZEXPR (1717 X Y) WILL STORE X AT LOCATION Y IN FIELD 2

ZEXPR (1724 X Y) WILL STORE X AT LOCATION Y IN FIELD 3

ZEXPR (1731 X 0) WILL GET CONTENT OF LOCATION X IN FIELD 2

ZEXPR (1733 X 0) WILL GET CONTENT OF LOCATION X IN FIELD 3

ZEXPR (1735 X Y) WILL RETURN EXCLUSIVE LOGICAL OR OF X AND Y

ZEXPR (1745 X Y) WILL RETURN INCLUSIVE LOGICAL OR OF X AND Y

ZEXPR (1755 X Y) WILL RETURN INCLUSIVE LOGICAL OR OF X AND Y USING MQ OPERATIONS. CAN BE USED ON PDP 8/E OR A PDP 8/I WITH EAE.

ZEXPR (2000 X Y) LISP LEFTSHIFT FUNCTION.  $X*(2^{**}Y)$   
IF  $Y > 0$  SHIFT X LOGICAL LEFT Y STEPS  
IF  $Y < 0$  SHIFT X LOGICAL RIGHT Y STEPS.

ZEXPR (2026 X Y) EAE VERSION OF ABOVE. NEEDS A PDP 8/I OR PDP 8/E WITH EAE. USES SHL/LSR.

ZEXPR (2054 X Y) LISP DIVIDE FUNCTION. RETURNS REMAINDER AND STORES QUOTIENT IN WORD 15 OF FIELD 1, WHERE IT CAN BE RETEIVED BY EXPR (3172 15-1)  
 $X/Y$  TO <15> ;  $X-Y*(X/Y)$  AS FUNCTION VALUE.

ZEXPR (2076 X Y) EAE VERSION OF LISP DIVIDE. FUNCTION. RESULTS AS ABOVE. USES DVI, NEEDS EAE.

ZEXPR (2112 X Y) UNSIGNED MULTIPLICATION, 12 BIT RESULT.  
 $X*Y$  IS RETURNED. USES EAE.

## USEFUL BUILT-IN MACHINE CODE FUNCTIONS

-----

EXPR(3172 X -1) [EXPR(1658 X -1) IN DECIMAL] GIVES THE CONTENTS OF MACHINE ADDRESS X IN FIELD 1. WHEN -1 IS REPLACED BY ANOTHER NUMBER, THIS SERVES AS A MASK.

EXPR(3202 X Y) [EXPR(1666 X Y) IN DECIMAL] WILL STORE THE NUMBER Y IN LOCATION X OF FIELD 1. THIS FUNCTION CAN BE USED TO CHANGE THE SYSTEM AND FORMS THE BASIC INGREDIENT FOR A USER-WRITTEN LAP COMPILER.

## APPLICATIONS:

(EXPR(1666 1374 3584) IN DECIMAL) SWITCHES THE SYSTEM FROM DECIMAL TO OCTAL READING

EXPR(3202 2536 1057) SWITCHES THE SYSTEM FROM OCTAL TO DECIMAL READING.

EXPR(3202 2034 1750) EXPR(3202 2035 144) EXPR(3202 2036 12)  
SWITCHES THE SYSTEM TO DECIMAL PRINTING.

EXPR(3202 2034 1000) EXPR(3202 2035 100) EXPR(3202 2036 10)  
SWITCHES THE SYSTEM TO OCTAL PRINTING.

EXPR(3202 2046 7061) SWITCHES THE SYSTEM TO PRINTING WITH SIGN.

EXPR(3202 2046 7000) SWITCHES THE SYSTEM TO PRINTING WITHOUT SIGN.  
(-1 WILL NOW PRINT AS 7777 IN OCTAL OR 4095 IN DECIMAL.)

EXPR(3174 X Y) WILL RETURN THE LOGICAL 'AND' OPERATION ON X AND Y.

EXPR(3170 X Y) WILL RETURN WITH THE SWITCH REGISTER PLUS X MASKED WITH Y.  
THE LOGICAL INVERSE CAN BE MADE WITH (MINUS -1 X).

EXPR(1306 X Y) WILL RETURN WITH (A SINGLE CHARACTER PLUS X) LOGICAL  
'AND'ED WITH Y FROM THE TAPE READER IN THE CODE AS IT STANDS  
ON THE TAPE.

EXPR(2160 X NIL) WILL PRINT A SINGLE CHARACTER WHOSE VALUE IN INTERNAL  
REPRESENTATION IS X. THIS CAN BE USED FOR OUTPUTTING CHARACTERS  
WHICH OTHERWISE WOULD BE REGARDED AS SYNTACTIC SYMBOLS  
SUCH AS SPACE, "(", OR ")". CARE MUST BE TAKEN, HOWEVER, FOR THE  
LINE COUNT AS CARRIAGE-RETURN NOW COUNTS AS A PRINTABLE  
CHARACTER AND DOES NOT RESET THE LINE COUNT.

THE INTERNAL CODE IS:

0 = SPACE

2 = SPACE

3 = LINE-FEED

6 = CARRIAGE-RETURN

ALL OTHER CHARACTERS IN ASCII FROM 241 TO 335 ARE FOUND BY  
SUBTRACTING 236 OCTALLY, E.G. 25 = "3", 43 = "A", ETC.



THIS RELEASE HAS THE FOLLOWING DIFFERENCES FROM RELEASE 1:

1. CORRECTION MADE FOR  
NIL, 0  
NIL1, 0
2. CORRECTION MADE IN LINE EDITOR FOR ^C TO RESET THE  
TELETYPE BUFFER.
3. ADDITION OF ZEXPR ROUTINE AND FOUR SUPPLEMENTAL ROUTINES  
(ONE TO STORE A VALUE IN FIELD 0, ONE TO LOAD  
A VALUE FROM FIELD 0, ONE TO MOVE A1P, A2P, AND A3P TO  
FIELD 0, AND ONE TO PRINT AN ASCII CHARACTER  
ON THE LINE PRINTER.)

## RELEASE 3 OF LIS

THIS RELEASE HAS THE FOLLOWING DIFFERENCES FROM RELEASE 2:

1. CORRECTION MADE IN GENSYM ROUTINE  
GEN6, NXTA6  
GEN7, GEN4+1
2. ADDITION OF 19 ADDITIONAL ZEXPR ROUTINES  
(LOAD AND STORE IN FIELD 2 AND 3, EXCLUSIVE AND INCLUSIVE OR,  
LEFTSHIFT AND DIVIDE FUNCTIONS WITH AND WITHOUT EAE, A  
BASIC PUNCH ROUTINE, EAE ROUTINES FOR DIVIDE, MULTIPLY AND  
ADD FOR UNSIGNED MULTIPLE WORD ARITHMETIC, SET OCTAL AND  
DECIMAL MODE).
3. THE USE OF LSB/LEB PRINTER OR KLEINSMIDT PRINTER AS  
ALTERNATIVE SPECIAL OUTPUT DEVICE.

ZEXPR (2125 X Y) UNSIGNED MULTIPLICATION, 24 BIT RESULT.  
 $X*Y+<15>$ . MOST SIGNIFICANT 12 BITS TO  $<15>$ ,  
 LEAST SIGNIFICANT 12 BITS AS RETURNED RESULT.  
 THE ROUTINE IS AIMED AT MULTIPLE PRECISION  
 INTEGER ARITHMETIC. USES EAE.  
 $<15>$  IS THE QUOTIENT CELL IN FIELD 1.

ZEXPR (2137 X Y) TWO WORD DIVIDE. ( $<15>,X$ )/Y TO  
 $<15>$  (QUOTIENT), RETURN REMAINDER USES EAE.  
 THE ROUTINE IS AIMED AT MULTIPLE PRECISION  
 INTEGER ARITHMETIC, USES EAE.  
 $<15>$  MUST BE LESS THAN Y.

ZEXPR (2200 X Y) ADDITION WITH CARRY. ( $X+Y+<15>$ ) TO  
 $<15>$  (CARRY), RESULT. THE ROUTINE IS AIMED AT  
 MULTIPLE PRECISION INTEGER ARITHMETIC.

ZEXPR (2211 X Y) BINARY PUNCH ROUTINE. IF  $Y \leq 0$  THEN PUNCH X  
 AND Y ELSE PUNCH X, -Y TIMES.

ZEXPR (2237 0 0) SET DECIMAL, SIGNED INPUT/OUTPUT MODE.

ZEXPR (2253 0 0) [ZEXPR (1195 0 0) IN DECIMAL] SET OCTAL  
 UNSIGNED INPUT/OUTPUT MODE.

ZEXPR (2307 X Y) ASSEMBLER FUNCTION CPAGE. CHECKS IF X AND X+Y  
 BELONG TO SAME PAGE. IF TRUE THEN RETURN X,  
 ELSE RETURN X+Y MASKED WITH 7600.  
 DEFINE((  
 (CPAGE(LAMBDA(X Y)(COND  
 ((EQ(LOGAND X 7600)(LOGAND(PLUS X Y)7600))X)  
 (T(LOGAND(PLUS X Y)7600)))))

#### ADDING NEW MACHINE LANGUAGE FUNCTIONS

---

ASSEMBLE THE NEW FUNCTIONS INTO BINARY FILES THEN LOAD THEM  
 OVER LISP AND RUN OR SAVE THE RESULTING PROGRAM. (STARTING ADDRESS  
 IS 13000)

#### EXAMPLE:

ASSUME YOU HAVE TWO NEW FUNCTIONS, OVL1.BN AND OVL2.BN

```
.R ABSLDR
*LISP.SV/I
*OVL1,OVL2S
.SAVE SYS LISP;13000
.R LISP
```

ZEXPR (SIMILAR TO EXPR ,EXCEPT USING FIELD 0)  
IS A FUNCTION OF THREE ARGUMENTS. THE ARGUMENTS ARE SUPPOSED TO  
EVALUATE TO NUMERICAL VALUES (UNLESS NOT USED).  
THIS JUMPS TO THE FIELD ZERO MACHINE ADDRESS INDICATED IN THE  
FIRST ARGUMENT WITH THE NUMERICAL VALUE OF THE SECOND ARGUMENT  
IN THE ACCUMULATOR. THE POINTER TO THE SECOND ARGUMENT  
CAN BE FOUND IN FIELD 0 LOCATION 37, AND THE POINTER  
TO THE THIRD ARGUMENT CAN BE FOUND IN FIELD 0 LOCATION 41.

THE MACHINE CODE PROGRAM CAN RETURN TO THE LISP  
SYSTEM WITH THE INSTRUCTION JMP 42 IN CASE  
THAT NO VALUE NEED BE RETURNED (CLEAR AC BEFORE RETURNING).  
IF A NUMERICAL VALUE MUST BE RETURNED, THIS CAN BE DONE  
BY HAVING T VALUE IN THE AC, AND DOING A JMP 44.  
BACK IN LISP, ONE THEN HAS A POINTER TO THAT VALUE  
AS A VALUE.

SPECIAL LOCATIONS

\*\*\*\*\*

- ZEXPR (1717 X Y) WILL STORE X AT LOCATION Y IN FIELD 2  
ZEXPR (1724 X Y) WILL STORE X AT LOCATION Y IN FIELD 3  
ZEXPR (1731 X 0) WILL GET CONTENT OF LOCATION X IN FIELD 2  
ZEXPR (1733 X 0) WILL GET CONTENT OF LOCATION X IN FIELD 3  
ZEXPR (1735 X Y) WILL RETURN EXCLUSIVE LOGICAL OR OF X AND Y  
ZEXPR (1745 X Y) WILL RETURN INCLUSIVE LOGICAL OR OF X AND Y  
ZEXPR (1755 X Y) WILL RETURN INCLUSIVE LOGICAL OR OF X AND Y USING  
MO OPERATIONS. CAN BE USED ON PDP 8/E OR A PDP 8/I  
WITH EAE.
- ZEXPR (2000 X Y) LISP LEFTSHIFT FUNCTION.  $X \ll Y$   
IF  $Y > 0$  SHIFT X LOGICAL LEFT Y STEPS  
IF  $Y < 0$  SHIFT X LOGICAL RIGHT Y STEPS.
- ZEXPR (2026 X Y) EAE VERSION OF ABOVE. NEEDS A PDP 8/I OR PDP 8/E  
WITH EAE. USES SHL/LSR.
- ZEXPR (2054 X Y) LISP DIVIDE FUNCTION. RETURNS REMAINDER AND  
STORES QUOTIENT IN WORD 15 OF FIELD 1, WHERE  
IT CAN BE RETEIVED BY EXPR (3172 15-1)  
 $X/Y$  TO  $\langle 15 \rangle$ ;  $X - Y \cdot (X/Y)$  AS FUNCTION VALUE.
- ZEXPR (2076 X Y) EAE VERSION OF LISP DIVIDE. FUNCTION. RESULTS AS  
ABOVE. USES DVI, NEEDS EAE.
- ZEXPR (2112 X Y) UNSIGNED MULTIPLICATION, 12 BIT RESULT.  
 $X \cdot Y$  IS RETURNED. USES EAE.
- ZEXPR (2125 X Y) UNSIGNED MULTIPLICATION, 24 BIT RESULT.  
 $X \cdot Y \ll 15$ . MOST SIGNIFICANT 12 BITS TO  $\langle 15 \rangle$ ,  
LEAST SIGNIFICANT 12 BITS AS RETURNED RESULT.  
THE ROUTINE IS AIMED AT MULTIPLE PRECISION  
INTEGER ARITHMETIC. USES EAE.  
 $\langle 15 \rangle$  IS THE QUOTIENT CELL IN FIELD 1.
- ZEXPR (2137 X Y) TWO WORD DIVIDE,  $(\langle 15 \rangle, X) / Y$  TO  
 $\langle 15 \rangle$  (QUOTIENT), RETURN REMAINDER USES EAE.  
THE ROUTINE IS AIMED AT MULTIPLE PRECISION  
INTEGER ARITHMETIC, USES EAE.  
 $\langle 15 \rangle$  MUST BE LESS THAN Y.
- ZEXPR (2200 X Y) ADDITION WITH CARRY.  $(X + Y \ll 15)$  TO  
 $\langle 15 \rangle$  (CARRY), RESULT. THE ROUTINE IS AIMED AT  
MULTIPLE PRECISION INTEGER ARITHMETIC.



\*\*\*\*\* \*\* \*\*\*\*\*

SOMETIMES A LISP USER HAS SPECIALIZED FUNCTIONS WHICH HE EXPECTS TO PERFORM A NUMBER OF TIMES. BECAUSE OF MEMORY LIMITATIONS AND TIME LIMITATIONS, IT IS OFTEN DESIRABLE TO IMPLEMENT THESE FUNCTIONS IN MACHINE LANGUAGE, AND CALL THEM FROM A LISP PROGRAM. SINCE THE LIST SPACE COULD NOT BE EXTENDED TO A WHOLE FLD (BECAUSE OF DESIGN DECISIONS WHEN LISP 102A WAS WRITTEN) A LARGE PORTION OF FIELD 0 IS NOW UNUSED (LOCATIONS 0-33, 47-77, 2400-5400).

A NEW FUNCTION HAS BEEN ADDED TO LISP CALLED ZEXPR, WHICH HAS 3 ARGUMENTS.

THE USER CAN WRITE HIS OWN FUNCTIONS IN PAL8 WHICH USE THE FREE SPACE IN FIELD 0.

AFTER ASSEMBLING THESE PROGRAMS, HE CAN THEN USE ABSLDR TO LOAD THEM OVER LISP, AND SAVE THE NEW PROGRAM. THEN IN LISP, HE CAN TRANSFER TO THESE ROUTINES USING ZEXPR(W,X,Y).

W IS THE ADDRESS TO TRANSFER TO, X IS THE FIRST ARGUMENT, AND Y IS THE SECOND ARGUMENT.

## EXAMPLE

\*\*\*\*\*

ASSUME THE USER HAS 2 OVERLAYS, OVL1.BN, D OVL2.BN

```
.R ABSLDR
*LISP.SV/I
*OVL1,OVL2$
.SA SYS:LISP;13000
.R LISP
```

THE FOLLOWING IS A COPY OF A SESSION USING LISP.  
THE TEXT WAS TYPED USING EDIT, SINCE EDIT HAS NICER EDITING  
FEATURES THAN LISP.  
IT IS POSSIBLE ALSO TO JUST SAY .R LISP  
AND THEN TYPE IN THE ROUTINES.

```
7EXPR (2211 X Y) BINARY PUNCH ROUTINE. IF Y<=0 THEN PUNCH X
AND Y ELSE PUNCH X, -Y TIMES.
7EXPR (2237 0 0) SET DECIMAL, SIGNED INPUT/OUTPUT MODE.
7EXPR (2253 0 0) (ZEXPR (1195 0 0) IN DECIMAL) SET OCTAL
UNSIGNED INPUT/OUTPUT MODE.
ZEXPR (2307 Y Y) ASSEMBLER FUNCTION CPAGE. CHECKS IF X AND X+Y
BELONG TO SAME PAGE. IF TRUE THEN RETURN X,
ELSE RETURN X+Y MASKED WITH 7600.
DEFINE((
 CPAGE(LAMBDA(X Y)(COND
 ((EQ(LOGAND X 7600)(LOGAND(PLUS X Y)7600)X)
 (T(LOGAND(PLUS X Y)7600))))))
```



#E

.R LISP

IOPEN(DSK SORT LP)

STOP 688 IOPEN

IOPEN(DSK SORT LI)

NIL

(MIN SMALLEST DELETE SORT COMB)

NIL

SORT ((10 9 8 3 7 5 6 1 2 4 ))

(1 2 3 4 5 6 7 8 9 10)

EXIT()

.

```
.R EDIT
*SORT.LI<
```

```
#A
DEFINE((
 (MIN(LAMBDA (X Y) (COND
 ((NULL Y) X)
 ((LESSP X Y) X)
 (T Y))))

 (SMALLEST (LAMBDA (X) (COND
 ((NULL X) NIL)
 (T (MIN (CAR X) (SMALLEST (CDR X)))))))

 (DELETE (LAMBDA (X Y) (COND
 ((NULL X) NIL)
 ((EQ (CAR X) Y) (CDR X))
 (T (CONS (CAR X) (DELETE (CDR X) Y))))))

 (SORT (LAMBDA (X) (COND
 ((NULL X) NIL)
 (T (COMB X (SMALLEST X))))))

 (COMB (LAMBDA (X Y)
 (CONS Y (SORT (DELETE X Y)))))))
```

```
ICL\NOSE()
```

```
#L
DEFINE((
 (MIN(LAMBDA (X Y) (COND
 ((NULL Y) X)
 ((LESSP X Y) X)
 (T Y))))

 (SMALLEST (LAMBDA (X) (COND
 ((NULL X) NIL)
 (T (MIN (CAR X) (SMALLEST (CDR X)))))))

 (DELETE (LAMBDA (X Y) (COND
 ((NULL X) NIL)
 ((EQ (CAR X) Y) (CDR X))
 (T (CONS (CAR X) (DELETE (CDR X) Y))))))

 (SORT (LAMBDA (X) (COND
 ((NULL X) NIL)
 (T (COMB X (SMALLEST X))))))

 (COMB (LAMBDA (X Y)
 (CONS Y (SORT (DELETE X Y)))))))
```

```
ICLOSE()
```