N E W S L E T T E R # 3

September 1980

This is the third issue of The Soft Warehouse Newsletter. The newsletters are devoted to bringing our customers up to date on the latest news, updates, and products of The Soft Warehouse. In addition, it provides a medium for the growing community of **muLISP** and **muMATH** users to exchange ideas and application programs. We welcome short articles describing applications for which our products have been used. Before we get down to business, several announcements are in order:

muLISP[tm]/muSTAR-80 and muSIMP/muMATH[tm]-80

The Soft Warehouse is proud to announce the release of the second generation of our products. More than mere revisions, the 1980 versions have been re-written from top (the documentation) to bottom (the machine language). The program storage barrier has been shattered by the incorporation of a **pseudo-code compiler** which produces extremely compact code. Since the compilation and de-compilation process is both fast and automatic, its use is invisible to the user. Thus the interactive nature of muLISP and muSIMP have not been compromised in the interest of efficiency. The luxury of incremental program development in muLISP has been enhanced with the inclusion of the resident **display-oriented editor** called muSTAR. The power of muMATH has also been extended with a powerful functional **limit package** and a series **summation package**.

LIFEBOAT to Distribute muLISP and muMATH

With the addition of muLISP and muMATH to Lifeboat Associates' Software Supermarket[tm], our products will become available to a much wider audience. Lifeboat, known for its ability to deliver software formatted for a wide variety of disk drives, is located at 1651 Third Avenue, N.Y., N.Y., 10028. If you know of anyone who might be interested, please let them know.

SYMSAC '81
ACM Symposium on Symbolic and Algebraic Computation

SIGSAM (ACM's Special Interest Group on Symbolic and Algebraic Manipulation) is sponsoring a conference on the whole range of symbolic math. It will be held from August 5 to 7, 1981 in Snowbird, Utah, U.S.A. Currently they are issuing a call for full papers or extended extracts with a deadline of February 2, 1981. For details write to the Program Chairman, Arthur C. Norman, University of Cambridge, Computer Laboratory, Corn Exchange St., Cambridge CB2 3QG, England.

PICOMATH-80^{tm}

It has always been a goal of The Soft Warehouse to promote the use and appreciation of computer algebra. To this end, we have developed a demonstration program called PICOMATH-80. It is written in BASIC so that it can be implemented on virtually any computer. Although not as powerful or flexible as muMATH, the programs do provide an inexpensive introduction to computer algebra. If you have any acquaintances who could benefit from such a product, they can contact our distributor: **Programma International** at 3400 Wilshire Blvd, Los Angeles, California, 90010. See the upcoming October issue of **BYTE** for an article on a predecessor to PICOMATH, written by David Stoutemyer of The Soft Warehouse.

muLISP and muMATH User Groups

Two of our customers have expressed an interest in forming user groups as a means of enhancing the diaglogue between users of our systems. Possibly a library of useful application programs could be developed and made available to users at a nominal cost. The individuals who expressed an interest are:

Tom Oliver
Blue Hills
Dewey, Arizona 86327, U.S.A.

and:

Nobuyuki Inada
Information Science Laboratory
The Institute of Physical & Chemical Research
Hirosawa 2-1, Wako-shi
Saitama 351, JAPAN

Newsletter Subscription Policy

Generally, copies of The Newsletter included with shipment of a system are not charged against the three (3) free copies promised in the SWH License Agreement. Thus this is the last free newsletter for owners of systems with serial numbers ending in 000 through 136. Those with serial numbers ending in 137 through 216 will receive one more issue. Finally, those with serial numbers ending in 217 through 270 will receive two more issues.

If you would like to extend your subscription for another three issues, please send $5 (U.S.) to The Soft Warehouse, P.O. Box 11174, Honolulu, Hawaii, 96828, U.S.A.

Typographical error in Newsletter #2

An error in the second newsletter was caught by Dr. Greg Whitten of Microsoft because of his appreciation for mathematical symmetry. For those who received the original version of the newsletter, in the listing of the file SIGMA.ALG the second line following the definition for the property ANTIDF, + should read:

```
WHEN EVSUB(EX2,INDET,INDET+1) + EX1 = 0, -EX2  EXIT
```

What was "HALF" doing in my answer?

The constant "HALF" is defined in TRGPOS.ALG as the number 1/2; however, it is not assigned any value in TRGNEG.ALG. Thus if a muMATH system was constructed by loading TRGNEG without TRGPOS, HALF may have turned up in some of your results. For those users who have not already corrected the error, please add the following line near the beginning of TRGNEG.ALG:

```
HALF: 1/2 $
```

Hyperbolic Function Package

The hyperbolic function package listed at the end of the newsletter was contributed by Woon Chan, a graduate student at the University of Hawaii. Using TRGPOS.ALG as a guide, he developed a muMATH package to simplify and manipulate a wide variety of expressions involving the hyperbolic trig functions. It is very interesting to note the one-for-one correspondence between the hyperbolic package and TRGPOS.ALG. A similar package for negative values of HYPEXPD could also be developed using TRGNEG.ALG as a model.

The control variable **HYPEXPD** is used to control the application of various hyperbolic function identities. When it is a positive multiple of 2, hyperbolic tangents, cotangents, secants, and cosecants are replaced my equivalent expressions using only hyperbolic sines and cosines. When a positive multiple of 3, integer powers of SINH and COSH are replaced by multiple angle expressions. When a positive multiple of 5, products of SINH and COSH are replaced by angle sum expressions.

If the control variable **HYPSQ** is a positive integer, then integer powers of COSH are expressed in terms of SINH. Conversely, when a negative integer, powers of SINH are expressed in terms of COSH.

```
% File:  HYPPOS.ALG           9/16/80               The SOFT WAREHOUSE %

% This package was developed by Woon Chan, of the University of Hawaii.%

FUNCTION NEGLT (EX1),
  WHEN SUM (EX1),
    WHEN POSMULT (NUMNUM,2), NEGCOEF (SECOND(EX1)) EXIT EXIT,
  NEGCOEF (EX1),
ENDFUN$

FUNCTION SINH (EX1, EX2),
  WHEN ZERO (EX1), EX1 EXIT,
  WHEN NEGLT (EX1), -SINH(-EX1) EXIT,
  SIMPU ('SINH, EX1),
ENDFUN$

FUNCTION COSH (EX1, EX2),
  WHEN ZERO (EX1), 1 EXIT,
  WHEN NEGLT (EX1), COSH(-EX1) EXIT,
  SIMPU ('COSH,EX1),
ENDFUN$

HYPEXPD: 0$
FLAGS: ADJOIN ('HYPEXPD, FLAGS)&

FUNCTION TANH (EX1),
  WHEN NEGLT (EX1), -TANH(-EX1) EXIT,
  WHEN POSMULT (HYPEXPD,2), SINH(EX1)/COSH(EX1) EXIT,
  SIMPU ('TANH, EX1),
ENDFUN$

FUNCTION COTH (EX1),
  WHEN NEGLT (EX1), -COTH(-EX1) EXIT,
  WHEN POSMULT (HYPEXPD,2), COSH(EX1)/SINH(EX1) EXIT,
  SIMPU ('COTH, EX1),
ENDFUN$

FUNCTION CSCH (EX1),
  WHEN POSMULT (HYPEXPD,2), SINH(EX1)^-1 EXIT,
  SIMPU ('CSCH, EX1),
ENDFUN$

FUNCTION SECH (EX1),
  WHEN POSMULT (HYPEXPD,2), COSH(EX1)^-1 EXIT,
  SIMPU ('SECH, EX1),
ENDFUN$

HYPSQ: 0$

PROPERTY BASE, SINH, FUNCTION (EX1, EX2),
  WHEN POSITIVE(EX1) AND POSMULT(HYPEXPD,3),
       SINH(EX2)^(EX1-2) * (-1/2+COSH(2*EX2)/2) EXIT,
  WHEN LESSER(EX1,-1) AND POSMULT(HYPEXPD,2),
       (SINH(EX2)^-EX1)^-1 EXIT,
  WHEN EX1<0 AND NEGMULT(HYPEXPD,2),
```

```
       CSCH(EX2)^-EX1 EXIT,
  WHEN POSITIVE(EX1) AND NEGATIVE(HYPSQ),
       (-1+COSH(EX2)^2)^QUOTIENT(EX1,2)*SINH(EX2)^MOD(EX1,2) EXIT,
  WHEN LESSER(EX1,-1) AND NEGATIVE(HYPSQ),
       EX1:-EX1,
       ((-1+COSH(EX2)^2)^QUOTIENT(EX1,2)*SINH(EX2)^MOD(EX1,2))^-1 EXIT,
ENDFUN$

PROPERTY BASE, COSH, FUNCTION (EX1, EX2),
  WHEN POSITIVE(EX1) AND POSMULT(HYPEXPD,3),
       COSH(EX2)^(EX1-2) * (1/2+COSH(2*EX2)/2) EXIT,
  WHEN LESSER(EX1,-1) AND POSMULT(HYPEXPD,3),
       (COSH(EX2)^-EX1)^-1 EXIT,
  WHEN EX1<0 AND NEGMULT(HYPEXPD,2),
       SECH(EX2)^-EX1 EXIT,
  WHEN POSITIVE(EX1) AND POSITIVE(HYPSQ),
       (1+SINH(EX2)^2)^QUOTIENT(EX1,2)*COSH(EX2)^MOD(EX1,2) EXIT,
  WHEN LESSER(EX1,-1) AND POSITIVE(HYPSQ),
       EX1: -EX1,
       ((1+SINH(EX2)^2)^QUOTIENT(EX1,2)*COSH(EX2)^MOD(EX1,2))^-1 EXIT,
ENDFUN$

PROPERTY *, SINH, FUNCTION (EX1, EX2),
  WHEN POSMULT (HYPEXPD,5),
    WHEN FIRST(EX1)='SINH,
      EX1: SECOND(EX1),
      COSH(EX1+EX2)/2 - COSH(EX1-EX2)/2 EXIT,
    WHEN FIRST(EX1) = 'COSH,
      EX1: SECOND(EX1),
      SINH(EX1+EX2)/2 + SINH(EX1-EX2)/2 EXIT EXIT,
  WHEN NEGMULT (HYPEXPD,2) AND POWER (EX1),
    WHEN FIRST(SECOND(EX1)) ='COSH AND THIRD(EX1) = -1,
      WHEN EX2= SECOND(SECOND(EX1)), TANH(EX2) EXIT EXIT EXIT,
ENDFUN$

PROPERTY *, COSH, FUNCTION (EX1, EX2),
  WHEN POSMULT (HYPEXPD,5),
    WHEN FIRST (EX1) ='SINH,
      EX1: SECOND(EX1),
      SINH(EX1+EX2)/2 + SINH(EX1-EX2)/2 EXIT,
    WHEN FIRST(EX1) = 'COSH,
    EX1: SECOND(EX1),
    COSH(EX1+EX2)/2 + COSH(EX1-EX2)/2 EXIT EXIT,
  WHEN NEGMULT (HYPEXPD,2) AND POWER (EX1),
    WHEN FIRST(SECOND(EX1)) ='SINH AND THIRD(EX1) = -1,
      WHEN EX2=SECOND(SECOND(EX1)), COTH(EX2) EXIT EXIT EXIT,
ENDFUN$

RDS()$
```

Matrix Determinant Package

The following function computes the determinant of a matrix. Consistent with the definition of ragged arrays, the determinant of an array is 0 if the array has more rows than columns. The determinant of an array that has more columns than rows is the determinant of the array's left square submatrix. This is a useful generalization of determinants.

```
FUNCTION DET (EX1,
  % Local: %  EX2, EX3, EX4, EX5, LEX3),
  WHEN COLMAT(EX1)
      OR ROW(EX1) AND COLMAT(EX1:EX1.IDMAT(LENGTH(REST(EX1)))),
    EX1: REST(EX1),  EX2: EX3: 1,
    LOOP
      LOOP
        WHEN (EX4:REST(FIRST(EX1))) AND NOT ZERO(EX4:FIRST(EX4)),
          EX2: EX2.EX4,
          EX4: EX4 \ ADJOIN('[,RREST(FIRST(EX1))),
        EX1: REST (EX1),
          LOOP
            WHEN ATOM(EX1), EX1: LEX3  EXIT,
            LEX3: ADJOIN (ADJOIN('[,RREST(FIRST(EX1)))
              - SECOND(FIRST(EX1)).EX4, LEX3),
          EX1: REST (EX1),
          ENDLOOP EXIT,
      LEX3: ADJOIN (ADJOIN('[,RREST(FIRST(EX1))), LEX3),
      EX1: REST (EX1),
        EX5: NOT EX5,
        WHEN ATOM(EX1), EX2: 0  EXIT
      ENDLOOP,
      WHEN ATOM(EX1)
        WHEN (MOD(EX3,4) = 3) = EX5, EX2 EXIT,
        -EX2 EXIT,
      EX3: EX3+1, LEX3: FALSE
    ENDLOOP EXIT,
ENDFUN $
```

Improved FCTR function

The negative setting of the control variable EXPBAS in the function FCTR() was found to prevent factorizations similar to the following:

```
X + X^2*Y^2  -->  X * (1+X*Y^2)
```

This transformation failed to occur because the term X^2*Y^2 was first converted to (X*Y)^2, thus disguising the common factor X. The assignment made in function FCTR() to the variable EXPBAS should be changed to +30 instead of -30. The definition of FCTR() occurs near the end of ALGEBRA.ARI.

Radix Base Conversion Program

Often it is necessary to convert between radix bases, especially in assembly language programming. Instead of buying an **in**expensive pocket calculator to do this, why not use your $3000+ computer system! The following function reads numbers in one base and displays the same number in another base. This cycle continues until a non-number is entered. It is defined using the function DEFUN (see Newsletter #2).

```
(DEFUN BASECON (LAMBDA (BASIN BASOUT NUM RADIX)
  (SETQ RADIX (RADIX))
  (LOOP
     (RADIX BASIN)
     (TERPRI)  (PRIN1 NUMBER" IN:  ")  (SETQ NUM (READ))
     ((NOT (NUMBERP NUM)))
     (RADIX BASOUT)
     (PRIN1 NUMBER" OUT: ")  (PRINT NUM) )
  (RADIX RADIX) ))
```

ANIMAL: A Learning Game

The brain-child of Kent Pitman of MIT, **ANIMAL** is a game in which the computer tries to guess an animal thought up by the player. By asking yes-no questions, the program can quickly make a good quess as to what animal was chosen, provided of course that you have "taught" it well. If it fails to guess the correct answer, the program asks the player to enter an appropriate question to extend its knowledge base. The muLISP implementation of ANIMAL has the ability to save the knowledge as a disk file for use in a later session. The following listing of ANIMAL has been compacted in order to save Newsletter space at the expense of readability. The definition for APPEND can be found in UTILITY.LIB. In the function FLUSH, **^M** denotes a control M (i.e. a 0D Hex) in the text.

```
(DEFUN ANIMAL (LAMBDA NIL
  (INSTRUCTIONS)  (TERPRI)  (FLUSH)  (SETQ *BASE* (LOAD-BASE))
  (LINELENGTH 78) (LOOP
    (DISPLAY (QUOTE (Think of an animal and type RETURN "."))) (FLUSH)
    (PLAY-ROUND *BASE*)
    ((NOT (QUERY (QUOTE (Thanks for the game"." Do you want to play
        again ?))))) )
  (SAVE-BASE)
  (DISPLAY (QUOTE (Thanks again"," and do come back !)))  T ))

(DEFUN INSTRUCTIONS (LAMBDA NIL
  ((NOT *NEW*))
  (SETQ *NEW*)
  (DISPLAY (QUOTE (Think of an animal and I'll try to guess what it is"."
      Just answer my questions with a """Y""" or """N""" followed by a
      carriage return "."))) ))
```

```
(DEFUN LOAD-BASE (LAMBDA (ECHO)
  ((AND (QUERY (QUOTE (Do you want to refresh my memory from a previous
      session ?)))  (RDS (QUOTE ANIMAL) (QUOTE BAS)) )
    (PROG1  (READ)  (RDS) ) )
  (QUOTE ((IS IT WARM BLOODED ?) ((DOES IT LIVE IN THE WATER ?)
      (COBRA) (SALMON)) ((DOES IT MILK ITS YOUNG ?) (ROBIN) (DOG)))) ))

(DEFUN PLAY-ROUND (LAMBDA (TREE ANIMAL)  (LOOP
    ((ATOM (CAR TREE))
      ((QUERY (APPEND (QUOTE (I bet it's)) (LIST (@ (CAR TREE)) (CAR TREE)
          (QUOTE ?)))))
      (DISPLAY (QUOTE (I give up"," what animal were you thinking of?
          "(Please" hyphenate multi-word names ")")))
      (SETQ ANIMAL (READ-NOUN))
      (DISPLAY (APPEND (QUOTE (What YES-NO question could I ask to
          distinguish)) (LIST (@ ANIMAL) ANIMAL (QUOTE from)
          (@ (CAR TREE)) (CAR TREE) ?)))
      (TERPRI)  (DISPLACE TREE
          (LIST (READ-QUESTION) (LIST (CAR TREE)) (LIST ANIMAL))))
    ( ((QUERY (CAR TREE)) (SETQ TREE (CADDR TREE)) )
      (SETQ TREE (CADR TREE)) ) ) ))

(DEFUN SAVE-BASE (LAMBDA (PRIN1)
  ((QUERY (QUOTE (Do you want me to remember what you've just taught me?)))
    (WRS (QUOTE ANIMAL) (QUOTE BAS))  (PRINT *BASE*)  (WRS) ) ))

(DEFUN DISPLACE (LAMBDA (LST1 LST2)
  (RPLACA LST1 (CAR LST2))  (RPLACD LST1 (CDR LST2)) ))

(DEFUN READ-QUESTION (LAMBDA (LST)  (LOOP
    ((MEMBER (RATOM) TERMINATORS)  (NCONC LST (LIST (QUOTE ?))) )
    (SETQ LST (NCONC LST (LIST RATOM))) ) ))

(DEFUN READ-NOUN (LAMBDA (NOUN)  (SETQ NOUN (RATOM))  (FLUSH)  NOUN ))

(DEFUN FLUSH (LAMBDA NIL  (LOOP  ((EQ (READCH) (QUOTE "^M"))) ) ))

(DEFUN QUERY (LAMBDA (LST CHAR)  (LOOP
    (DISPLAY LST)  (SETQ CHAR (CAR (UNPACK (RATOM))))  (FLUSH)
    ((OR (EQ CHAR (QUOTE Y)) (EQ CHAR (QUOTE y))))
    ((OR (EQ CHAR (QUOTE N)) (EQ CHAR (QUOTE n))) NIL)
    (DISPLAY (QUOTE (I don't understand"," please re-enter ".")))
    (TERPRI) ) ))

(DEFUN DISPLAY (LAMBDA (LST)  (TERPRI)  (LOOP
    (PRIN1 (CAR LST))  (SETQ LST (CDR LST))
    ((NULL (CDR LST))  (PRIN1 (CAR LST))  (SPACES 2) )
    (SPACES 1) ) ))

(DEFUN @ (LAMBDA (NOUN)
  ((MEMBER (CAR (UNPACK NOUN)) VOWELS) (QUOTE an))
  (QUOTE a) ))

(SETQ TERMINATORS (QUOTE ("." ? !)))
(SETQ VOWELS (QUOTE (A E I O U)))
```