N E W S L E T T E R # 1

November 1979

This is the first edition of **The Soft Warehouse Newsletter**. It will be devoted to bringing our customers up to date on the latest news, updates, and products of The Soft Warehouse. We invite your suggestions and complaints regarding your encounters with either of our software systems. Also we hope to provide a medium for the exchange of programs and ideas concerning both **muLISP-79** and **muSIMP/muMATH-79**. We welcome short contributions or announcements regarding your current application for our products. The content, level and format of this newsletter will evolve according to your desires. Current plans are to publish every 4 months; however, this too is flexible depending on interest.

Our software has been distributed for over 3 months now, and we are pleased to report that we have received word of only two minor problems. One was a conflict with a particular disk operating system BIOS. That problem was resolved by making a minor change to the muSIMP and muLISP interpreter code. The other problem only affects muMATH users having systems with a relatively small amount of memory (i.e. 32K and less). The symptom and cure are described below under the heading "A Non-Recursive MATCH Function".

The Newsletter is divided into two sections corresponding to our products. This issue we are contributing some clarifications, modifications, and extensions which should prove useful.

* * * * * * * T h e m u M A T H e m a t i c i a n * * * * * *

A TAYLOR Series Expansion Function

Here is a simple **muMATH** function which produces an N-degree Taylor series expansion of a sufficiently differentiable expression EXPN about the point X = A:

```
FUNCTION TAYLOR (EXPN, X, A, N,
    % Local: % J, C, ANS, NUMNUM, DENNUM),
  NUMNUM: DENNUM: 30,
  J: ANS: 0,   C: 1,
  LOOP
    ANS: ANS + C * EVSUB (EXPN, X, A),
    WHEN J=N, ANS EXIT,
    EXPN: DIF (EXPN, X),
    J: J + 1,
    C: C * (X-A) / J,
  ENDLOOP,
ENDFUN $

RDS() $
```

..

Use your computer's text editor to create the file TAYLOR.DIF containing the definition of the TAYLOR function given above. Then, after building a muMATH system that includes as a minimum the differentiation package, DIF.ALG, you can load TAYLOR.DIF using the RDS command:

```
RDS (TAYLOR, DIF);
```

After the file has has been read in, you can, for example, find the first 5 terms of the Taylor expansion of e^x about the point x=0, by entering the expression

```
TAYLOR (#E^X, X, 0, 5);
```

muMATH will respond with the series

```
1 + X + X^2/2 + X^3/6 + X^4/24 + X^5/120
```

A muSIMP-79 WRITE Command

Although the **muSIMP** WRS function can be used to direct output to a disk file, standard **muSIMP** output includes the prompt and herald, while suppressing all but the result of an assignment. Thus, WRS alone is fine for saving output for subsequent display or printing, but it is not sufficient for saving the values of variables for use at a later time. What is needed is a WRITE command which, when used in the form

```
WRITE (fname, ftype, var1, var2, ... varN);
```

generates a file named "fname" of type "ftype" on the current drive containing the successive muSIMP command statements:

```
var1:  value1;
var2:  value2;
 .        .
 .        .
 .        .
varN:  valueN;
RDS ();
```

where each value is that of the corresponding variable. The following is a muSIMP definition for the subroutine WRITE which can be transcribed using the text editor supplied with your computer's operating system:

```
SUBROUTINE WRITE LEX1,
  WRITE1 (FIRST(LEX1), SECOND(LEX1), RREST(LEX1))
ENDSUB$
```

```
FUNCTION WRITE1 (FNAME, FTYPE, LEX1),
  WHEN NOT WRS (FNAME, FTYPE),
    PRINTLINE ("Improper file name or type"),
    FALSE EXIT,
  LOOP
    WHEN EMPTY (LEX1) EXIT,
    PRINT (FIRST (LEX1)),
    PRINT (": "),
    PRTMATH (EVAL (FIRST (LEX1)), 0, 0, TRUE),
    PRINT (';),  NEWLINE (),  NEWLINE (),
    LEX1 : REST (LEX1),
  ENDLOOP,
  PRINTLINE ("RDS ();"),
  WRS (),
  TRUE,
ENDFUN $

RDS () $
```

After a file has been generated by using the WRITE command, the assignments to the variables can be re-established with an RDS command of the form:

```
RDS (fname, ftype, drive);
```

A RANDOM Number Generator

The following pseudo-random number generator is designed according to the principles described by Knuth in "The Art of Computer Programming, Vol. 2" (Addison-Wesley). The user should initialize the global variable named SEED to any arbitrary integer, such as one derived from the time of day. Then, each successive invoking of RANDOM () yields the next number in a sequence which is uniformly distributed over the range 0 through 9999990, inclusive:

```
FUNCTION RANDOM (),
   SEED:  MOD (2113233 + 314157*SEED, 9999991)
ENDFUN $
```

A Non-Recursive MATCH Function

Although valid, the recursive definition of the function MATCH (line 266 in the 7/27/79 version of file MUSMORE.MUS) can cause a stack overflow to occur while parsing in some long function definitions or mathematical expressions, especially for relatively small systems (i.e. 32K and less). Thus it should be replaced with the following iterative definition:

```
FUNCTION MATCH (DELIM, LEX1),
  LEX1: FALSE,
  LOOP
    BLOCK
      WHEN SCAN = COMMA,
```

```
      SCAN () EXIT,
    ENDBLOCK,
    WHEN SCAN = DELIM,
      SCAN (),
      LEX1 EXIT,
    WHEN DELIMITER (),
      SYNTAX (DELIM, "NOT FOUND") EXIT,
    LEX1: CONCATEN (LEX1, LIST (PARSE(SCAN,0))),
  ENDLOOP,
ENDFUN $
```

A DRIVER Function Mod to Save Node Space

The following change to the DRIVER Function defined around line 160 in file MUSMORE.MUS will free a number of nodes while parsing an expre-sion:

```
ENDBLOCK,
EX1: FALSE                %This is the new line to add%
EX1: PARSE (SCAN(), 0),
EX2: SCAN,
```

Simplified MATRIX and EQUATION Operators

It is easy to forget and use ^ rather than " ^ " with blanks and quotes to indicate **muMATH** matrix powers, including inverses. Thus, the following replacement for "Optional Matrix Power & Inverse Package" in the 7/16/79 version of file MATRIX.ARR makes ^ work for matrix powers, at the (admittedly minor) expense of no longer indicating elementwise powers for the elements of arrays:

```
PROPERTY BASE, [, FUNCTION (EX1, LEX1),
  (ADJOIN ('[, LEX1) . IDMAT (LENGTH (LEX1))) ^ EX1
ENDFUN $

PROPERTY BASE, {, FUNCTION (EX1, LEX1),
  WHEN EX1 = -1,
    ADJOIN ('{, LEX1) \ IDMAT (LENGTH (LEX1)) EXIT,
  WHEN ZERO (EX1),
    IDMAT (LENGTH (LEX1)) EXIT,
  WHEN EX1 = 1,
    ADJOIN ('{, LEX1) EXIT,
  WHEN POSITIVE (EX1),
    ADJOIN ('{, LEX1) . ADJOIN ('{, LEX1)^(EX1-1) EXIT,
  WHEN NEGATIVE (EX1),
    (ADJOIN ('{, LEX1) ^ -1) ^ -EX1 EXIT,
  LIST ('^, ADJOIN ('{, LEX1), EX1),
ENDFUN $
```

Naturally, file MATRIX.DOC should be modified accordingly.

The DEAR ALGY Column

> **DEAR ALGY:** Although RECLAIM () indicated sufficient space for a condensed version of a file, I ran out of space while trying to RDS it with CONDENSE TRUE. -- **Perplexed**

DEAR PERPLEXED: For each function definition or property value, the parser generates an uncondensed version, then the condenser (if active) builds a condensed version, after which the uncondensed version can be reclaimed. Thus, temporary space requirements exceed the ultimate requirements. To minimize ths effect move any lengthy function definitions and property values toward the beginning of the file, at the admittedly minor expense of violating our bottom-up ordering style. Alternatively, such long definitions near the end of the file could be subdivided into several definitions. Also it will save space to make the change to the DRIVER function described above.

> **DEAR ALGY:** How come I have sufficient space if I do RECLAIM () before a certain computation but not if I omit the RECLAIM? -- **Baffled**

DEAR BAFFLED: As a programming convenience #ANS is always bound to the last expression computed. If that expression is lengthy, it ties up a lot of node and vector space which would otherwise be free. Thus if #ANS is no longer required and the next expression will also be lengthy, inserting any computation yielding a short #ANS, such as "RECLAIM ()" or "1", frees substantial space.

> **DEAR ALGY:** What is the purpose of STOP near the end of most files? -- **Curious Yellow**

DEAR CURIOUS YELLOW: In the event someone wants to write a function which loads in a sequence of files, the special value "STOP" can be used to regain control before RDS () returns control to the terminal. To date we have not used this feature although we may in the future.

> **DEAR ALGY:** I have only one single-density minifloppy drive. There is not enough room on one diskette for **muSIMP**, the **muMATH** source files and a SYS file. Also, as warned by the operating system manual, the disk directory gets messed up if I change diskettes before making a SYS file, because the write operation is not preceeded by a control-C. To make matters worse, I am unable to save the system using Method "A" described in file READ1ST.TXT because my CP/M will not save that large a file. Am I doomed to building **muMATH** every time I want to use it? -- **Disheartened**

DEAR DISHEARTENED: A one-drive system is masochistic, especially if it is a single-density single-sided minifloppy. I strongly recommend buying a second drive if you value your time and peace of

mind. However, if you wish to operate under such trying circumstances, here is how you can generate a SYS file:

1. Save a copy of MUSIMP79.COM on a fresh diskette.
2. Build a muMATH System as describe in READ1ST.TXT up to the point of saving the System.
3. Remove the diskette with the source files on it and mount the fresh diskette.
4. Type a Control-C which will return control to the DOS.
5. Using the front panel switches or a resident monitor program, start execution beginning at location 100H.
6. The system should respond with the muSIMP logon prompt.
7. Execute the muSIMP SAVE Command as described in the file READ1ST.TXT.

* * * * * * * * * * **T h e m u L I S P e r** * * * * * * * * * *

M E T A M I N D

METAMIND is a computer version of the **Master Mind** (c) game produced by INVICTA. It is a good test of your logical abilities. The object is to break a color code made up by your opponent. The biggest problem in playing the game is finding someone to make the code and doing the rather tedious chore of providing you hints. The following program generates such a code and provides the requisite hints. It also contains several useful subroutines such as the random number generator which may prove helpful in other applications.

```
(SET ECHO)
(PUTD DRIVER (QUOTE (LAMBDA (RDS WRS)
  (LOOP
    (APPLY (READ) (READ)) ) )))
(DRIVER TRUE)

PUTD (METAMIND (LAMBDA (KEYLIST SEED)
  (TERPRI) (TERPRI) (SPACES 18)
  (PRINT "Welcome to METAMIND!!_") (TERPRI) (TERPRI)
  (SETQ KEYLIST (QUOTE (
    (BLU GRN WHI YEL RED BLK)
    (RED YEL GRN BLK BLU WHI)
    (BLK BLU YEL GRN WHI RED)
    (YEL WHI RED BLK BLU GRN)
      )))
  (LOOP
    (PRIN1 "Please enter any random number between 1 and 100:  ")
    (SETQ SEED (RATOM))
    (TERPRI)
    ((PLUSP SEED)) )
  (TERPRI)
  (PRINT "Let me think of a code.")
  (TERPRI) (RECLAIM) (RECLAIM)
```

```
    (PRINT "Ok, I have got one, now make a guess by typing in 4")
    (PRINT "of the colors in the following list:")
    (SPACES 10)  (PRINT (CAR KEYLIST))
    (PRINT "Then after you type a carriage return, I will tell you")
    (PRINT "the number of blacks (i.e. the number of guesses of the")
    (PRINT "right color and right column), a space, and the number")
    (PRINT "of whites (i.e. of the remaining non-black guesses, the")
    (PRINT "number of correct colors).")
    (TERPRI)
    (LOOP
      (CODEMAKER)
      (TERPRI)
      (TERPRI) ) ))

  PUTD (CODEMAKER (LAMBDA (CODE MOVE CTR)
    (SETQ KEYLIST (MAPLIST KEYLIST PERMUTE))
    (SETQ CODE (MKCODE KEYLIST))
    (SETQ CTR 1)
    (LOOP
      (SPACES 8) (PRIN1 "Move: ") (PRIN1 CTR) (SPACES 4)
      ((CODEMATCH CODE (READMOVE CODE) 0)
        (TERPRI) (PRIN1 "That took ") (PRIN1 CTR)
        ( ((EQ CTR 1)
     (PRINT " move.") )
   (PRINT " moves.") )
        ((LESSP CTR 6)
   (PRINT "Hey you're good, let's play again!") )
        ((LESSP CTR 8)
(PRINT "That was a hard one, want to try to improve your score?") )
        (PRINT "Wow, you are lousy, better stick to chess.") )
      (TERPRI)
      (SETQ CTR (ADD1 CTR)) ) ))

  PUTD (MAPLIST (LAMBDA (LST FUN)
    ((NULL LST) NIL)
    (CONS (FUN (CAR LST)) (MAPLIST (CDR LST) FUN)) ))

  PUTD (PERMUTE (LAMBDA (LST1 LST2 LST3)
    ((NULL LST1)
      (NCONC LST2 LST3) )
    ((NULL (CDR LST1))
      (NCONC (CONS (CAR LST1) LST3) LST2) )
    ((NULL (CDDR LST1))
      (NCONC (PERMUTE (CONS (CAR LST1) LST2))
             (PERMUTE (CONS (CADR LST1) LST3))) )
    (PERMUTE (CDDDR LST1) (CONS (CADR LST1) LST3)
    (CONS (CADDR LST1) (CONS (CAR LST1) LST2))) ))

  PUTD (MKCODE (LAMBDA (KEYLST)
    ((NULL KEYLST) NIL)
    (CONS (NTH (CAR KEYLST) (PLUS (RANDOM) 1))
      (MKCODE (CDR KEYLST))) ))

  PUTD (CODEMATCH (LAMBDA (CODE1 MOVE1 BLACKS CODE2 MOVE2)
    ((NULL CODE1)
```

```
    (SPACES 40)
    ((EQ (PRIN1 BLACKS) 4))
    (SPACES 2)
    (SAMETYPE CODE2 MOVE2 0)
    NIL )
  ((EQ (CAR CODE1) (CAR MOVE1))
    (CODEMATCH (CDR CODE1)(CDR MOVE1)(ADD1 BLACKS) CODE2 MOVE2) )
  (CODEMATCH(CDR CODE1)(CDR MOVE1) BLACKS (CONS(CAR CODE1)CODE2)
    (CONS (CAR MOVE1) MOVE2)) ))

PUTD (SAMETYPE (LAMBDA (CODE MOVE WHITES)
  ((NULL CODE)
    (PRINT WHITES) )
  ((MEMBER (CAR CODE) MOVE)
    (SAMETYPE(CDR CODE)(REMBER1(CAR CODE)MOVE) (ADD1 WHITES)) )
  (SAMETYPE (CDR CODE) MOVE WHITES) ))

PUTD (READMOVE (LAMBDA (CODE)
  ((NULL CODE) NIL)
  (CONS (RATOM) (READMOVE (CDR CODE))) ))

PUTD (ADD1 (LAMBDA (X)
  (PLUS X 1) ))

PUTD (REMBER1 (LAMBDA (X L)
  ((NULL L) NIL)
  ((EQ X (CAR L)) (CDR L))
  (CONS (CAR L) (REMBER1 X (CDR L))) ))

PUTD (NTH (LAMBDA (L N)
  ((NOT (PLUSP N)) NIL)
  (LOOP
    ((EQ N 1) (CAR L))
    (SETQ N (DIFFERENCE N 1))
    (SETQ L (CDR L)) ) ))

PUTD (RANDOM (LAMBDA ()
  (SETQ SEED(REMAINDER(PLUS 2113233(TIMES SEED 271821))9999991))
  (QUOTIENT SEED 2499998) ))

PUTD (DRIVER (LAMBDA (RDS WRS)
  (LOOP
    (TERPRI)
    (PRIN1 "$ ")
    (PRINT (APPLY (READ) (READ))) ) ))

LOOP ( (SETQ RDS) (METAMIND) )
```

Save the program as a text file named METAMIND.LIB and then it can be read in with the muLISP command: (RDS METAMIND LIB). There are many inhancements which could be made such as allowing the player to choose how many colors and how many columns are in the code. For the next issue of Newsletter we will publish a program for reversing the roles and making the computer break your code!