This program is an update of the program created by Erik Swanson in 2018. I gratefully acknowledge his work in creating the original program.

UPDATES

- All average atomic masses for naturally occurring elements are expressed to four decimal places.
- There was an issue where the original program could not correctly parse a formula if a two-letter symbol was immediately followed by another symbol. That has been corrected
- The original program also made calls to external programs during the calculation. Those external programs have now been embedded within the main program.

INSTALLATION

To install this program, simply copy and paste the '***Molar_Mass.hpprgm***' file into the "*Programs*" directory on your HP Prime.

To find the molar mass of a chemical compound X, type in: **MM("X")**
[Unfortunately, due to the way that the HP Prime differentiates strings from variables, you must enclose the chemical formula with double quotes ""]

The calculator will then return a molar mass for that compound in units of grams per mole. Fortunately the units will NOT be included with the result, as was the case with the HP48 series version of the same program. This will make it easier to integrate the result directly into subsequent calculations.

EXAMPLES

MM("CH3CH2COOH")	74.0783
MM("NaAl(OH)4")	118.0004
MM("Mg3(PO4)2")	262.8578
MM("NaOCl")	74.4421

CODE [// WITH DESCRIPTIONS]

The source code is included below. I have included comments within the code using "//". You can safely remove all lines beginning with // from this code.

```
EXPORT MM(A)
BEGIN
//
// The following lines of code load the average atomic mass of the elements as temporary variables. The LOCAL command
// can only load a maximum of 8 variables per command, which is why there are so many lines of LOCAL here.
//
 LOCAL H:=1.0079,He:=4.0026,Li:=6.941,Be:=9.0122,B:=10.811,C:=12.0107,N:=14.0067,O:=15.9994;
 LOCAL F:=18.9984,Ne:=20.1797,Na:=22.9897,Mg:=24.305,Al:=26.9815,Si:=28.0855,P:=30.9738,S:=32.065;
 LOCAL Cl:=35.453,Ar:=39.948,K:=39.0983,Ca:=40.078,Sc:=44.9559,Ti:=47.867,V:=50.9415,Cr:=51.9961;
 LOCAL Mn:=54.938,Fe:=55.845,Co:=58.9332,Ni:=58.6934,Cu:=63.546,Zn:=65.39,Ga:=69.723,Ge:=72.64;
 LOCAL As:=74.9216,Se:=78.96,Br:=79.904,Kr:=83.80,Rb:=85.4678,Sr:=87.62,Y:=88.9059,Zr:=91.224;
 LOCAL Nb:=92.9064,Mo:=95.94,Tc:=98,Ru:=101.07,Rh:=102.9055,Pd:=106.42,Ag:=107.8682,Cd:=112.411;
 LOCAL In:=114.818,Sn:=118.71,Sb:=121.76,Te:=127.6,I:=126.91,Xe:=131.293,Cs:=132.9055,Ba:=137.327;
 LOCAL La:=138.9055,Ce:=140.116,Pr:=140.90771,Nd:=144.24,Pm:=145,Sm:=150.36,Eu:=151.964,Gd:=157.25;
 LOCAL Tb:=158.9253,Dy:=162.5,Ho:=164.9303,Er:=167.259,Tm:=168.9342,Yb:=173.04,Lu:=174.967,Hf:=178.49;
 LOCAL Ta:=180.9479,W:=183.84,Re:=186.207,Os:=190.23,Ir:=192.217,Pt:=195.078,Au:=196.9665,Hg:=200.59;
 LOCAL Tl:=204.3833,Pb:=207.2,Bi:=208.9804,Po:=209,At:=210,Rn:=222,Fr:=223,Ra:=226.0254;
 LOCAL Ac:=227.0278,Th:=232.0381,Pa:=231.0359,U:=238.0289,Np:=237.0482,Pu:=244,Am:=243,Cm:=247;
 LOCAL Bk:=247,Cf:=251,Es:=252,Fm:=257,Md:=258,No:=259,Lr:=266,Rf:=267;
 LOCAL Db:=268,Sg:=269,Bh:=270,Mt:=278,Ds:=281,Rg:=282,Cn:=285,Nh:=286;
 LOCAL Fl:=289,Mc:=290,Lv:=293,Ts:=294,Og:=294;
//
// This last LOCAL command is creating the following variables:
//   D = The number of characters in the formula
//   inc = the increment number for the current iteration of the following command sequence
//   a,b = starting from the last two characters in the formula, the IF-THEN-ELSE commands compare the two characters to determine
//       which mathematical operator to INSERT into the formula
//
 LOCAL a,b,inc,D:=DIM(A);
  FOR inc FROM 1 TO D-1 DO
   a:=D-inc;
   b:=D+1-inc;
//
// Check if the last two characters in the formula are letters
//
   IF (ISNUM(CHAR(A[a])) AND ISALPH(CHAR(A[b]))) THEN
    A:=INSERT(A,b,"+");
//
// Check if the last character in the formula is a number but the 2nd to last is a letter
//
   ELSE IF (ISNUM(CHAR(A[b])) AND ISALPH(CHAR(A[a]))) THEN
    A:=INSERT(A,b,"*");
//
// Check if the last two characters are both uppercase letters
//
   ELSE IF (ISCAP(CHAR(A[b])) AND ISCAP(CHAR(A[a]))) THEN
    A:=INSERT(A,b,"+");
//
// Check if the last character is uppercase, but the 2nd to last character is lowercase
//
   ELSE IF (ISCAP(CHAR(A[b])) AND ISLOW(CHAR(A[a]))) THEN
    A:=INSERT(A,b,"+");
//
// Check if the 2nd to last character is a closed bracket and the character after it is NOT a number
//
   ELSE IF (A[a]==41 AND (NOT ISNUM(CHAR(A[b])))) THEN
    A:=INSERT(A,b,"+");
```

```
//
// Check if the last character is an open bracket
//
    ELSE IF (A[b]==40) THEN
      A:=INSERT(A,b,"+"); END; END; END; END; END; END;
    END;
//
// The program will now move one character to the left in the formula and repeat the series of checks.
// At the end of this iteration, the program has inserted either a plus '+' or a multiply '*' into
// the formula between the chemical symbols and turned it into a mathematical expression.
// The following command will then evaluate that mathematical expression using the stored atomic mass
//  variables and return the calculated result as a number.
//
  RETURN EXPR(A);
END;
//
// This procedure determines whether or not a character is a letter
//
EXPORT ISALPH(A)
BEGIN
 LOCAL L:=CONCAT(MAKELIST(X,X,65,90,1),MAKELIST(X,X,97,122,1));
  IF SIZE(INTERSECT(ASC(A),L))==0 THEN
    RETURN 0;
  ELSE
    RETURN 1;
  END;
END;
//
// This procedure determines if the letter is uppercase
//
EXPORT ISCAP(A)
BEGIN
 LOCAL L:=MAKELIST(X,X,65,90,1);
  IF SIZE(INTERSECT(ASC(A),L))==0 THEN
    RETURN 0;
  ELSE
    RETURN 1;
  END;
END;
//
// This procedure determines if the letter is lowercase
//
EXPORT ISLOW(A)
BEGIN
 LOCAL L:=MAKELIST(X,X,97,122,1);
  IF SIZE(INTERSECT(ASC(A),L))==0 THEN
    RETURN 0;
  ELSE
    RETURN 1;
  END;
END;
//
// This procedure determines whether or not a character is a number
//
EXPORT ISNUM(A)
BEGIN
 LOCAL L:=MAKELIST(X,X,48,57,1);
  IF SIZE(INTERSECT(ASC(A),L))==0 THEN
    RETURN 0;
  ELSE
    RETURN 1;
  END;
END;
```