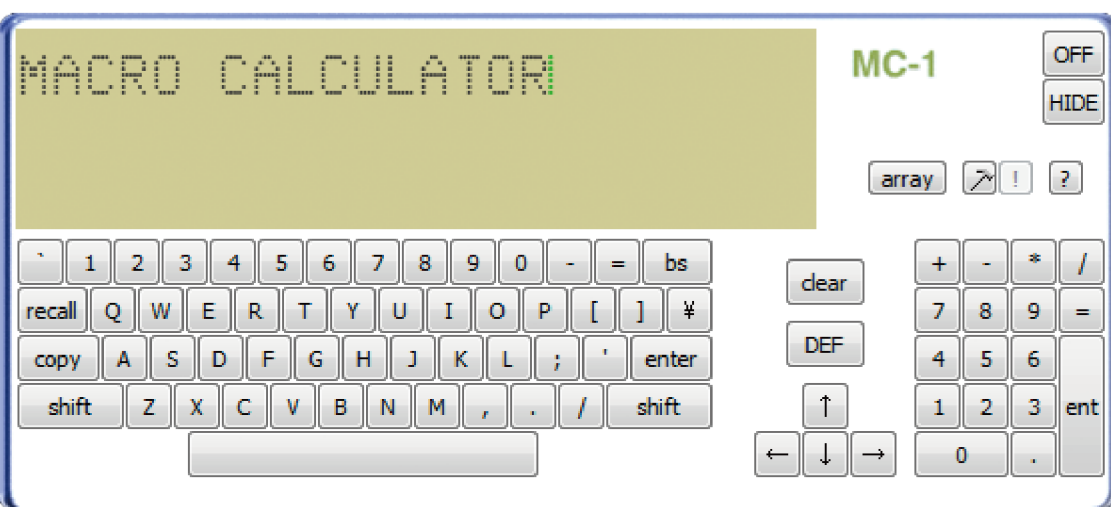


マクロ電卓

# MC-1

## ユーザズマニュアル



【ご注意】本ソフトウェア開発者は、本ソフトウェアの使用に際して生じる金銭上の損害などについては、一切の責任を負いかねますのであらかじめご了承ください。

# 目次

1. 主な特徴.....	1
2. インストール.....	2
3. 各部の機能.....	3
4. 基本事項.....	6
5. 通常計算モード.....	11
6. マクロ定義モード.....	13
7. エラーメッセージ.....	17
8. 設定画面.....	18
9. 配列設定画面.....	20
10. マクロ例題.....	21
11. 技術仕様.....	27



# 1. 主な特徴

本ソフトウェアには次のような特徴があります.

- ・ パソコンデスクトップ上で使用できる電卓です.
- ・ 三角関数や対数などの数学関数が使用できます.
- ・ 数式通りのキー入力で計算を行ないます.
- ・ 変数をメモリーとして使用できます.
- ・ 複数の数式を登録しておくことができます. (マクロ計算)
- ・ 変数値や登録した数式はパソコンのオンオフに関係なく記憶されます.
- ・ ソフトウェアキーボードだけでなくハードウェアキーボードでも使えます.

上記のような特徴を持っていることから, 設計や研究の現場で関数電卓やプログラム電卓と同様の感覚で使用していただけるものと思います.

## V1.1 の新しい機能

- ・ 一次元配列変数が使えるようになりました.

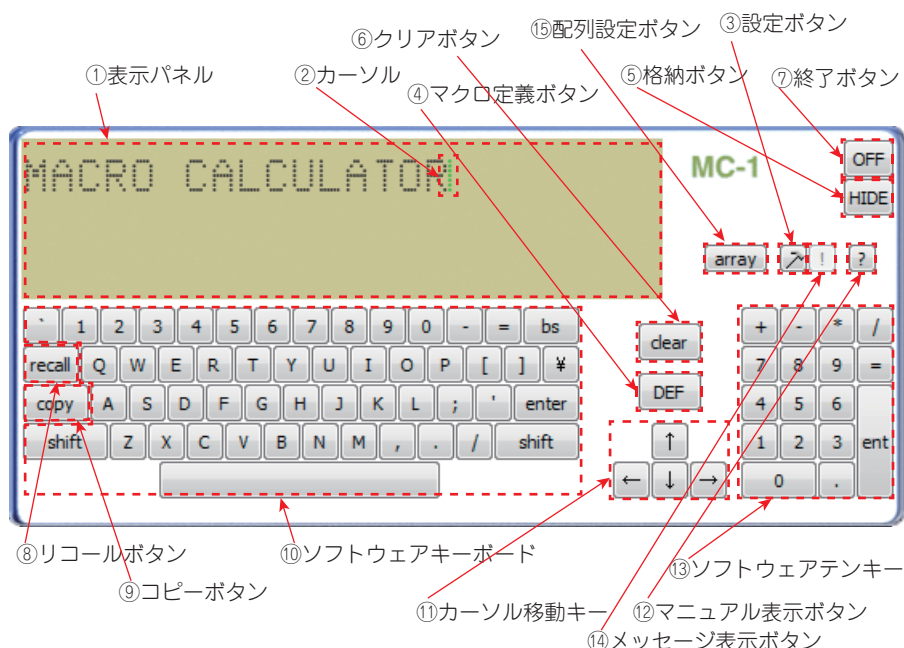
## 2. インストール

本ソフトウェアにインストーラーはありません。自己解凍ファイルを展開してできた「MC-1」フォルダーをハードディスク内のお好きな場所にコピーしてください。

起動するには「MC-1」フォルダー内の CALC.exe をダブルクリックしてください。

**参考** CALC.exe のショートカットをデスクトップに作成しておくとキーボード操作だけで起動することもできるようになります。

### 3. 各部の機能



#### ①表示パネル

数式や結果を表示する部分です。左上には状態によっていくつかの記号が表示されます。

① パネルの左側に数式の一部分が隠れている場合に表示されます。

② パネルの右側に数式の一部分が隠れている場合に表示されます。

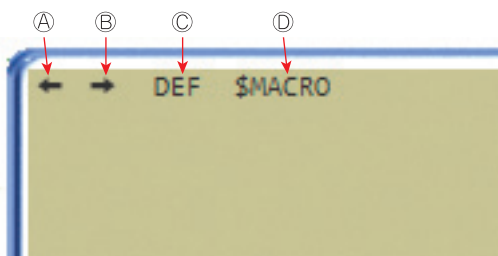
③ マクロ定義モードになっている場合に表示されます。

④ マクロ定義モードで編集のマクロ名が表示されます。マクロ編集モードであっても編集するマクロ名が決まっていな場合は表示されません。

マクロ定義モードについては 13 ページの「6. マクロ定義モード」を参照してください。

#### ②カーソル

数式の編集位置を緑色の縦棒（カーソル）で示します。打ち込んだ文字はカーソ



ル位置に挿入されます。

### ③設定ボタン

計算結果の表示桁数などを設定する画面を表示します。

### ④マクロ定義ボタン

マクロ定義モードと通常計算モードの切り替えを行ないます。マクロ定義モードになると表示パネルにⒸが表示されます。

### ⑤格納ボタン


本ソフトウェアを起動状態のまま Windows タスクバーに格納します。タスクバー内のアイコンをクリックするともとの状態に戻ります。

### ⑥クリアボタン

表示パネルの内容をクリアします。変数値やマクロ定義はそのまま保持されます。Shift キーを押した状態でこのボタンをクリックすると記憶内容をリセットします。リセットでは変数値の 0 クリアとマクロ定義の全消去が行なわれます。

### ⑦終了ボタン

本ソフトウェアを終了します。変数値とマクロ定義はすべて自動的に保存されて次の起動時に引き続き使用できます。

 **参考** 本ソフトウェアを終了し忘れたまま Windows をシャットダウンしてしまった場合も変数値とマクロ定義は正しく保存されます。

なお、マクロ評価時に限りこのボタンがソフトウェアの終了ではなく実行停止ボタンとして機能することがあります。実行停止機能はマクロの実行に非常に時間がかかってしまう場合（多くの場合「無限ループ」です）に働きます。実行停止機能が働くと表示パネルに「BREAK」という結果が表示されます。

### ⑧リコールボタン

最後に行なった計算結果をカーソル位置に挿入します。数値は常にフォーマットなしで挿入されます。

### ⑨コピーボタン

最後に行なった計算結果をクリップボードにテキスト形式でコピーします。Shift キーを押した状態でこのボタンをクリックすると、クリップボード内のテキストをカーソル位置に挿入（ペースト）します。

### ⑩ソフトウェアキーボード

マウスクリックで使用するキーボードです。ハードウェアキーボードで代用する



ことができます。

#### ⑪カーソル移動キー

カーソル位置を移動します。ハードウェアキーボードのカーソルキーで代用することができます。通常計算モードでは2行以上になることはないのです。上矢印と下矢印は意味を持ちません。また、通常計算モードで計算結果が表示されている状態（カーソルが表示パネル内に表示されていない状態）で左矢印キーを押すと、計算式を編集し直すことができます。

#### ⑫マニュアル表示ボタン

ユーザズマニュアル（本書）を表示します。お使いのパソコンにPDF ファイルを表示するソフトウェアがインストールされていなければなりません。

#### ⑬ソフトウェアテンキー

マウスクリックで使用するテンキーです。ハードウェアテンキーで代用することができます。

#### ⑭メッセージ表示ボタン

エラーが起こったときにエラーの詳細を表示するボタンです。マクロの構文エラーがあった場合に役に立ちます。

#### ⑮配列設定ボタン

配列変数の定義と解除を行なうための画面を表示します。配列は一次元配列のみ使用可能です。

## 4. 基本事項

### 使用できる文字

英大文字, 数字, 空白, 以下の記号

`()-=~!@#\$\$%^&\*~+[]{}|\;:~",./<>?

### 数式中の空白

空白はどこにあっても意味を持ちません。たとえば HENSUU と HEN SUU はまったく同じ文字列と見なされます。式を読みやすくするために適宜使用してください。

### データ型

8バイトで表現される実数だけを扱います。

### 予約語

「PAI」は円周率  $\pi$  ( $=3.14\dots$ ) を表わす予約語です。その他に予約語はありません。

### 変数

変数には値を代入しておいて数式の中で引用することができます。変数名は予約語以外の任意の名前を付けることができます。名前の長さに制限はありませんが、使用できる文字は先頭は英字のみ、先頭以外は英数字とアンダーバー（`_`）です。初めて引用される変数は必ず0になっています。また、reset ボタンを押すとすべての変数が0になります。

### 配列

配列は変数の一種ですが順序づけられた複数の値を保持していることが特徴です。変数が名前だけで引用されるのに対して、配列は名前と順番を表わす整数を使って引用されます。たとえば配列 ARRAY の5番目の値を引用する場合は ARRAY(5) と表現します。名前 ARRAY を配列名、順番を表わす整数 5 を添字と呼びます。

配列名は予約語以外の任意の名前を付けることができます。名前の長さに制限はありませんが、使用できる文字は先頭は英字のみ、先頭以外は英数字とアンダーバー（`_`）です。変数名と同じ配列名を使用することもできます。このとき変数と配列には何の関係もありません。たとえば変数 `A` と配列要素 `A(1)` は異なる値を保持できます。

添字は 1 から始まる整数です。

配列は計算する前に配列設定画面（20 ページの「9. 配列設定画面」）で作成しておかなければなりません。計算時に動的に定義したり破棄したりすることはできません。配列要素を引用するときに定義した最大要素数を超える要素を引用しようとするエラーになります。

**参考** 変数名／配列名として `EXP` や `SIN` のような組み込み関数名を使用することもできます。たとえば `EXP(-1)` は組み込み関数と見なされますが、`EXP-1` は変数を使った引き算と見なされます。とはいえミスを誘発する可能性があるので、特に必要がないかぎり組み込み関数名を変数名や配列名として使用することは避けるべきです。

## 二項演算子

- `+` 足し算
- `-` 引き算
- `*` かけ算
- `/` 割り算
- `^` 乗数が整数であるべき乗
- `**` 乗数が整数でないべき乗
- `=` 等しい場合に 1、等しくない場合に 0
- `~` 両辺を四捨五入した整数値で比較して等しければ 1、そうでなければ 0
- `<` 左辺が右辺より小さければ 1、そうでなければ 0
- `>` 左辺が右辺より大きければ 1、そうでなければ 0
- `<=`, `=<` 左辺が右辺より小さいか等しければ 1、そうでなければ 0
- `>=`, `=>` 左辺が右辺より大きいかわ等しければ 1、そうでなければ 0
- `<>`, `><` 等しくなければ 1、等しければ 0
- `&` 両辺とも 0 以外なら 1、そうでなければ 0
- `|` 両辺とも 0 なら 0、そうでなければ 1

二項演算子の評価の優先順は次のとおりです。

`**`, `^` ▶ `*`, `/` ▶ `+`, `-` ▶ `=`, `~`, `<`, `>`, `<=`, `=<`, `>=`, `=>`, `<>`, `><` ▶ `&`, `|`

同じ優先順の演算子が続いた場合は左側から評価されます。

**参考** べき乗が二種類あるのは少しわかりにくいでしょう。実際のところ他の多くの処理系では見かけ上この区別はありません。しかしながら乗数が整数であるかどうかによって内部の計算手法はまったく異なります。見かけ上区別のない処理系ではこの区別を内部で自動的に行なっているのです。残念ながら本ソフトウェアでは自動判定までは行なっていないのでユーザーによる使い分けをお願いしているということです。たとえば本ソフトウェアでは  $(-3)^2$  はエラーになりませんが  $(-3)**2$  はエラーになります。

## 組み込み関数

組み込み関数は必ず「組み込み関数名（引数）」という形式で引用します。本ソフトウェアは組み込み関数名と変数名の区別をそれに続く左括弧で認識するからです。次の組み込み関数が使用できます。（角度の単位はすべてラジアン）

ABS(x)	x の絶対値
ACOS(x)	x の逆余弦
AINT(x)	x の小数部切り捨て
ANINT(x)	x の小数部四捨五入
ASIN(x)	x の逆正弦
ATAN(x)	x の逆正接
ATAN2(x,y)	(x,y) の方向角
COS(x)	x の余弦
COSH(x)	x の双曲線余弦
DIM(x,y)	x の y に対する超過分
EXP(x)	x の指数関数
LOG(x)	x の自然対数
LOG10(x)	x の常用対数
MAX(x,y)	x と y の大きい値
MIN(x,y)	x と y の小さい値
MOD(x,y)	x ÷ y の剰余
SIGN(x,y)	x に y の符号を付けた値
SIN(x)	x の正弦
SINH(x)	x の双曲線正弦
SQRT(x)	x の平方根
TAN(x)	x の正接

## 「配列の添字」と「繰り返しの制御変数」

MC-1 では数値は実数型しかありません。しかし整数値が必要になる局面が二つあります。このとき必要な整数を実数で代用するため細心の注意がいります。

### (1) 配列の添字

添字を実数で指定するため、丸め誤差を認識しておく必要があります。たとえば見かけ上 1 と表示される値であっても内部的には 0.999999 とか 1.000001 であると考えておく必要があります。したがって実数で指定された添字がどのように整数値に丸められるかを知っておくことは非常に重要です。

MC-1 では添字は**四捨五入**されます。このため見かけ上 1 と表示される実数は確実に整数値 1 に変換されます。配列の添字に関しては丸め誤差を考慮した四捨五入処理は不要です。

### (2) 繰り返しの制御変数

MC-1 では %WHILE 文を使った繰り返し計算が可能です (15 ページ)。このとき繰り返し回数を制御するために整数値を保持する変数を使うことがあります。この変数のことを**繰り返しの制御変数**と言います。

たとえば 5 回の反復計算を行ないたい場合は変数 I に 0 を代入しておいて一回の計算ごとに I=I+1 を実行します。この直後に I が 5 になっていれば 5 回の反復が終わったことになります。ここで問題になるのが I が 5 になったかどうかの判定です。論理演算子 = を使って I=5 と記述すると実数値の比較になるため、5 回反復直後も丸め誤差分の差で等号が成り立たない可能性があります。そこでこのような整数値の比較の場合は論理演算子 ~ を使って I~5 と記述します。論理演算子 ~ は両辺をそれぞれ四捨五入した整数値に変換し、整数比較を行なう演算子です。これを使うと丸め誤差による等号の不成立を回避できます。

さらに言うなら繰り返しの制御変数への代入文 I=I+1 は I=ANINT(I+1) とするほうが安全です。なぜなら反復回数が非常に多い場合**累積誤差**が無視できなくなることがあるからです (8 バイト IEEE 浮動小数点形式ではほとんど問題にならないとは思いますが)。

#### ●反復計算の中で配列を使う場合のマクロ例

```
I=0
%WHILE:(I~5)^0 # repeat 5 times
  A(I+1)=I
  I=ANINT(I+1)
```

%WEND

## 5. 通常計算モード

表示パネルに DEF マークが表示されていない状態を「通常計算モード」といいます。この状態で次のいずれかを実行できます。

### (1) 1 行で表現できる数式の計算

入力形式： 数式

例	キー操作	表示パネル
$\sqrt{2+3 \times (4-1)}$		SQRT(2+3*(4-1)) 3.3166247903554
$\sin^2 x + \cos^2 x$		SIN(X)^2+ COS(X)^2 1

### (2) 1 行で表現できる数式の計算結果を変数に代入

入力形式： 変数=数式

例	キー操作	表示パネル
$a \leftarrow 12.3 \exp(-3.5)$		A=12.3* EXP(-3.5) 0.371427816094518
$a \leftarrow a + 1$		A=A+1 1.37142781609452

### (3) 定義済みマクロの実行

入力形式： \$ マクロ名

例	キー操作	表示パネル
\$MACRO の実行		\$MACRO (various)

いずれの場合も enter キーを押す時のカーソル位置は特に意味を持ちません。必ず入力全体が評価されます。

結果は表示パネルの最下段に右詰めに表示されます。結果の表示形式は設定ボタン (18 ページの「計算結果の表示形式」) で指定しておくことができます。

## 再編集機能

結果が表示された状態で左矢印キーを押すと入力式が再編集できる状態になります。左矢印キー以外のキーを押した場合は新しい入力の始まりになります。

例	キー操作	表示パネル
$2 + 3 \times 4$ を計算した後、 $2 + 3 \div 4$ に変更して再 計算	$2$ $+$ $3$ $*$ $4$ ENTER $\leftarrow$ $\leftarrow$ BS $\div$ ENTER	$2+3*4$ 14 $2+3*4$ $2+3/4$ 2.75

## リコール機能

入力中に recall ボタンを押すと最後の計算結果がフォーマットなしで入力されます.

例	キー操作	表示パネル
$\frac{\pi}{6}$ を計算した後、その結 $6$ 果を使って $\sin \frac{\pi}{6}$ を計算	$\pi$ $\div$ $6$ ENTER S $\pi$ N ( $\leftarrow$ (RECALL) ) ENTER	$\pi \div 6$ 0.523598775598299 $\sin(0.52359\dots)$ 0.5



## 6. マクロ定義モード

表示パネルに DEF マークが表示されている状態を「マクロ定義モード」といいます。この状態ではマクロの定義だけが可能です。

### マクロとは

マクロは一つ以上の文で構成されます。「通常計算モード」でマクロが引用されると定義された文が一行ずつ評価されていきます。評価の順序は通常定義された順ですが、評価順を制御する文が評価されると前に戻ったり先にジャンプしたりすることもあります。マクロ内で使用できる文の詳細は 15 ページの「マクロ内で使用できる文」で説明します。

### マクロ名

\$ に 1 文字以上の英字名を付け足します。英字名は先頭が英字で 2 文字目以降は英数字とアンダーバー ( `_` ) が使えます。

### マクロの引用結果

「通常計算モード」でマクロを引用すると最後の演算結果が計算結果となります。

```
A=2  
B=3  
C=A+B
```

このマクロの結果は C の値つまり 5 となります。なお、マクロ評価前に設定されていた変数値はそのままマクロ内で引用され、マクロ評価中に値が設定された変数はその後の計算でその値が使用できます（この例では A, B, C はマクロ評価後もそれぞれ 2, 3, 5 を保持します）。したがってマクロで複数の計算結果を得たい場合は変数に結果を代入するように文を記述してください。この場合マクロの実行後に表示パネルに表示される結果はあまり重要ではありません。

### マクロの記述形式

マクロは文を列挙する形で記述します。このとき次の特殊記号が使用できます。  
# この文字以降の文字列は意味のある文とはみなしません。注釈などの記述に適宜使用してください。

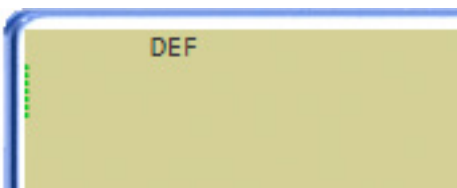
```
# CALCULATION OF TOTAL FRUITS
#
A=2   # NUMBER OF APPLES
B=3   # NUMBER OF ORANGES
C=A+B # TOTAL FRUITS
```

& 行の最後の文字がこの文字の場合、文は次の行に継続しているものとみなします。長い文を読みやすくするために行を分けて記述したい場合に使用します。

```
GAMMA =SQRT(ALPHA^2+BETA^2)* &
      SIN(ALPHA*BETA-PAI/2)+1
```

## マクロ定義操作

マクロ定義モードに移行したばかりの状態では表示パネル上部にマクロ名が表示されません（右図）。これはまず編集するマクロ名を指定する必要があることを表わしています。



マクロ名を指定するには次のいずれかの操作を行ないます。

### (1) 編集するマクロ名を直接打ち込む

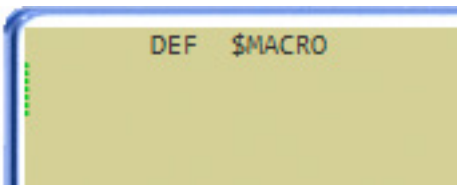
この方法の場合、既存マクロの編集または新規マクロの編集のどちらも可能です。マクロ名を打ち込んだ後 enter キーを押すと表示パネル上部にマクロ名が表示され、マクロの編集が可能になります。

### (2) 定義済みマクロの一覧の中から選択する

この方法は既存マクロの編集が基本になりますから、マクロがまったく定義されていない場合は使えません。まず \$ だけを打ち込んで enter キーを押してください。表示パネルに定義済みマクロの一覧が表示されます。定義マクロがまったくない場合は何も表示されませんので (1) の方法でマクロを指定してください。定義済みマクロが表示されたらカーソルキーを上下矢印キーで動かして編集したいマクロ名の位置に置きます。最後に enter キーを押すとカーソル位置のマクロの編集ができる状態になります。

**参考** マクロ名一覧の中から選択する方法で、enter キーを押す前にカーソル位置のマクロ名を修正することで新規マクロの編集を行なうこともできます。

マクロ名が指定されると右図のように表示パネル上部にマクロ名が表示されます。この状態でマクロの編集を行なってください。



マクロの編集が終わったらマクロ定義ボタン（DEF ボタン）を押して「通常計算モード」に戻ります。これでマクロの定義は終了です。続けてマクロの定義を行ないたい場合はもう一度マクロ定義ボタン（DEF ボタン）を押して再びマクロ定義モードに移行します。

**注意** このように複数のマクロを続けて定義する場合は、一つのマクロの定義ごとにいったん「マクロ定義モード」から「通常計算モード」に移行しなければなりません。

## マクロ内で使用できる文

### ●代入文

**形式** 変数 = 数式

**説明** 数式を評価した結果を変数に代入します。

**文の値** 数式を評価した結果、すなわち変数の値。

### ●条件分岐

**形式** %IF: 条件式 1

複文 1

%ELSEIF: 条件式 2

複文 2

%ELSE

複文 3

%ENDIF

複数記述可能

一つの %IF に対して一回だけ記述可能

[ ] 内はなくてもよい

**説明** 条件式を順に評価して真（0 以外）になった行に続く複文を実行します。真になる条件式がなかった場合は %ELSE があればその行に続く複文を実行します。複文の実行後はいずれの場合も %ENDIF の次の行までスキップします。複文とは文の集まりを表わし、空の場合もあります。

**文の値** 実行された複文の最後の文の値。実行された複文が空の場合は最後の条件式の値が文の値になります。

**制限** 条件分岐を入れ子にすることができますが、その深さは 100 段までです。

### ●繰り返し

**形式** %WHILE: 条件式

複文

## %WEND

**説明** 条件式が真（0 以外）なら複文を実行します。複文の実行後は再び条件式の評価に戻ります。こうして条件式が偽（0）になるまで反復します。条件式が偽になったら %WEND の次の行にスキップします。

**文の値** 常に 0。（最後の条件判定値です。）

**制限** %WHILE ～ %WEND は入れ子にすることができますが、その深さは 100 段までです。

## ●プリント文

**形式** %PRINT: 数式

**説明** 数式を計算した結果をエラーメッセージファイルに出力します。マクロ中の %PRINT 文が実行されると、エラーの有無に関係なくメッセージ表示ボタンが押せるようになります。

メッセージとしてはアドレスと値だけが表示されるのであまり親切な表示ではありません。デバッグなどの目的に限定して使用することをお勧めします。

**文の値** 数式を評価した結果。

## 7. エラーメッセージ

「通常計算モード」の計算で式やマクロに誤りがあったり数値が不適切な場合、エラーが発生することがあります。エラーが発生すると表示パネルの最下行に数値のかわりに次のようなメッセージが表示されます。

ERROR x

エラーの原因によって x には次のようなコードが表示されます。

コード	原因
1	項があるべきところに演算子、あるいは演算子があるべきところに項がある。単なるタイプミスの可能性が高い。
3	演算子用バッファが足りない。式をもう少し単純にして括弧の深さを浅くすると回避できる可能性がある。
4	括弧がアンバランス。右括弧が多すぎる。
5	括弧がアンバランス。左括弧が多すぎる。
6	不正な組み込み関数名がある。
8	組み込み関数の引数の数が間違っている。
12	認識できない文
81	引用されたマクロは存在しない。
82	マクロ内に不正な文がある。メッセージ表示ボタンを押して詳細を確認可能。
M	マクロ定義モードで正しいマクロ名が指定されていない。
F	浮動小数点演算エラー

エラーが発生するとメッセージ表示ボタンが押せる状態になります。メッセージ表示ボタンを押すとエラーの発生箇所がもう少し詳しく表示されます。

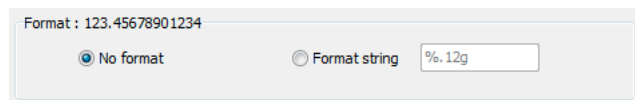
エラーになった場合の対処の例を示します。

例	キー操作	表示パネル
$3.1 \div 0.2$ とするところ を誤って $3.1 \div 0.0$ とし てしまった	 ENTER ← BS 2 ENTER	$3.1/0.0$ ERROR F $3.1/0.0$ $3.1/0.2$ 15.5
$\sqrt{5^2 - 3^2}$ とするところ を誤って $\sqrt{5^2 - -3^2}$ とし てしまった	 ENTER ← ← ← ← BS ENTER	$\text{SQRT}(5^2--3^2)$ ERROR 1 $\text{SQRT}(5^2--3^2)$ $\text{SQRT}(5^2-3^2)$ 4

## 8. 設定画面

設定ボタンを押すと設定画面が表示されます。設定画面では次の設定が行なえます。

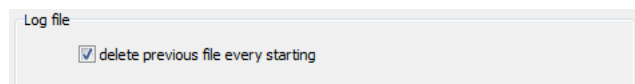
### 計算結果の表示形式



形式指定なし (No format) を選択すると丸めをまったく行なわないで結果を表示します。

形式指定 (Format string) を選択した場合はテキストボックス内に形式指定文字列を入力してください。形式指定文字列は C 言語の printf 書式指定子のうち実数値に対して使える文字列を使用してください。(例. %8.3f, %12.4e など) 実際の表示の例が上部に表示されますので参考にしてください。

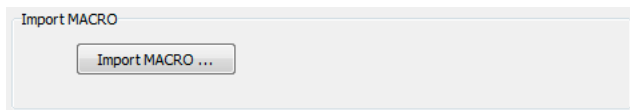
### ログファイルの作成方法



本ソフトウェアは起動してから終了するまでのいくつかの要所で実行内容をログファイルに出力します。ログファイルはお使いのユーザーのドキュメントフォルダー内の「MACRO\_CALCULATOR」というフォルダー内に「MC.LOG」という名前で作成されます。

このチェックボックスをチェックしておくとし起動するたびに前回のログファイルを削除して新たにログファイルを作成します。チェックしていない場合は前回のログファイルの最後から追加する形でログファイルを作成しますので、あまり膨大なログファイルにならないように注意してください。

# マクロの読み込み



## ●外部テキストファイルからマクロを読み込む操作

- (1) Import MACRO... ボタンをクリックします.
- (2) ファイル選択ダイアログが表示されるのでマクロが格納されたファイルを選択します.

## ●読み込み動作について

同じ名前のマクロが存在する場合はそのマクロの内容が置き換えられます. 新たな名前のマクロは追加されます. その他の既存のマクロは何も変更されずに残ります.

設定画面を Cancel ボタンで終了した場合はファイルからの読み込みはすべて破棄されます.

## ●外部ファイルを作成する際の注意

- 拡張子は TXT
- 文字コードは UTF8
- 使える文字は, 英大文字, 数字, 空白, 以下の記号

`()-=~!@#\$%^&\*+[]{}|\:;'"../<>?

(使用できない文字があると \* に変換されます.)

- 内容の形式は次のとおり

\ マクロ名

マクロ

\END

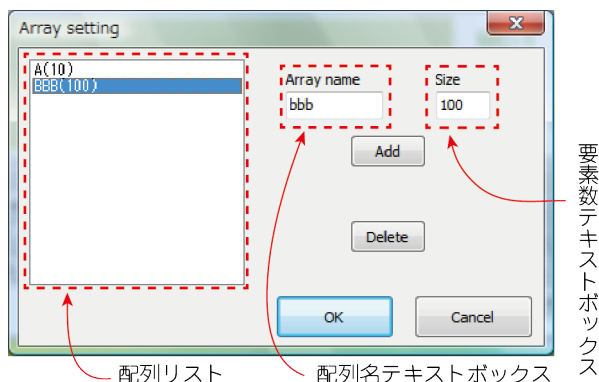
これを繰り返して複数のマクロを記述しておくことができます.

(\ はフォントによっては ¥ と表示される場合があります.)

## 9. 配列設定画面

配列の作成と削除を行なう画面です。配列には次の制限があります。

- 1) 一次元配列のみ使用可能
- 2) 計算を行なう前に配列を作成する必要がある（動的作成は不可）
- 3) 計算時に配列の削除はできない
- 4) 配列名に PAI（予約語）は使用できない



配列リストには現在定義されている配列が表示されています。カッコ内は最大要素数です。

### ●新規配列を定義する方法

- (1) 配列名テキストボックスに配列名を入力します。配列名は PAI 以外の英字名なら何でもよいです。同じ変数名があっても別々に扱われます。
- (2) 要素数テキストボックスに最大要素数を入力します。最大要素数にソフトウェア上の制限はありません。
- (3) Add ボタンをクリックします。配列リストに表示されたら定義完了です。すべての要素の初期値は 0 です。

### ●既存の配列を削除する方法

- (1) 配列リストで削除したい配列をクリックして選択します。
- (2) Delete ボタンをクリックします。配列リストから消えたら削除完了です。



## 10. マクロ例題

簡単なマクロの例を示します。

**参考** ここに示す例はすべて example.txt に入っています。設定画面のマクロの読み込み機能を使って読み込めばすぐに計算してみることができます。

### ● \$ELEMENT (原子量)

原子量は元素周期表などを調べればわかる定数です。このマクロを実行すると「元素記号 \_ELM」という変数に原子量を設定します。したがって、たとえば H<sub>2</sub>O の分子量を知りたければ H\_ELM\*2+O\_ELM という数式で計算できるわけです。このような定数ライブラリーのような使い方としては他に物理定数などにも応用できます。

```
# ELEMENT WEIGHT
H _ELM=1.00794
HE _ELM=4.002602
LI _ELM=6.941
BE _ELM=9.012182
B _ELM=10.811
C _ELM=12.0107
N _ELM=14.0067
O _ELM=15.9994
F _ELM=18.9984032
NE _ELM=20.1797
NA _ELM=22.98976928
MG _ELM=24.3050
AL _ELM=26.9815386
SI _ELM=28.0855
P _ELM=30.973762
S _ELM=32.065
CL _ELM=35.453
AR _ELM=39.948
K _ELM=39.0983
CA _ELM=40.078
SC _ELM=44.955912
TI _ELM=47.867
V _ELM=50.9415
CR _ELM=51.9961
MN _ELM=54.938045
FE _ELM=55.845
CO _ELM=58.933195
NI _ELM=58.6934
CU _ELM=63.546
ZN _ELM=65.38
GA _ELM=69.723
```

GE\_ELM=72.64  
AS\_ELM=74.92160  
SE\_ELM=78.96  
BR\_ELM=79.904  
KR\_ELM=83.798  
RB\_ELM=85.4678  
SR\_ELM=87.62  
Y \_ELM=88.90585  
ZR\_ELM=91.224  
NB\_ELM=92.90638  
MD\_ELM=95.96  
RU\_ELM=101.07  
RH\_ELM=102.90550  
PD\_ELM=106.42  
AG\_ELM=107.8682  
CD\_ELM=112.411  
IN\_ELM=114.818  
SN\_ELM=118.710  
SB\_ELM=121.760  
TE\_ELM=127.60  
I \_ELM=126.90447  
XE\_ELM=131.293  
CS\_ELM=132.9054519  
BA\_ELM=137.327  
LA\_ELM=138.90547  
CE\_ELM=140.116  
PR\_ELM=140.90765  
ND\_ELM=144.242  
SM\_ELM=150.36  
EU\_ELM=151.964  
GD\_ELM=157.25  
TB\_ELM=158.92535  
DY\_ELM=162.500  
HO\_ELM=164.93032  
ER\_ELM=167.259  
TM\_ELM=168.93421  
YB\_ELM=173.054  
LU\_ELM=174.9668  
HF\_ELM=178.49  
TA\_ELM=180.94788  
W \_ELM=183.84  
RE\_ELM=186.207  
OS\_ELM=190.23  
IR\_ELM=192.217  
PT\_ELM=195.084  
AU\_ELM=196.966569  
HG\_ELM=200.59  
TL\_ELM=204.3833  
PB\_ELM=207.2  
BI\_ELM=208.98040

```

TH_ELM=232.03806
PA_ELM=231.03588
U _ELM=238.02891

```

例	キー操作	表示パネル
H2O の分子量を計算します.	<pre> \$ELEMENT ENTER H ELEMENT * 2 + O ELEMENT ENTER </pre>	<pre> \$ELEMENT 238.02891  H_ELM*2+O_ELM 18.01528 </pre>

### ● \$NEWTON（方程式の解）

この例は  $32x^5 - 64x + 31 = 0$  の解をニュートン法で求めるものです。F\_M に 0 になるべき方程式を定義し、FX\_M にその微分式を定義しておけば他の式でも利用できます。

このような例としては他にも逐次代入法や挟み撃ち法、ガウス・ザイデル法などが応用できます。

```

# SOLVE 32*X^5-64*X+31=0
X=0
F_M=32*X^5-64*X+31
%WHILE:ABS(F_M)>1E-6
  FX_M=160*X^4-64
  DX_M=-F_M/FX_M
  X=X+DX_M
  F_M=32*X^5-64*X+31
%WEND
DUMMY_M=X

```

### ● \$PAI（級数を使った $\pi$ の計算）

級数展開を使った計算例です。  $\pi$  の値は atan(1) の級数展開から次のように表わされることが知られています。

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

（ただしこの式は収束が異常に遅いので実用的とはいえません。）これをマクロで計算しようとする次のようになります。

```

# PAI FROM ATAN(1) EXPANSION
I_M=1
P_M=0
S_M=1
DP_M=S_M/(2*I_M-1)
%WHILE:ABS(DP_M)>1E-6

```

```

P_M=P_M+DP_M
I_M=ANINT(I_M+1)
S_M=-S_M
DP_M=S_M/(2*I_M-1)
%WEND
P_M=4*P_M

```

**補足** 項の番号  $I_M$  をプラス 1 するときに組み込み関数 ANINT を使っています。 $I_M$  は本来整数でしかあり得ないのですが、本ソフトウェアは実数で代用するため丸め誤差の累積を防ぐためにこうします。（実際には組み込み関数を使わなくても問題とはならない程度の誤差ですが、あえて正しい記述方法を示しました。）

### ● \$IGAMMA (不完全ガンマ関数)

不完全ガンマ関数は比較的良好に使われる特殊関数で次の式で定義されます。

$$P(a, x) \equiv \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt$$

この式を級数展開および連分数を使って計算する方法が「NUMERICAL RECIPES in C」(技術評論社)に紹介されています。この方法をマクロで表現すると次のようになります。このマクロを引用する前に A と X に値を代入しておいてください。関数値は表示パネルに表示されますが、IGAMMA という変数にも代入されています。

```

# INCOMPLETE GAMMA FUNCTION
# (NUMERICAL RECIPES IN C)
# INPUT : A, X (A>0, X>=0)
# OUTPUT: IGAMMA
EPS_M=3E-7
# G_M=GAMMA(A)
Y_M=A
X_M=A
TMP_M=X_M+5.5
TMP_M=TMP_M-(X_M+0.5)*LOG(TMP_M)
SER_M=1.000000000190015
SER_M=SER_M+76.18009172947146/(Y_M+1)
SER_M=SER_M-86.50532032941677/(Y_M+2)
SER_M=SER_M+24.01409824083091/(Y_M+3)
SER_M=SER_M-1.231739572450155/(Y_M+4)
SER_M=SER_M+0.1208650973866179E-2/(Y_M+5)
SER_M=SER_M-0.5395239384953E-5/(Y_M+6)
G_M=-TMP_M+LOG(2.5066282746310005*SER_M/X_M)
%IF:X<A+1
%IF:X<=0

```

```

    IGAMMA=0
%ELSE
    AP_M=A
    DEL_M=1/A
    SUM_M=DEL_M
    ITER_M=0
    %WHILE:ABS(DEL_M)>ABS(SUM_M)*EPS_M
        AP_M=AP_M+1
        DEL_M=DEL_M*X/AP_M
        SUM_M=SUM_M+DEL_M
        ITER_M=ANINT(ITER_M+1)
    %WEND
    IGAMMA=SUM_M*EXP(-X+A*LOG(X)-G_M)
%END IF
%ELSE
    FAC_M=1
    GOLD_M=0
    A0_M=1
    B0_M=0
    B1_M=1
    A1_M=X
    ITER_M=0
    REPEAT_M=1
    %WHILE:REPEAT_M
        ITER_M=ANINT(ITER_M+1)
        ANA_M=ITER_M-A
        A0_M=(A1_M+A0_M*ANA_M)*FAC_M
        B0_M=(B1_M+B0_M*ANA_M)*FAC_M
        ANF_M=ITER_M*FAC_M
        A1_M=X*A0_M+ANF_M*A1_M
        B1_M=X*B0_M+ANF_M*B1_M
        %IF:A1_M<>0
            FAC_M=1/A1_M
            GNEW_M=B1_M*FAC_M
            REPEAT_M=(ABS((GNEW_M-GOLD_M)/GNEW_
M)>EPS_M)
            GOLD_M=GNEW_M
        %END IF
    %WEND
    IGAMMA=1-EXP(-X+A*LOG(X)-G_M)*GNEW_M
%END IF

```

例	キー操作	表示パネル
誤差関数 erf(0.3) を計算 します。 $\operatorname{erf}(x) \equiv P\left(\frac{1}{2}, x^2\right)$	<div> <div>A 0 . 5</div> <div>ENTER</div> <div>X 0 . 3 A 2</div> <div>ENTER</div> <div>S T G A M M A</div> <div>ENTER</div> </div>	A=0.5 0.5 X=0.3^2 0.09 \$IGAMMA 0.328626759380301

◎ \$GAMMALN (対数ガンマ関数)

不完全ガンマ関数の例の中で対数ガンマ関数を計算している部分があります。その部分だけを配列を使って書き直した例を示します。このマクロを実行する前には配列 GAM\_COF を最大要素数 6 で定義しておく必要があります。変数 X に値を設定しておくくと変数 GAMMALN に対数ガンマ関数値が計算されます。

```
# GAMMA FUNCTION
# (NUMERICAL RECIPES IN C)
# INPUT : X (X>0)
# OUTPUT: GAMMALN
# NOTE : GAMMALN=LOG(GAMMA(X))
# PLEASE DECLARE GAM_COF(6) BEFORE RUN
GAM_COF(1)=76.18009172947146
GAM_COF(2)=-86.50532032941677
GAM_COF(3)=24.01409824083091
GAM_COF(4)=-1.231739572450155
GAM_COF(5)=0.1208650973866179E-2
GAM_COF(6)=-0.5395239384953E-5
GAM_Y=X
GAM_TMP=X+5.5
GAM_TMP=GAM_TMP-(X+0.5)*LOG(GAM_TMP)
GAM_SER=1.000000000190015
GAM_I=0
%WHILE:(GAM_I^6)^0
    GAM_Y=GAM_Y+1
    GAM_SER=GAM_SER+GAM_COF(GAM_I+1)/GAM_Y
    GAM_I=ANINT(GAM_I+1)
%WEND
GAMMALN=-GAM_TMP+LOG(2.5066282746310005*GAM_
SER/X)
```

## 11. 技術仕様

対応 OS	Windows XP/Vista/7
数式処理方式	逆ポーランド記法
計算精度	8 バイト IEEE 浮動小数点数
演算用バッファ	100 段
条件分岐入れ予用バッファ	100 段
繰り返し入れ予用バッファ	100 段
繰り返し複文長さ	100000000 項まで
マクロの数	PC メモリーにのみ制約される
マクロの長さ	PC メモリーにのみ制約される
変数 / 配列の数	PC メモリーにのみ制約される