

---

# ECKERT

バージョン 2019 ユーザーズマニュアル

きくち ゆいしん  
菊地唯真

April 13, 2020

---

# Contents

Chapter 0	<b>はじめに</b>	1
0.1	ECKERT とは . . . . .	1
0.2	対応機能 . . . . .	2
0.3	動作環境 . . . . .	2
0.4	免責事項 . . . . .	2
Chapter 1	<b>ECKERT を使う準備</b>	3
1.1	インストールとアンインストール . . . . .	3
1.2	本マニュアルの読み方 . . . . .	3
1.3	本マニュアルの表記・表示 . . . . .	4
Chapter 2	<b>画面の見方と操作方法</b>	5
2.1	起動と終了 . . . . .	5
2.2	計算モードの画面の全体 . . . . .	5
2.3	計算設定・状態表示領域 . . . . .	7
2.4	スタック表示領域の見かた . . . . .	9
2.5	メッセージ表示領域 . . . . .	11
2.6	コンフィグモードの画面の見かた . . . . .	12
2.7	基本的な操作方法 . . . . .	13
2.8	数の入力方法 . . . . .	14
2.9	接頭辞付きの数の入力 . . . . .	15
2.10	数の入力例 . . . . .	16
2.11	エラーメッセージが表示されたとき . . . . .	17
Chapter 3	<b>設定・表示切り替え</b>	19
3.1	コンフィグモードでの設定 . . . . .	19
3.2	計算モードでの設定 . . . . .	21
3.3	計算モードの設定のまとめ . . . . .	25
3.4	スタックのページめくり . . . . .	26
3.5	レジスタのページめくり . . . . .	27
3.6	値の全体表示 . . . . .	28
3.7	バージョン表示 . . . . .	29
Chapter 4	<b>基本操作～四則演算</b>	31
4.1	基本的なスタック操作 . . . . .	31

4.2	四則演算 . . . . .	33
4.3	複合的な四則演算 . . . . .	34
4.4	四則演算以外の基本的な演算 . . . . .	36
4.5	逆ポーランド表記法に慣れるコツ . . . . .	36
Chapter 5	<b>数学関数</b>	37
5.1	数学関数の使い方 . . . . .	37
5.2	指数・対数 . . . . .	37
5.3	三角関数・逆三角関数 . . . . .	38
5.4	双曲線関数・逆双曲線関数 . . . . .	39
5.5	統計関連の関数 . . . . .	39
5.6	整数丸め関数 . . . . .	40
5.7	整数の関数 . . . . .	40
Chapter 6	<b>便利な計算機能</b>	41
6.1	パーセント計算 . . . . .	41
6.2	時間換算 . . . . .	42
6.3	DMS 変換・逆変換 . . . . .	43
6.4	総和・平均など . . . . .	43
6.5	接頭辞との乗算 . . . . .	44
6.6	接頭辞による除算 . . . . .	45
6.7	角度変換 . . . . .	46
6.8	角度計算 . . . . .	46
6.9	有理数の分解 . . . . .	46
6.10	乱数 . . . . .	47
6.11	キャスト . . . . .	47
6.12	エンジニア向け計算 . . . . .	47
6.13	地震に関する計算 . . . . .	48
6.14	健康や生活に関する計算 . . . . .	48
Chapter 7	<b>複素数の計算</b>	49
7.1	複素数の表示 . . . . .	49
7.2	複素数の作り方 . . . . .	50
7.3	複素数の演算 . . . . .	51
7.4	複素数の分解 . . . . .	52
7.5	複素関数 . . . . .	52
Chapter 8	<b>論理計算</b>	53
8.1	符号なし整数・ブーリアンの表示 . . . . .	53
8.2	プッシュするビット長の設定 . . . . .	53
8.3	符号なし整数の表示切り替え . . . . .	54
8.4	符号なし整数・ブーリアンの入力 . . . . .	54
8.5	基本的な論理演算 . . . . .	55
8.6	ビットシフト . . . . .	55

8.7	ローテイト . . . . .	56
8.8	その他符号なし整数に使える機能 . . . . .	56
8.9	論理演算の積算 . . . . .	56
Chapter 9	<b>ベクトルの計算</b>	57
9.1	ベクトルの表示 . . . . .	57
9.2	ベクトルの作りかた . . . . .	57
9.3	ベクトルからの抽出 . . . . .	59
9.4	ベクトルを含む四則演算 . . . . .	60
9.5	内積・外積 . . . . .	60
9.6	ベクトルノルム . . . . .	61
9.7	ベクトルの転置 . . . . .	61
Chapter 10	<b>行列の計算</b>	63
10.1	行列の表示 . . . . .	63
10.2	行列の作りかた . . . . .	63
10.3	行列からの抽出 . . . . .	65
10.4	行列を含む四則演算 . . . . .	67
10.5	行列式と逆行列 . . . . .	67
10.6	行列の転置 . . . . .	68
10.7	その他の行列演算 . . . . .	68
Chapter 11	<b>レジスタ機能</b>	69
11.1	レジスタとは . . . . .	69
11.2	レジスタの表示切り替え . . . . .	70
11.3	指定レジスタへのストア . . . . .	70
11.4	指定レジスタのロード . . . . .	72
11.5	指定レジスタのデリート . . . . .	73
11.6	レジスタ演算 . . . . .	74
11.7	レジスタクリア . . . . .	75
11.8	文字列とレジスタ . . . . .	75
Chapter 12	<b>スタック操作</b>	77
12.1	スタックの特殊操作 . . . . .	77
12.2	基本的なスタック操作 . . . . .	77
12.3	スタックの順番を変更する操作 . . . . .	77
12.4	コピー・上書きをする操作 . . . . .	80
12.5	削除をする操作 . . . . .	83
12.6	その他のスタック操作 . . . . .	85
Chapter 13	<b>単位換算機能</b>	87
13.1	対応する単位の種類 . . . . .	87
13.2	単位換算機能の使い方 . . . . .	87
13.3	長さの単位 . . . . .	88

13.4	長さの逆数の単位 . . . . .	88
13.5	面積の単位 . . . . .	89
13.6	面積の逆数の単位 . . . . .	89
13.7	体積の単位 . . . . .	90
13.8	体積の逆数の単位 . . . . .	91
13.9	時間の単位 . . . . .	91
13.10	時間の逆数の単位 . . . . .	91
13.11	質量の単位 . . . . .	92
13.12	速度の単位 . . . . .	92
13.13	加速度の単位 . . . . .	93
13.14	力の単位 . . . . .	93
13.15	圧力の単位 . . . . .	93
13.16	エネルギーの単位 . . . . .	94
13.17	温度の単位 . . . . .	94
Chapter 14	<b>ソフトウェア内蔵定数</b>	95
14.1	定数の入力 . . . . .	95
14.2	数学定数 . . . . .	95
14.3	基本的な物理定数 . . . . .	95
14.4	電磁気学 . . . . .	96
14.5	原子・核物理学 . . . . .	96
14.6	物理化学 . . . . .	97
14.7	協定値 . . . . .	98
14.8	プランク単位 . . . . .	98
14.9	天文学 . . . . .	98
14.10	紙のサイズ . . . . .	99
Chapter 15	<b>その他の機能</b>	101
15.1	オールクリア . . . . .	101
15.2	オールリセット . . . . .	101
15.3	元に戻す・やり直し . . . . .	101
15.4	JSON 出力 . . . . .	102
15.5	マクロ機能 . . . . .	102
15.6	精度の検証 . . . . .	103
15.7	特殊な起動 . . . . .	103
Chapter 16	<b>メッセージ一覧</b>	105
16.1	エラーメッセージ . . . . .	105
16.2	確認メッセージ . . . . .	106
16.3	入力待ち・入力確認メッセージ . . . . .	106
Chapter 17	<b>技術情報</b>	107
17.1	本ソフトウェアの型 . . . . .	107
17.2	計算精度 . . . . .	107

17.3	数学的定義 . . . . .	108
Chapter 18	<b>トラブルシューティング</b>	<b>111</b>
18.1	まったく操作がわからない . . . . .	111
18.2	値の全体を表示したい . . . . .	111
18.3	分数や小数の表示を変えたい . . . . .	111
18.4	複素数の表示を変えたい . . . . .	111
18.5	スタックやレジスタの全要素を一覧したい . . . . .	112
18.6	異常な計算結果が出た . . . . .	112
18.7	エラーになって計算できない . . . . .	113
18.8	バグと思われる挙動を見つけた . . . . .	113

# Chapter 0

## はじめに

### 0.1 ECKERT とは

ECKERT はキーボード操作の関数電卓ソフトウェアです。

正式名称は Engineering Calculator with KEyboard and Refined Tools です。

Engineering Calculator with KEyboard and Refined Tools (C) 2014-2019 Yuishin Kikuchi		
-----		
HOMURA: (FD) (Rad) (Hex) (Dword)		
Std: 9/15, Stack: 6, History: 0/10		
=====		
#	TYPE :	VALUE
6:	Integer :	12
5:	Floating :	1.5
4:	Complex :	3/25 - i4/25
Z:	Matrix :	[[2, 3], [3, 4]]
Y:	Rational :	2.1/4
X:	Tuple(Col) :	(1 + i2, 2 + i3, 3 + i4)
-----		
MAKE COLUMN TUPLE		
Ready to operate		
-----		
>		

画面を確認しつつ、数値やキーワードを入力しながら計算します。逆ポーランド記法を採用しているため、複雑な計算もカッコを用いずに行うことができます。なお、画面表示はすべて英語です。

## 0.2 対応機能

ECKERT は次の機能・計算に対応します。

- 数値の接頭辞付加<sup>\*1</sup>
- パーセント計算
- 論理計算
- 有理数計算
- 税込み・税抜き計算
- ベクトルの計算
- 複素数・複素関数
- 接頭辞との乗除算
- 行列の計算
- 指数・対数
- $2\pi$  との乗除算
- レジスタ機能<sup>\*2</sup>
- 三角関数
- デシベル変換
- 単位換算機能<sup>\*3</sup>
- 双曲線関数
- 基数（進数）変換
- 数学・科学定数<sup>\*3</sup>

## 0.3 動作環境

公開しているバイナリファイル (\*.exe) の動作確認は、次の Windows で行っております。

- Windows 8
- Windows 8.1
- Windows 10

これ以前の Windows では正常に動作しない可能性があります。

## 0.4 免責事項

本ソフトウェアおよび本マニュアルの著作権は、菊地唯真 (Yuishin Kikuchi) に属します。

ECKERT は無償かつ無保証です。本ソフトウェアによって生じるいかなるトラブルについても保証しかねます。

また、画面表示の乱れや、疑わしい計算結果・挙動にお気づきの場合、再現手順を添えて、ご連絡ください。数学的に不自然な仕様についても、報告していただくと助かります。

作者宛て E メールアドレス：mailto:only.my.truth@gmail.com

---

<sup>\*1</sup> 「12k」（12 キロ）や「32u」（32 マイクロ）などの入力です。

<sup>\*2</sup> 計算結果を別の場所に保存し、呼び出せる機能（いわゆるメモリー機能）です。

<sup>\*3</sup> 2014 CODATA を基にしています。



# Chapter 1

## ECKERT を使う準備

### 1.1 インストールとアンインストール

配布ファイル内の eckert86.exe と eckert64.exe が実行ファイルです。eckert86.exe は 32 ビットで、eckert64.exe は 64 ビットで動作します。ビット数については、お使いのオペレーティングシステムをご確認ください。

なお、どちらの実行ファイルも独立しており、単体で動作するため、片方が不要な場合は削除していただいても問題ありません。本ソフトウェアはレジストリなどのシステムの変更を行わないため、USB フラッシュディスクなどで持ち運びができます。

インストールは、実行ファイルのコピーのみです。実行ファイルのいずれかを、好きなディレクトリ（フォルダ）にコピーしてお使いいただけます。なお、本ソフトウェアは、設定の読み書きや計算結果のファイル出力があるため、専用のディレクトリを用意することを強くおすすめします。

アンインストールは、実行ファイルの削除のみです。場合によっては、同ディレクトリに config ファイルが作られていますが、その場合は同時に削除します。なお、専用のディレクトリを用意していた場合は、ディレクトリごと削除していただいても問題ありません。

### 1.2 本マニュアルの読み方

本マニュアルでは、関数電卓アプリケーション ECKERT（以下、本ソフトウェア）の機能をすべて説明します。ただし、本マニュアルでは基本的な数学的定義を示しません。あくまで操作マニュアルとしてお使いください。

逆ポーランド記法の電卓を初めてご使用の場合、Chapter 2「画面の見かたと操作方法」、Chapter 4「基本操作～四則演算」をお読みください。基本操作を把握したら、Chapter 5「数学関数」、Chapter 6「便利な計算機能」、Chapter 7「複素数の計算」、Chapter 11「レジスタ機能」などをお読みください。

逆ポーランド記法を既にご存知の場合、Chapter 4「基本操作～四則演算」を斜め読みする程度で基本操作がすぐに理解できます。各チャプターのキーワードの表にざっと目を通すと、時間をかけずに機能を把握できます。

表示桁数や画面表示の設定を行うには、Chapter 3「設定・表示切り替え」をお読みください。ただし、四則演算などの基本的な操作を理解してから読むことをおすすめします。

## 1.3 本マニュアルの表記・表示

本マニュアルでは、次のような表示を用います。

重要なことを示します。

注意すべきことを示します。

### 入力するコマンド

コマンドは `Cmd INPUT` の形で示します。

この表示の右側の文字列を、本ソフトウェアの入力欄に入力する操作を示します。

本ソフトウェアはスタックという計算スペース（Chapter 3「画面の見かたと操作方法」で詳しく説明）を用います。スタックの状態を表すのに、以下の形の表を用います。

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	12
X:	Floating	:	1.5

「TYPE」の列はデータ型を、「VALUE」の列は値を示しています。キーワード一覧を示す際には、次のような表を示します。

機能	キーワード	R	D	演算内容
加算	ADD, +	2	2	$Y + X$
減算	SUB, -	2	2	$Y - X$

「機能」の列は、実行する機能を表しています。「キーワード」の列は、機能に対応する文字列です。ひとつの機能に対して複数のキーワードが割り当てられている場合、どれを使ってもよいことを示しています。

機能とキーワードの表の詳しい読みかたは、Chapter 4「基本操作～四則演算」で説明します。

## Chapter 2

# 画面の見方と操作方法

## 2.1 起動と終了

起動するには、実行ファイルをダブルクリックします。コンソールからコマンドライン引数なしで起動しても同じです。

終了するには、キーボードで以下のいずれかを入力して、Enter キーを押下します。なお、基本的に大文字・小文字は区別されませんが、一部例外があります。

機能	キーワード
終了	EXIT, QUIT, Q

## 2.2 計算モードの画面の全体

計算モードの画面表示の見方を説明します。まずは画面全体の見かたです。

Engineering Calculator with KEyboard and Refined Tools (C) 2014-2019 Yuishin Kikuchi		
-----		
HOMURA: (FD) (Rad) (Hex) (Dword)		
Std: 9/15, Stack: 6, History: 0/10		
=====		
#	TYPE :	VALUE
6:	Integer :	12
5:	Floating :	1.5
4:	Complex :	3/25 - i4/25
Z:	Matrix :	[[2, 3], [3, 4]]
Y:	Rational :	2.1/4
X:	Tuple(Col) :	(1 + i2, 2 + i3, 3 + i4)
-----		
MAKE COLUMN TUPLE		
Ready to operate		
-----		
>		

最初の 2 行はソフト名と作者の表示です。モードに関係なく表示されます。

区切り線をはさみ、計算設定・状態表示領域があります。

```
HOMURA: (FD) (Rad) (Hex) (Dword) [i.a/b]
Std: 9/15, Stack: 6, History: 0/10
```

二重区切り線をはさみ、もう一つの区切り線の下にあるのがスタック表示領域です。入力したデータや計算結果の確認はここです。

#	TYPE	:	VALUE
6	Integer	:	12
5	Floating	:	1.5
4	Complex	:	3/25 - i4/25
Z	Matrix	:	[[2, 3], [3, 4]]
Y	Rational	:	2.1/4
X	Tuple(Col)	:	(1 + i2, 2 + i3, 3 + i4)

スタック表示領域の左側にはデータの型（種類）、右側に値が表示されます。  
スタック表示の下はメッセージ表示領域です。ここには、直前に実行した機能やメッセージが表示されます。

```
-----
MAKE COLUMN TUPLE
Ready to operate
-----
```

画面の一番下が入力欄です。ここにコマンドを入力します。

```
>
```

次節から、各表示について細かく説明します。

## 2.3 計算設定・状態表示領域

計算設定・状態表示領域の見方を説明します。

HOMURA: (FD) (Rad) (Hex) (Dword) [i.a/b]  
 Std: 9/15, Stack: 6, History: 0/10

上の行では、次のシンボルが表示されます。

### 2.3.1 有理数・整数の表示モード

シンボル	意味
(AD)	自動小数表示
(FD)	強制小数表示
(FF)	強制分数表示

### 2.3.2 角度のモード

シンボル	意味
(Deg)	度数法モード
(Rad)	ラジアンモード
(Gra)	グラードモード

### 2.3.3 バイナリの表示モード

シンボル	意味
(Bin)	2進数表示
(Oct)	8進数表示
(Sdec)	符号あり 10進数表示
(Udec)	符号なし 10進数表示
(Hex)	16進数表示

### 2.3.4 論理計算のモード

シンボル	意味
(Byte)	8ビットモード
(Word)	16ビットモード
(Dword)	32ビットモード
(Qword)	64ビットモード

## 2.3.5 一般モード

シンボル	意味
[Reg]	レジスタ表示
[Eul]	オイラー表示
[Eul(Pi)]	オイラー表示 ( $\pi$ ラジアン)
[i.a/b]	帯分数表示

(丸カッコ) のシンボルは、各区分について必ずいずれかが表示されます。[角カッコ] のシンボルは、表示・非表示となります。

HOMURA: (FD) (Rad) (Hex) (Dword) [i.a/b]

上のような表示ならば、強制分数表示、ラジアンモード、16 進数表示、32 ビットモード、帯分数表示の設定です。

下の行では、3 つの要素が表示されます。最初に、実数の表示モードと桁数です。

シンボル	意味
Std	実数の標準表示
Fix	実数の固定小数点表示
Sci	実数の指数表示
Eng	実数のエンジニアリング表示

この直後の「整数/整数」の表示は、現在の表示桁数（変更可能）と、設定可能な最大の表示桁数（変更不可能）を示しています。なお、桁数の変更方法は Chapter 3「設定・表示切り替え」で説明します。

次に、スタックの要素数です。0 の場合は Empty、それ以外は数が表示されます。

3 番目は履歴の表示です。

表示	意味
OFF	履歴機能オフ
Init	初期状態
整数/整数	※後述

「整数/整数」の表示は、元に戻す機能で履歴をさかのぼった回数と、管理している（過去の）履歴数を示しています。

Std: 6/15, Stack: 6, History: 0/10

上のような表示ならば、標準表示で、6 桁を表示しており、最大 15 桁まで表示できることを示しています。スタックに 11 個のデータが含まれていることを示しています。4 回履歴をさかのぼった状態で、最大で 10 回までさかのぼることができることを示しています。

## 2.4 スタック表示領域の見かた

本ソフトウェアの鍵となる、スタックについて説明します。

#	TYPE	:	VALUE
6:	Integer	:	12
5:	Floating	:	1.5
4:	Complex	:	3/25 - i4/25
Z:	Matrix	:	[[2, 3], [3, 4]]
Y:	Rational	:	2.1/4
X:	Tuple(Col)	:	(1 + i2, 2 + i3, 3 + i4)

スタックとは、数を格納する領域です。本ソフトウェアのスタック表示領域は、画面の下側（後述する「末尾」）から押し上げたり取り除いたりする形で動きます。

各行には、スタックの中の位置を示すアルファベットまたは番号、データの型、値が表示されます。データの型とは、いわゆる数の種類のことです。たとえば、整数ならば Integer が、有理数ならば Rational が表示されます。

なお、本マニュアルでは、スタックを次のようにも表示します。

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	12
X:	Floating	:	1.5

本ソフトウェアのスタックの容量は、実行環境の許す限り無制限です。

X の行をスタックの末尾と呼びます。Y は末尾 2 番目、Z は末尾 3 番目です。それ以降は、4, 5, …と数字で示します。また、X の行にあるデータを単に X と呼び、Y や Z についても同様に表現します。

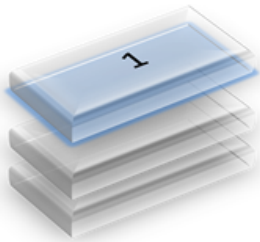
次に、スタックの動きを視覚的に示します。

スタックは、カードの山にカードを重ねたり、取り除いたりするのと同じ扱いの構造です。



上図のように、カードが重なっている様子を想像します。カードを 1 枚ずつ重ねたとすると、1 のカードを重ねた後に 2 のカードを重ねています。これを、次のように表現します。

#	TYPE	:	VALUE
Z:		:	
Y:	Integer	:	1
X:	Integer	:	2



上図では、上図からカードが 1 枚取り除かれています。スタックで言えば、末尾の X が削除されています。

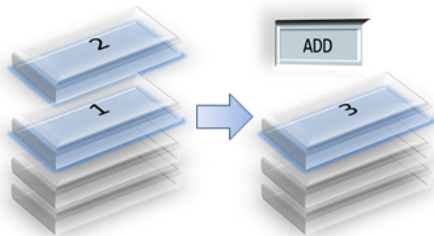
#	TYPE	:	VALUE
Z:		:	
Y:		:	
X:	Integer	:	1

では、加算を説明します。2 枚のカードがある状態で、2 枚のカードを引き、その加算結果のカードを上重ねます。これが、スタックで計算する基本的な流れです。

#	TYPE	:	VALUE
Z:		:	
Y:	Integer	:	1
X:	Integer	:	2
> +			

加算実行

#	TYPE	:	VALUE
Z:		:	
Y:		:	
X:	Integer	:	3



スタックへの追加、削除、演算という操作があることを最初に理解しましょう。  
なお、左側に表示される型には、次のようなものがあります。



シンボル	意味
Error	エラーを示す文字列
String	文字列
Integer	整数
Floating	浮動小数点数
Rational	有理数
Infinity	無限大
Complex	複素数
Boolean	ブーリアン
Byte	8 ビット 整数
Word	16 ビット 整数
Dword	32 ビット 整数
Qword	64 ビット 整数
Tuple[Row]	横方向ベクトル
Tuple(Col)	縦方向ベクトル
Matrix	行列

## 2.5 メッセージ表示領域

メッセージ表示領域では、直前に実行した機能やメッセージを表示します。

```
-----
MAKE COLUMN TUPLE
Ready to operate
-----
```

上の行では実行した機能を、下の行ではその他のメッセージを表示させます。ゼロ除算など、実行できない操作が入った場合、その場で処理を中止し、メッセージ表示領域の下の方にエラーメッセージを表示します。

```
[!] ADD Y+X
Error: Too few arguments
```

なお、実行結果に関するメッセージが表示される場合、次のシンボルが上の行、つまり実行した機能の左側に表示されます。

シンボル	意味
[!]	エラーによる中止
[i]	特別な内部処理を行った
[C]	入力待ち / 設定の確認

具体的なメッセージの意味については、Chapter 17「メッセージ一覧」をお読みください。

※ [?] が表示された場合、バグとしてご報告をお願いします。

## 2.6 コンフィグモードの画面の見かた

計算モードで「Cmd CONFIG」と入力すると、コンフィグモードに移ります。

```
Engineering Calculator with KEyboard and Refined Tools
(C) 2014-2019 Yuishin Kikuchi
-----
CONFIGURATION MODE
=====
Interface
  History size  (hist): 10
  Display width (width): 79
  Display lines (lines): 11
Management
  Load config  (load)
  Save config  (save)
  Reset config (reset)
-----
ECKERT Config
Type "calc" or "homura" to quit config
-----
>
```

```
Interface
  History size  (hist): 10
  Display width (width): 79
  Display lines (lines): 11
```

最大履歴数、画面幅、スタック表示領域の行数の設定が表示されています。設定方法は、Chapter 3「設定・表示切り替え」で示します。

```
Management
  Load config  (load)
  Save config  (save)
  Reset config (reset)
```

コンフィグ管理用のコマンドです。

## 2.7 基本的な操作方法

本ソフトウェアは、キーボードで数値やキーワードを入力することによって操作します。なお、対応する入力には半角英数字のみで、全角数字や日本語の入力に対する動作は保証しません。

数値やキーワードを 1 つ、または半角スペース区切りで複数入力し、Enter キーで確定することにより、入力した順番の通りに機能・命令を実行します。ただし、設定変更のキーワードについては 1 つで入力する必要がある場合もあります。

このように、文字を入力して Enter キーという流れの操作となります。Enter キーを押さない限りは次の画面に移りませんので、ご注意ください。なお、数値入力の際の接頭辞を除き、大文字・小文字は区別されません。

また、文字キー以外の入力に対する動作を定義していません。

たとえば、加算、乗算の順番に機能を実行するには、次のように入力します。


 + \*

ひとつの機能に対して複数のキーワードが割り当てられている場合もあります。その場合、どちらを使用しても同じ結果となります。また、同一のキーワードでも、計算モードに応じて異なる機能が実行される場合がありますので、ご注意ください。

数を入力するには、単純に数を記述します。なお、数値の入力についても、スペース区切りで並べることができます。

 1 2

数の入力と計算キーワードを混ぜて記述することもできます。

 2 5 /

数の入力について、詳しくは次節で説明します。

## 2.8 数の入力方法

本ソフトウェアで対応する数の入力について示します。

### 2.8.1 整数・小数

整数はそのまま入力します。

小数は小数点を使って入力します。「Cmd.2」のような整数部分の省略や「Cmd1.」のような小数部分の省略ができます。

### 2.8.2 指数表記

指数表記とは、E または e を使って、10 進数の指数を表現する表記です。「**仮数部 E 指数部**」の形で表記します。この仮数部には整数または小数が、指数部には符号を認める整数が入ります。数学的な表現では、「 $m \times 10^x$ 」のように書きますが、この「 $\times 10$ 」を「E」に置き換えて表現します。

たとえば、 $6.02 \times 10^{-23}$  は「6.02E-23」という表記、 $1.01325 \times 10^5$  は「1.01325E5」という表記になります。

### 2.8.3 虚数

プラスの虚数単位は「Cmdi」「Cmd+i」と、マイナスの虚数単位は「Cmd-i」と入力します。

整数、小数、指数表記に対して、「i」を前置して、虚数を入力できます。

※虚数単位  $i$  の大文字・小文字は区別されません。

### 2.8.4 無限大

プラスの無限大は、「Cmd INF」「Cmd+INF」「Cmd+INFINITY」と入力します。マイナスの無限大は「Cmd-INF」「Cmd-INFINITY」と入力します。

### 2.8.5 ブーリアン

ブーリアンは、真の場合「Cmd TRUE」「CmdT」、偽の場合「Cmd FALSE」「CmdF」と入力します。

### 2.8.6 符号なし 10 進数

符号なし整数は、「u」を前置し、直後に符号をつけずに整数を続けて入力します。

### 2.8.7 N 進数

2 進数は、「0b」を前置し、直後に 0 と 1 による 2 進数表現を入力します。

8 進数は、「0o」を前置し、直後に 0～7 による 8 進数表現を入力します。

16 進数は、「0x」を前置し、直後に 0～9, A～F による 16 進数表現を入力します。

## 2.9 接頭辞つきの数の入力

整数・小数・指数表記・虚数（前置表記）に対して、接頭辞をつけた入力が可能です。対応する SI 接頭辞、2 進接頭辞の一覧を示します。大文字・小文字を区別します。

記号	名前	値	記号	名前	値
da	DECA	$\times 10^{+01}$	d	DECI	$\times 10^{-01}$
h	HECTO	$\times 10^{+02}$	c	CENTI	$\times 10^{-02}$
K, k	KILO	$\times 10^{+03}$	m	MILLI	$\times 10^{-03}$
M	MEGA	$\times 10^{+06}$	u	MICRO	$\times 10^{-06}$
G	GIGA	$\times 10^{+09}$	n	NANO	$\times 10^{-09}$
T	TERA	$\times 10^{+12}$	p	PICO	$\times 10^{-12}$
P	PETA	$\times 10^{+15}$	f	FEMTO	$\times 10^{-15}$
E	EXA	$\times 10^{+18}$	a	ATTO	$\times 10^{-18}$
Z	ZETTA	$\times 10^{+21}$	z	ZEPTO	$\times 10^{-21}$
Y	YOTTA	$\times 10^{+24}$	y	YOCTO	$\times 10^{-24}$
Ki, ki	KIBI	$\times 1024^1$			
Mi, mi	MEBI	$\times 1024^2$			
Gi, gi	GIBI	$\times 1024^3$			
Ti, ti	TEBI	$\times 1024^4$			
Pi, pi	PEBI	$\times 1024^5$			
Ei, ei	EXBI	$\times 1024^6$			
Zi, zi	ZEBI	$\times 1024^7$			
Yi, yi	YOBI	$\times 1024^8$			

カンマ区切りで複数表示されているものは、どちらを使っても同じという意味です。

## 2.10 数の入力例

整数

`-3`

接頭辞

`3k`

指数表記

`2.998e8`

虚数単位

`-i`

虚数

`i12`

虚数（符号）

`-i5`

無限大

`-inf`

ブーリアン

`t`

符号なし整数

`u65536`

2進数

`0b1010`

8進数

`0o100`

16進数

`0xFFFE`

また、この他、数学・科学定数のキーワードにも対応しています。詳しくは Chapter 14 「ソフトウェア内蔵定数」をご覧ください。

定数名	キーワード	値
円周率	PI	3.141 592 653 589 79
ネイピア数	E	2.718 281 828 459 05
オイラーの定数	EG	0.577 215 664 901 533

ちなみに、数ではありませんが、文字列にも対応しています。文字列を入力するには、半角ダブルクォーテーションを用います。

文字列 `"This is test"`


レジスタに目印やメッセージを残したり、マクロ機能を使ったりするために用います。

## 2.11 エラーメッセージが表示されたとき

ある機能を実行する際にエラーが起きたとき、そこで処理を中断します。つまり、その機能が実行される前の状態で止まります。そして、メッセージ表示領域にエラーメッセージが表示されます。

エラーメッセージが表示されても、入力の方法は変わりません。また同様にコマンドを入力し、正常に実行されれば、エラーメッセージは消えます。

スペース区切りで複数の入力をしている場合でも、処理はスペース区切りごとなので、あくまでエラーになるまで処理を実行します。

 5 0 /

(この意味は Chapter 4「基本操作～四則演算」を読むとおわかりいただけます)

たとえば、上の入力の場合、ゼロ除算でエラーとなります。ただ、その前に 5 と 0 はスタックに追加されているので、その状態で処理が中止されます。

エラーが表示された場合、元に戻す機能が便利です。

機能	キーワード
元に戻す	UNDO, U
やり直し	REDO, R

詳しくは、Chapter 15「その他の機能」をお読みください。

エラーメッセージの意味が知りたい場合、Chapter 16「メッセージ一覧」を参照ください。

また、非対応の入力を検出すると、次のように表示します。

[!] OPERATIONAL ERROR  
Error: Unsupported operation or notation

このメッセージが表示された場合、入力するコマンドのスペルの間違いがないかをご確認ください。

また、対応しているキーワードであっても、使うモードが異なったり、設定変更などの単体で使われるキーワードが複数で使われていたりすると、この表示になることがあります。





## Chapter 3

# 設定・表示切り替え

基本操作を理解してから本章をお読みください。

### 3.1 コンフィグモードでの設定

最大履歴数、画面幅、スタック表示領域の行数はコンフィグモードで設定できます。

コンフィグモードに入るには、キーワード「`Cmd CONFIG`」を入力します。コンフィグモードから計算モードに移る場合は、キーワード「`Cmd HOMURA`」または「`Cmd CALC`」と入力します。

#### 3.1.1 最大履歴数の設定

キーワード「`Cmd HIST`」を入力し、続けて整数を指定します。この際、「`Cmd hist 10`」のように、スペース区切りで入力することもできます。ここで0を指定すると、履歴機能をオフにできます。

なお、最大履歴数のデフォルト値は10です。

たとえば、最大履歴数を20にしたい場合、次のように入力します。

```
Cmd hist 20
```

#### 3.1.2 画面幅の設定

キーワード「`Cmd WIDTH`」を入力し、続けて整数を指定します。この際、「`Cmd width 79`」のように、スペース区切りで入力することもできます。ただし、最小値は決められておりますので、その値よりも小さいものが指定された場合、最小値に設定されます。

なお、画面幅のデフォルト値は79で、最小幅は60です。

たとえば、画面幅を69にしたい場合、次のように入力します。

```
Cmd width 69
```

#### 3.1.3 スタック表示領域の行数の設定

キーワード「`Cmd LINES`」を入力し、続けて整数を指定します。この際、「`Cmd lines 11`」のように、スペース区切りで入力することもできます。ただし、最小値は決められておりますので、その値よりも小さいものが指定された場合、最小値に設定されます。

なお、行数のデフォルト値は11で、最小は4です。

たとえば、スタック表示領域の行数を 20 にしたい場合、次のように入力します。

**Cmd.** `lines 20`

### 3.1.4 設定のロード・セーブ・リセット

コンフィグモードにおける設定は、config ファイルに保存できます。コンフィグの管理を行うには、次のキーワードを用います。

機能	キーワード
コンフィグのロード	LOAD
コンフィグのセーブ	SAVE
コンフィグのリセット	RESET, RST

コンフィグのセーブを行うと、次回起動時にその設定が読み込まれ、履歴数や画面の幅が自動的に適用されます。

コンフィグのロードを行うと、明示的に config ファイルを読み込みます。

コンフィグのリセットを行うと、コンフィグの設定すべてがデフォルトに戻ります。ただ、コンフィグのリセットを行っても config ファイルは変更されません。

### 3.1.5 コンフィグモードでの設定まとめ

コンフィグモードでのキーワードをまとめます。

機能	キーワード
コンフィグモード	CONFIG
計算モード	CALC, HOMURA
最大履歴数	HIST
画面の表示幅	WIDTH
スタック表示領域の行数	LINES
コンフィグのロード	LOAD
コンフィグのセーブ	SAVE
コンフィグのリセット	RESET, RST

## 3.2 計算モードでの設定

角度モードや型表示、表示桁数などの設定は計算モードで行えます。

### 3.2.1 有理数の表示モード

実数の表示形式が通常表示のとき、有理数の表示を 3 種類から選ぶことができます。

#### 自動小数表示

有理数が 10 進数の有限小数で表示可能な場合、小数で表示します。それ以外の場合、分数で表示します。

#### 強制小数表示

すべての有理数を小数で表示します。

#### 強制分数表示

すべての有理数を分数で表示します。

それぞれのモードを指定するには、次のキーワードを単体で用います。



モード	キーワード	画面表示
自動小数表示	AD	(AD)
強制小数表示	FD	(FD)
強制分数表示	FF	(FF)

なお、デフォルトは強制小数表示です。

### 3.2.2 角度モード

三角関数や複素偏角の計算について、角度の単位を設定できます。それぞれのモードを指定するには、次のキーワードを単体で用います。

モード	キーワード	画面表示
度数法	DEG	(Deg)
ラジアン	RAD	(Rad)
グラード	GRAD, GRA	(Gra)

このモードは、三角関数「 SIN」や、複素偏角「 ARG」などの計算に関わります。なお、デフォルトはラジアンモードです。

### 3.2.3 N 進数表示モード

8 ビット整数、16 ビット整数、32 ビット整数、64 ビット整数の表示に関して、進数表示を変えることができます。

それぞれのモードを指定するには、次のキーワードを単体で用います。

モード	キーワード	画面表示
2 進数表示	BIN	(Bin)
8 進数表示	OCT	(Oct)
符号つき 10 進数表示	SDEC	(Sdec)
符号なし 10 進数表示	UDEC	(Udec)
16 進数表示	HEX	(Hex)

なお、デフォルトは 16 進数モードです。

### 3.2.4 N ビットモード

論理計算を行う際に、扱いたいビット数を 8, 16, 32, 64 の中から選ぶことができます。符号なし整数の入力があった場合、選ばれたビット数でデータが追加されます。

それぞれのモードを指定するには、次のキーワードを単体で用います。

モード	キーワード	画面表示
8 ビットモード	BYTE	(Byte)
16 ビットモード	WORD	(Word)
32 ビットモード	DWORD	(Dword)
64 ビットモード	QWORD	(Qword)

なお、デフォルトは 32 ビットモードです。

### 3.2.5 型表示

スタック表示領域の左側の型表示は、非表示にもできます。キーワード「Cmd TYPE」を単体で入力すると、表示・非表示を切り替えられます。なお、デフォルトでは表示される設定です。

### 3.2.6 レジスタ表示

レジスタの表示・非表示を切り替えるには、キーワード「Cmd REG」または「Cmd REGISTER」を単体で入力します。レジスタは、スタック表示領域の上半分に表示されます。そのため、レジスタが表示されているとき、スタックの表示領域は狭くなります。

レジスタ表示が有効のとき、計算設定・状態表示領域に [Reg] が表示されます。

なお、デフォルトでは非表示の設定です。

### 3.2.7 オイラー表示

複素数の表示形式について、 $a + ib$ （直交座標）の形式と  $r \exp(i\theta)$ （極座標）の形式を切り替えるには、キーワード「Cmd EULER」または「Cmd EUL」を単体で入力します。オイラー表示が有効のとき、計算設定・状態表示領域に [Eul] が表示されます。また、表示される偏角は角度モードに依存します。

モード	数式	画面表示
直交座標	$5 + 12i$	5 + i12
極座標（度数法）	$13 \angle 67 [\text{deg}]$	13 exp(+i67.d)
極座標（ラジアン）	$13 \angle 1.3 [\text{rad}]$	13 exp(+i1.3) 13 exp(+i0.41 Pi)
極座標（グラード）	$13 \angle 75 [\text{gra}]$	13 exp(+i75.g)

また、ラジアンモードのとき、さらに角度を  $\pi$  ラジアンに直すことができます。 $\pi$  ラジアンモードの有効・無効を切り替えるには、「Cnld PIRAD」または「Cnld PRAD」を単体で入力します。

オイラー表示が有効かつ  $\pi$  ラジアンモードも有効のとき、計算設定・状態表示領域に [Eul(Pi)] が表示されます。

なお、デフォルトは直交座標形式です。

### 3.2.8 帯分数表示

有理数の仮分数・帯分数表示を切り替えるには、キーワード「Cnld FRACTION」または「Cnld FRAC」を単体で入力します。有理数の表示は、次のようになります。帯分数は小数点「.」を用います。

数値	仮分数	帯分数	小数
+3/2	3/2	1.1/2	1.5
-6/5	-6/5	-1.1/5	-1.2

計算設定・状態表示領域には、帯分数表示のとき [i.a/b] が表示されます。なお、デフォルトでは仮分数表示です。

### 3.2.9 実数の表示形式

実数の表示は、通常表示、固定小数点表示、指数表示、エンジニアリング表示の4種類から選ぶことができます。

#### 通常表示

値に応じて、柔軟に表示桁数が変化する表示です。有理数は有理数の表示モードに従います。

#### 固定小数点表示

小数点以下の桁数を固定する表示です。整数や有理数は小数で表示されます。

#### 指数表示

「1.2E+10」のように、仮数部  $m$  を  $0 \leq m < 10$  にして、指数部は E を使って表示します。例の場合、 $1.2 \times 10^{10}$  という意味です。整数や有理数は小数で表示されます。

#### エンジニアリング表示

「12E-03」のように、仮数部  $m$  を  $0 \leq m < 1000$  にして、指数部は E を使って、かつ3の倍数にする表示です。例の場合、 $12 \times 10^{-3}$  ですが、これを「12 ミリ」と読むことができます。整数や有理数は小数で表示されます。

それぞれのモードを指定するには、次のキーワードを単体で使用します。

モード	キーワード	画面表示
通常表示	STD	Std
固定小数点表示	FIX	Fix
指数表示	SCI	Sci
エンジニアリング表示	ENG	Eng

実数の表示形式が通常以外の場合、有理数は小数表示になります。また、デフォルトでは通常表示です。

### 3.2.10 実数の表示桁数

本ソフトウェアでは、実数の表示桁数を設定できます。ここでの桁数の定義は、表示形式ごとに、次のようにしています。

モード	「桁数」の意味
通常表示	有効数字
固定小数点表示	小数点下の桁数
指数表示	有効数字
エンジニアリング表示	有効数字

実数の表示桁数を操作するには、キーワード「**Cmd** DISP」または「**Cmd** DIGIT」を単体で入力します。続けて整数を入力することにより、設定を変更できます。たとえば、表示桁数を3桁にするには、次のように入力します。

**Cmd** digit 3

なお、この桁数は実数の表示形式ごとの設定です。

また、最大桁数と最小桁数は表示形式ごとに決まっており、小さすぎる値が指定された場合は最小桁数に、大きすぎる値が指定された場合には最大桁数に設定されます。

モード	最小桁数	最大桁数
通常表示	1	15
固定小数点表示	0	15
指数表示	1	15
エンジニアリング表示	1	15

表示例: 円周率の10倍 (31.4159265358979) の場合

```
Std:  5/15  31.416
Fix:  5/15  31.41593
Sci:  5/15  3.1416E+01
Eng:  5/15  31.416E+00
```

なお、デフォルトの表示桁数は、すべて9桁です。

また、「**Cmd** disp 10 36」のように、桁数指定の整数に続けてスペース区切りで何かトークン（列）を入力しても、それらは無視されます。

### 3.3 計算モードの設定のまとめ

計算モードでの設定のキーワードをまとめます。

モード	キーワード	画面表示
自動小数表示	AD	(AD)
強制小数表示	FD	(FD)
強制分数表示	FF	(FF)
度数法モード	DEG	(Deg)
ラジアンモード	RAD	(Rad)
グラードモード	GRA, GRAD	(Gra)
2 進数表示	BIN	(Bin)
8 進数表示	OCT	(Oct)
符号つき 10 進数表示	SDEC	(Sdec)
符号なし 10 進数表示	UDEC	(Udec)
16 進数表示	HEX	(Hex)
8 ビットモード	BYTE	(Byte)
16 ビットモード	WORD	(Word)
32 ビットモード	DWORD	(Dword)
64 ビットモード	QWORD	(Qword)
型表示	TYPE	
レジスタ表示	REG	[Reg]
オイラー表示	EULER, EUL	[Eul]
$\pi$ ラジアンの偏角表示	PIRAD, PRAD	[Eul(Pi)]
帯分数表示	FRACTION, FRAC	[i.a/b]
通常表示	STD	Std
固定小数点表示	FIX	Fix
指数表示	SCI	Sci
エンジニアリング表示	ENG	Eng
桁数設定	DISP, DIGIT	

### 3.4 スタックのページめくり

スタックにたくさんのデータがあると、下図のように、一部が表示されなくなります。

Std: 9/15, Stack: 11, History: 0/10		
=====		
#	TYPE :	VALUE
^	.....	..... ^
6:	Integer :	6
5:	Integer :	7
4:	Integer :	8
Z:	Integer :	9
Y:	Integer :	10
X:	Integer :	11
-----		
PUSH Integer		
Ready to operate		
-----		

画面に表示されていない位置のデータを見たい場合、スタックのページめくりの機能を使います。上図では、11 個のデータがあるのに対し、6 番目までしか表示されていません。さらに後ろのデータを表示させるには、スタックのページめくりの機能を使います。

キーワード「Cmd NEXT」または「Cmd N」を単体で入力すると、次のページが表示できます。

Std: 9/15, Stack: 11, History: 0/10		
=====		
#	TYPE :	VALUE
-:	:	
11:	Integer :	1
10:	Integer :	2
9:	Integer :	3
8:	Integer :	4
7:	Integer :	5
v	.....	..... v
NEXT PAGE of STACK		
Ready to operate		
-----		

逆に、前のページに戻るには、キーワード「Cmd PREV」または「Cmd P」を単体で入力します。最初のページ、つまり末尾を表示させるには、キーワード「Cmd FIRST」または「Cmd FST」を単体で入力します。なお、スタックが変更される機能が実行されると、自動的に最初のページに戻ります。これらをまとめると、次のようになります。

機能	キーワード
スタックの次のページ	NEXT, N
スタックの前のページ	PREV, P
スタックの最初のページ	FIRST,
	FST



## 3.5 レジスタのページめくり

本ソフトウェアのレジスタ機能は、値を再利用するための退避場所として利用できます。レジスタは RA～RZ の 26 本を利用できますが、レジスタ表示機能では、一度にすべてのレジスタを表示できないことがあります。

下図の場合、RA～RC は表示されていますが、それ以外は見えません。

Std: 9/15, Stack: 3, History: 0/10		
#	TYPE :	VALUE
RA:	Floating :	3.14159265
RB:	:	
RC:	:	
Z:	Integer :	4
Y:	Integer :	5
X:	Integer :	6
NEXT PAGE of STACK		
Ready to operate		

レジスタ表示にも、スタック表示と同様、ページめくりがあります。

レジスタの次のページを表示させるには、「[Cmd. REGNEXT](#)」または「[Cmd. RN](#)」を単体で入力します。

Std: 9/15, Stack: 3, History: 0/10		
#	TYPE :	VALUE
RD:	:	
RE:	:	
RF:	:	
Z:	Integer :	4
Y:	Integer :	5
X:	Integer :	6
NEXT PAGE of REGISTERS		
Ready to operate		

逆に、前のページに戻るには、キーワード「[Cmd. REGPREV](#)」または「[Cmd. RP](#)」を単体で入力します。

最初のページ、つまり RA を表示させるには、キーワード「[Cmd. REGFIRST](#)」または「[Cmd. RF](#)」を単体で入力します。

これらをまとめると、次のようになります。

機能	キーワード
レジスタの次のページ	REGNEXT, RN
	次のページに続く

前ページからの続き

機能	キーワード
レジスタの前のページ	REGPREV, RP
レジスタの最初のページ	REGFIRST, RF

### 3.6 値の全体表示

値の表示が長すぎると、画面幅に収まらない場合があります。下図では、分母と分子がそれぞれ大きな有理数を含む複素数の表示が省略されています。

#	TYPE	:	VALUE
-:	:		
-:	:		
-:	:		
Z:	:		
Y:	:		
X: Complex	:		2432902008176640000/243290200817664000...



このような場合、値の全体表示を利用します。キーワード「CmndVIEW」または「CmndV」で、値の全体表示の画面に移行します。

機能	キーワード
値の全体表示	VIEW, V

----- ----- STACK VIEW =====
X: Complex: 2432902008176640000/2432902008176640001 + i24329020081766400 00/2432902008176640001
----- (Press Return or Enter) >

全体表示となる対象は、スタック及びレジスタで直前に表示されているものです。つまり、ページめくりなどを利用してれば、直前に表示されているデータが省略されずに表示されます。Enter キーを押すと、全体表示から計算モード画面に戻ります。

## 3.7 バージョン表示

キーワード「 VER」または「 VERSION」で実行中のバージョンを表示できます。バージョン表示画面で Enter キーを押すと、直前のモードに復帰します。

機能	キーワード
バージョン表示	VERSION, VER

不具合など、気になる点がある場合は、バージョンを添えて報告ください。



## Chapter 4

# 基本操作～四則演算

本ソフトウェアは逆ポーランド記法を採用しているため、操作が特殊です。本章は飛ばさずにお読みください。

### 4.1 基本的なスタック操作

まずは、何か整数を入力してみましょう。

**Cmd** 12

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	12

スタックの  $X$  に 12 が追加されます。

次に、もう一度別の整数を入力してみましょう。

**Cmd** 9

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	12
X:	Integer	:	9

スタックの  $X$  に 9 が追加されます。

このように、データの追加は、スタック表示領域の  $X$  に行われます。

今度は、小数を入力してみましょう。スペース区切りで複数入力することもできます。

**Cmd** 1.6 6.0e-23

#	TYPE	:	VALUE
4:	Integer	:	12
Z:	Integer	:	9
Y:	Floating	:	1.6
X:	Floating	:	6E-23

このように、数をスタックに追加するには、数をそのまま書きます。スタックの末尾にデータを追加することをプッシュと言います。

不要なデータを削除したい場合、「Cmd DROP」または「Cmd \」と入力します。これで、末尾1つのデータを削除できます。これをドロップと言います。

Cmd \

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	12
Y:	Integer	:	9
X:	Floating	:	1.6

なお、計算モードで、設定の入力待ちでない場合、空の状態 Enter キーを押すと、X のコピーが実行されます（X と同じデータがプッシュされます）。

キーワード「Cmd DUP」でも同じ機能が実行されます。

Cmd （そのまま Enter）

#	TYPE	:	VALUE
4:	Integer	:	12
Z:	Integer	:	9
Y:	Floating	:	1.6
X:	Floating	:	1.6

また、スタックを一掃したい場合、「Cmd CLEAR」または「Cmd CLR」と入力します。

Cmd clear

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:		:	

説明した機能を表にまとめます。

機能	キーワード	R	D
プッシュ		0	0
ドロップ	DROP, \	1	1
コピー※	DUP	1	1
クリア	CLEAR, CLR	N>0	N

※入力欄に何も入力しない状態で Enter キーを押下しても実行されます  
プッシュ、ドロップ、コピー、クリアの操作が理解できたら、四則演算に移ります。

## 4.2 四則演算

四則演算の操作があらゆる計算機能の使い方の基本となります。まずはこれらの使い方をおさえましょう。

四則演算を行うには、次のキーワードを用います。詳しい表の読み方は後述します。

機能	キーワード	R	D	演算内容
加算	ADD, +	2	2	$Y + X$
減算	SUB, -	2	2	$Y - X$
乗算	MUL, *	2	2	$Y \times X$
除算	DIV, /	2	2	$Y \div X$
剰余	MOD, %	2	2	$Y \bmod X$
商と剰余	QM	2	2	$Y \leftarrow Y \div X$ $X \leftarrow Y \bmod X$

これらを用いた計算例を示していきます。

まずは、「2 + 3」を計算します。最初に、計算に使いたい数を入力します。

**Cmd.** 2 3

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	2
X:	Integer	:	3

**Cmd.** +

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	5

X には、 $Y + X$  を計算した結果の 5 が残ります。直前の Y と X はドロップされています。この入力は、「2 と 3 をプッシュし、足し算する」という意味です。

これに続けて、次のように入力してみましょう。

**Cmd.** 9 -

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	-4

X の表示は -4 となります。この入力は、「9 をプッシュし、引き算する」という意味で

す。このように、本ソフトウェアは、スタックの末尾を使って計算します。

機能	キーワード	R	D	演算内容
加算	ADD, +	2	2	$Y + X$

上表の R は必要とされるデータ数で、実行時に足りない場合はエラーが表示されます。

D は、ドロップされるデータ数です。演算内容が最後にプッシュされます。

加算は、2 つのデータが必要であり、実行すると 2 つがドロップされ、 $Y + X$  の結果がプッシュされる、と読みます。他の四則演算も同様です。

ちなみに、実行しようとしている機能に対して、スタックのデータ数が足りない場合は、下図のようにエラーが表示されます。

```

Engineering Calculator with KEyboard and Refined Tools
(C) 2014-2019 Yuishin Kikuchi
-----
HOMURA: (FD) (Rad) (Hex) (Dword)
Std: 9/15, Stack: 1, History: 0/8
=====
#  TYPE      :                               VALUE
-----
-:          :
-:          :
-:          :
Z:          :
Y:          :
X: Integer   :                               5
-----
[!] ADD Y+X
Error: Too few arguments
-----
>

```

「Too few arguments」というメッセージは、「データ数が足りず、実行できなかった」という意味です。

### 4.3 複合的な四則演算

もう少し複雑な例を見てみましょう。

台形の面積を計算してみます。上底を 2、下底を 1、高さを 5 とする台形を考えてみましょう。計算するのは、次の式です。

$$5 \times (2 + 1) \div 2$$

まず、「5 と  $(2 + 1)$  との掛け算」と読みます。これを求めるには、5 をプッシュしておいて、「 $2 + 1$ 」を計算し、掛け算を行います。あとは、これを 2 で割り算します。

これを一度に記述すれば、次のようになります。

**Cmd.** 5 2 1 + \* 2 /

ただ、説明のため、これを分けて入力します。次に示す (1)～(5) の手順に従って、読みながら操作すると理解しやすいでしょう。

(1) 整数の 5、2、1 をプッシュ

**Cmd.** 5 2 1



#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	5
Y:	Integer	:	2
X:	Integer	:	1

(2) 加算を実行

 +

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	5
X:	Integer	:	3

(3) 乗算を実行

 \*

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	15

(4) 整数の 2 をプッシュ

 2

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	15
X:	Integer	:	2

(5) 除算を実行

 /

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Rational	:	7.5

プッシュと計算命令を適切な順番で実行することにより、カッコを使うことなく、所望の計算が行えます。

## 4.4 四則演算以外の基本的な演算

四則演算以外の基本的な演算として、次のものを説明します。

機能	キーワード	R	D	演算内容
インクリメント	INC, ++	1	1	$X + 1$
デクリメント	DEC, --	1	1	$X - 1$
絶対値	ABS	1	1	$ Y \times X $
符号反転	PM, NEG	1	1	$-X$
逆数・逆行列	INV	1	1	$X^{-1}$

インクリメント・デクリメントは整数に対してのみ使える機能で、インクリメントは1増やす、デクリメントは1減らす機能です。

たとえば、5の逆数を求めるには、次のように入力します。

**Cmd.** 5 inv

これらは、1つに対して計算を行います。そのため、上表に示す演算のように、計算に1つのデータが必要とし、結果が1つの場合は、1つのデータが削除され、1つの結果がスタックにプッシュされます。

## 4.5 逆ポーランド表記法に慣れるコツ

スタックを使いながら計算を行うと、計算命令が後置されます。このように、演算命令（または演算子）が被演算子の後ろに置かれる記法を、逆ポーランド記法と呼びます。

この記法を用いると、並べる順番で計算のタイミングが決定されるため、優先順位が問われません。したがって、カッコを用いる必要がないのです。また、逆ポーランド記法の記述を日本語で読み上げると、直感的に理解しやすい場合があります。

**Cmd.** 2 1 -

この入力を、「2と1で引き算」と読むと、結果が理解できます。

このように「～を（で）……する」という順番で読むことができるので、これに当てはめると、逆ポーランド記法が浮かびやすくなります。

たとえば、「5と6で掛け算して、符号反転」ならば次のようになります。

**Cmd.** 5 6 \* pm

この考え方は、四則演算以外に対しても同様に適用できます。

# Chapter 5

## 数学関数

### 5.1 数学関数の使い方

本ソフトウェアは多くの数学関数に対応します。基本的な注意は四則演算と同様で、スタックにデータをプッシュした後に演算を実行します。

ただし、関数の中には、定義域が制限されているものもあります。また、数学的に定義可能な値の入力にも対応しない場合があります。あらかじめご了承ください。

### 5.2 指数・対数

指数・対数は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
2 乗	SQ	1	1	$X^2$
平方根	SQRT	1	1	$\sqrt{X}$
立方根	CBRT	1	1	$\sqrt[3]{X}$
直角三角形の斜辺	HYPOT	2	2	$\sqrt{ Y ^2 +  X ^2}$
冪乗	POW, ^, **	2	2	$Y^X$
冪乗根	NRT	2	2	$\text{sqrt}[X]Y$
指数関数	EXP	1	1	$\exp(X)$
10 の冪乗	TPOW	1	1	$10^X$
2 の冪乗	BPOW	1	1	$2^X$
Y に対する X の対数	LOGB	2	2	$\log_Y(X)$
自然対数	LN	1	1	$\log_e(X)$
常用対数	LOG	1	1	$\log_{10}(X)$
2 進対数	LB	1	1	$\log_2(X)$

計算例 1  $\log_{10} 3000$

`Cmd` 3000 log

計算例 2  $\sqrt{5^2 + 12^2}$

`Cmd` 5 sq 12 sq + sqrt

計算例 3  $\log_3 22$

`Cmd` 3 22 logb

計算例 4  $\exp(-3/2)$

`Cmd` 3 sq 2 / pm exp

### 5.3 三角関数・逆三角関数

三角関数・逆三角関数は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
サイン	SIN	1	1	$\sin(X)$
コサイン	COS	1	1	$\cos(X)$
タンジェント	TAN	1	1	$\tan(X)$
アークサイン	ASIN	1	1	$\arcsin(X)$
アークコサイン	ACOS	1	1	$\arccos(X)$
アークタンジェント	ATAN	1	1	$\arctan(X)$

上表の関数は角度モードに依存します。本ソフトウェアでは、角度モード非依存の三角関数のキーワードも利用できます。

ラジアン三角関数を示します。モード依存のキーワードの後ろに「R」をつけています。

機能	キーワード	R	D	演算内容
サイン (rad)	SINR	1	1	$\sin(X)[\text{rad}]$
コサイン (rad)	COSR	1	1	$\cos(X)[\text{rad}]$
タンジェント (rad)	TANR	1	1	$\tan(X)[\text{rad}]$
アークサイン (rad)	ASINR	1	1	$\arcsin(X)[\text{rad}]$
アークコサイン (rad)	ACOSR	1	1	$\arccos(X)[\text{rad}]$
アークタンジェント (rad)	ATANR	1	1	$\arctan(X)[\text{rad}]$

度数法の三角関数を示します。モード依存のキーワードの後ろに「D」をつけています。

機能	キーワード	R	D	演算内容
サイン (deg)	SIND	1	1	$\sin(X)[\text{deg}]$
コサイン (deg)	COSD	1	1	$\cos(X)[\text{deg}]$
タンジェント (deg)	TAND	1	1	$\tan(X)[\text{deg}]$
アークサイン (deg)	ASIND	1	1	$\arcsin(X)[\text{deg}]$
アークコサイン (deg)	ACOSD	1	1	$\arccos(X)[\text{deg}]$
アークタンジェント (deg)	ATAND	1	1	$\arctan(X)[\text{deg}]$

グラードの三角関数を示します。モード依存のキーワードの後ろに「G」をつけています。

機能	キーワード	R	D	演算内容
サイン (gra)	SING	1	1	$\sin(X)[\text{gra}]$
コサイン (gra)	COSG	1	1	$\cos(X)[\text{gra}]$
タンジェント (gra)	TANG	1	1	$\tan(X)[\text{gra}]$
アークサイン (gra)	ASING	1	1	$\arcsin(X)[\text{gra}]$
アークコサイン (gra)	ACOSG	1	1	$\arccos(X)[\text{gra}]$
アークタンジェント (gra)	ATANG	1	1	$\arctan(X)[\text{gra}]$

計算例 1  $\sin(30)$  (モード依存)

Cmd. 30 sin

計算例 2  $\cos(52^\circ)$

Cmd. 52 tand

## 5.4 双曲線関数・逆双曲線関数

双曲線関数・逆双曲線関数は次のキーワードを用います。

機能	キーワード	R	D	演算内容
ハイパーボリックサイン	SINH	1	1	$\sinh(X)$
ハイパーボリックコサイン	COSH	1	1	$\cosh(X)$
ハイパーボリックタンジェント	TANH	1	1	$\tanh(X)$
インバースハイパーボリックサイン	ASINH	1	1	$\operatorname{asinh}(X)$
インバースハイパーボリックコサイン	ACOSH	1	1	$\operatorname{acosh}(X)$
インバースハイパーボリックタンジェント	ATANH	1	1	$\operatorname{atanh}(X)$

計算例  $\cosh(1.2)$

Cmd. 1.2 cosh

## 5.5 統計関連の関数

統計に関連する関数は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ベータ関数	BETA	2	2	$B(Y, X)$
ガンマ関数	GAMMA	1	1	$\Gamma(X)$
ガンマ関数の絶対値の対数	LNGAMMA	1	1	$\log_e  \Gamma(X) $
誤差関数	ERF	1	1	$\operatorname{erf}(X)$
1 と誤差関数の差	ERFC	1	1	$1 - \operatorname{erf}(X)$

計算例 1  $B(0.5, 1.6)$

Cmd. 0.5 1.6 beta

計算例 2  $\Gamma(2)$

Cmd. 2 gamma

## 5.6 整数丸め関数

整数丸め演算は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
床関数	FLOOR, FLR	1	1	$\lfloor X \rfloor$
天井関数	CEIL	1	1	$\lceil X \rceil$
四捨五入	ROUND, RND	1	1	$\lfloor X + 0.5 \rfloor$

計算例 1  $\lfloor -2.2 \rfloor$

`Cmd. -2.2 flr`

計算例 2  $\lceil \pi \rceil$

`Cmd. pi ceil`

## 5.7 整数の関数

最大公約数、最小公倍数などの整数の関数は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
階乗	FACT, !	1	1	$X!$
最大公約数	GCD	1	1	$\text{GCD}(Y, X)$
最小公倍数	LCM	1	1	$\text{LCM}(Y, X)$
パーミュテーション	PERM	1	1	$P(Y, X)$ $= \frac{Y!}{(Y-X)!}$
コンビネーション (二項係数)	COMB	1	1	$C(Y, X)$ $= \frac{Y!}{X!(Y-X)!}$

計算例 1  $P(5, 2)$

`Cmd. 5 2 perm`

計算例 2  $\text{LCM}(12, 50)$

`Cmd. 12 50 lcm`

## Chapter 6

# 便利な計算機能

## 6.1 パーセント計算

税率などのパーセント計算を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
Y の X パーセント※	PERC, PC	2	1	$Y \times X/100$
Y に対する X の相対誤差				
パーセント表示	DPERC, CP	2	2	$\frac{X - Y}{Y \times 100}$
Y の X パーセント税込み計算	INTAX	2	2	$\frac{Y \times (100 + X)}{100}$
Y の X パーセント税抜き計算	EXTAX	2	2	$\frac{Y \times 100}{100 + X}$

※この機能では、X のみがドロップされ、元の Y は残ります

パーセント計算はスカラー（整数、浮動小数点数、有理数）にのみ有効です。

計算例 1 5.15 の 3%

`Cmd 5.15 3 pc`

計算例 2 1.2 に対する 1.3 の相対誤差

`Cmd 1.2 1.3 dp`

計算例 3 1250 円に 8% の課税

`Cmd 1250 8 intax`

計算例 4 税込み 120 円から 8% 税抜き

`Cmd 120 8 extax`

## 6.2 時間換算

秒、分、時、日、週の換算を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
秒 から 分	STOM	1	1	$X/60$
秒 から 時	STOH	1	1	$X/3600$
秒 から 日	STOD	1	1	$X/86400$
秒 から 週	STOW	1	1	$X/604800$
分 から 秒	MTOS	1	1	$X \times 60$
分 から 時	MTOH	1	1	$X/60$
分 から 日	MTOD	1	1	$X/1440$
分 から 週	MTOW	1	1	$X/10080$
時 から 秒	HTOS	1	1	$X \times 3600$
時 から 分	HTOM	1	1	$X \times 60$
時 から 日	HTOD	1	1	$X/24$
時 から 週	HTOW	1	1	$X/168$
日 から 秒	DTOS	1	1	$X \times 86400$
日 から 分	DTOM	1	1	$X \times 1440$
日 から 時	DTOH	1	1	$X \times 24$
日 から 週	DTOW	1	1	$X/7$
週 から 秒	WTOS	1	1	$X \times 604800$
週 から 分	WTOM	1	1	$X \times 10080$
週 から 時	WTOH	1	1	$X \times 168$
週 から 日	WTOD	1	1	$X \times 7$

時間換算はスカラー（整数、浮動小数点数、有理数）にのみ有効です。

計算例 2 45 分を時に換算

`45 mtoh`

計算例 1 65536 秒を日に換算

`65536 stod`



### 6.3 DMS 変換・逆変換

DMS 変換は、10 進数の 1 つの数値を度・分・秒の 60 進数表示に分解します。DMS 逆変換は、3 つの数値を度・分・秒とみなし、1 つの数値に変換します。DMS 変換・逆変換を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
10 進数から DMS	TODMS	1	1	$Z \leftarrow D$ $Y \leftarrow M$ $X \leftarrow S$
DMS から 10 進数	DMSTO	3	3	$Z + \frac{Y}{60} + \frac{X}{3600}$

DMS 変換・逆変換はスカラー（整数、浮動小数点数、有理数）にのみ有効です。

計算例 1 4096 秒を時分秒に換算

Cmd. 4096 stoh todms

計算例 2 30 度 20 分 10 秒を換算

Cmd. 30 20 10 dmsto

### 6.4 総和・平均など

スタック全体について、総和や平均を求めることができます。エラーなく求まると、スタックには結果だけが残ります。

機能	キーワード	R	D	演算内容
総和	SUM	N>1	N	$\sum_{i=1}^n x_i$
総積	PROD	N>1	N	$\prod_{i=1}^n x_i$
算術平均	AVR	N>1	N	$\frac{1}{n} \sum_{i=1}^n x_i$
幾何平均	GAVR	N>1	N	$\sqrt[n]{\prod_{i=1}^n x_i}$
調和平均	HAVR	N>1	N	$\frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$

途中で加算や乗算が定義できない組み合わせが検出されるとエラーになります。

また、個数指定やドロップをしないバリエーションもあります。

機能	キーワード	R	D	演算内容
総和（個数指定）	PSUM	N>2	M+1	
総積（個数指定）	PPROD	N>2	M+1	
算術平均（個数指定）	PAVR	N>2	M+1	
幾何平均（個数指定）	PGAVR	N>2	M+1	
調和平均（個数指定）	PHAVR	N>2	M+1	
総和（ドロップなし）	SUMW	N>1	0	
総積（ドロップなし）	PRODW	N>1	0	
算術平均（ドロップなし）	AVRW	N>1	0	

次のページに続く

前ページからの続き

機能	キーワード	R	D	演算内容
幾何平均（ドロップなし）	GAVRW	N>1	0	
調和平均（ドロップなし）	HAVRW	N>1	0	
総和（個数指定・ドロップなし）	PSUMW	N>2	1	
総積（個数指定・ドロップなし）	PPRODW	N>2	1	
算術平均（個数指定・ドロップなし）	PAVRW	N>2	1	
幾何平均（個数指定・ドロップなし）	PGAVRW	N>2	1	
調和平均（個数指定・ドロップなし）	PHAVRW	N>2	1	

## 6.5 接頭辞との乗算

接頭辞の値との乗算を行うと、接頭辞を取り除いた値に変換できます。たとえば、キロメートル表示をメートル表示に変換するには、キロの値である 1000 を乗算します。

接頭辞の値を乗算するには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ヨクトとの乗算	YOCTO	1	1	$X \times 10^{-24}$
zeptoとの乗算	ZEPTO	1	1	$X \times 10^{-21}$
アトとの乗算	ATTO	1	1	$X \times 10^{-18}$
フェムトとの乗算	FEMTO	1	1	$X \times 10^{-15}$
ピコとの乗算	PICO	1	1	$X \times 10^{-12}$
ナノとの乗算	NANO	1	1	$X \times 10^{-9}$
マイクロとの乗算	MICRO	1	1	$X \times 10^{-6}$
ミリとの乗算	MILLI	1	1	$X \times 10^{-3}$
センチとの乗算	CENTI	1	1	$X \times 10^{-2}$
デシとの乗算	DECI	1	1	$X \times 10^{-1}$
デカとの乗算	DECA	1	1	$X \times 10^{+1}$
ヘクトとの乗算	HECTO	1	1	$X \times 10^{+2}$
キロとの乗算	KILO	1	1	$X \times 10^{+3}$
メガとの乗算	MEGA	1	1	$X \times 10^{+6}$
ギガとの乗算	GIGA	1	1	$X \times 10^{+9}$
テラとの乗算	TERA	1	1	$X \times 10^{+12}$
ペタとの乗算	PETA	1	1	$X \times 10^{+15}$
エクサとの乗算	EXA	1	1	$X \times 10^{+18}$
ゼタとの乗算	ZETTA	1	1	$X \times 10^{+21}$
ヨタとの乗算	YOTTA	1	1	$X \times 10^{+24}$
キビとの乗算	KIBI	1	1	$X \times 2^{10}$
メビとの乗算	MEBI	1	1	$X \times 2^{20}$

前ページからの続き

機能	キーワード	R	D	演算内容
ギビとの乗算	GIBI	1	1	$X \times 2^{30}$
テビとの乗算	TEBI	1	1	$X \times 2^{40}$
ペビとの乗算	PEBI	1	1	$X \times 2^{50}$
エクシビとの乗算	EXBI	1	1	$X \times 2^{60}$
ゼビとの乗算	ZEBI	1	1	$X \times 2^{70}$
ヨビとの乗算	YOBI	1	1	$X \times 2^{80}$

## 6.6 接頭辞による除算

接頭辞の値で除算を行うと、接頭辞をつけた値に変換できます。たとえば、メートル表示をミリメートル表示に変換するには、ミリの値である 0.001 で除算します。

接頭辞の値で除算するには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ヨクトによる除算	TOYECTO	1	1	$X/10^{-24}$
zeptoによる除算	TOZEPTO	1	1	$X/10^{-21}$
アトによる除算	TOATTO	1	1	$X/10^{-18}$
フェムトによる除算	TOFEMTO	1	1	$X/10^{-15}$
ピコによる除算	TOPICO	1	1	$X/10^{-12}$
ナノによる除算	TONANO	1	1	$X/10^{-9}$
マイクロによる除算	TOMICRO	1	1	$X/10^{-6}$
ミリによる除算	TOMILLI	1	1	$X/10^{-3}$
センチによる除算	TOCENTI	1	1	$X/10^{-2}$
デシによる除算	TODECI	1	1	$X/10^{-1}$
デカによる除算	TODECA	1	1	$X/10^{+1}$
ヘクトによる除算	TOHECTO	1	1	$X/10^{+2}$
キロによる除算	TOKILO	1	1	$X/10^{+3}$
メガによる除算	TOMEGA	1	1	$X/10^{+6}$
ギガによる除算	TOGIGA	1	1	$X/10^{+9}$
テラによる除算	TOTERA	1	1	$X/10^{+12}$
ペタによる除算	TOPETA	1	1	$X/10^{+15}$
エクサによる除算	TOEXA	1	1	$X/10^{+18}$
ゼタによる除算	TOZETTA	1	1	$X/10^{+21}$
ヨタによる除算	TOYOTTA	1	1	$X/10^{+24}$
キビによる除算	TOKIBI	1	1	$X/2^{10}$
メビによる除算	TOMEBI	1	1	$X/2^{20}$
ギビによる除算	TOGIBI	1	1	$X/2^{30}$
テビによる除算	TOTEBI	1	1	$X/2^{40}$
ペビによる除算	TOPEBI	1	1	$X/2^{50}$
エクシビによる除算	TOEXBI	1	1	$X/2^{60}$

前ページからの続き

機能	キーワード	R	D	演算内容
ゼビによる除算	TOZEBI	1	1	$X/2^{70}$
ヨビによる除算	TOY0BI	1	1	$X/2^{80}$

## 6.7 角度変換

角度の単位を変換するには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ラジアン から 度数法	RT0D	1	1	$180X/\pi$
ラジアン から グラード	RT0G	1	1	$200X/\pi$
度数法 から ラジアン	DT0R	1	1	$\pi X/180$
度数法 から グラード	DT0G	1	1	$10X/9$
グラード から ラジアン	GT0R	1	1	$\pi X/200$
グラード から 度数法	GT0D	1	1	$9X/10$

## 6.8 角度計算

余角や補角を求めるには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
余角 ※	CANG	1	1	
余角 (ラジアン)	CANGR	1	1	$\pi/2 - X$
余角 (度数法)	CANGD	1	1	$90 - X$
余角 (グラード)	CANGG	1	1	$100 - X$
補角 ※	SANG	1	1	
補角 (ラジアン)	SANGR	1	1	$\pi - X$
補角 (度数法)	SANGD	1	1	$180 - X$
補角 (グラード)	SANGG	1	1	$200 - X$

※角度モードに依存します。

## 6.9 有理数の分解

$X$  が有理数のとき、 $Y$  に分子、 $X$  に分母を展開します。

機能	キーワード	R	D	演算内容
有理数の分解	RATIO	1	1	$Y \leftarrow \text{Numerator}$ $X \leftarrow \text{Denominator}$

## 6.10 乱数

乱数をプッシュするには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
整数乱数	RAND	0	0	Push Int
浮動小数点数乱数	FRAND	0	0	Push Flt

整数乱数は 63 ビット、浮動小数点数は整数乱数を元にして生成されます。乱数生成アルゴリズムはメルセンヌ・ツイスタを用いています。

## 6.11 キャスト

型変換を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
整数キャスト	TOINT	1	1	
浮動小数点数キャスト	TOFLT	1	1	
有理数キャスト	TORAT	1	1	
ブーリアンキャスト	TOBOOL	1	1	
8 ビットキャスト	TOBYTE	1	1	
16 ビットキャスト	TOWORD	1	1	
32 ビットキャスト	TODWORD	1	1	
64 ビットキャスト	TOQWORD	1	1	
16 ビットキャスト（符号拡張）	TOSWORD	1	1	
32 ビットキャスト（符号拡張）	TOSDWORD	1	1	
64 ビットキャスト（符号拡張）	TOSQWORD	1	1	

有理数キャストを用いることにより、浮動小数点数を有理数に近似できます。連分数近似によるアルゴリズムを用いています。

## 6.12 エンジン向け計算

エンジン向けの計算は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
$2\pi$ との乗算	TPIX	1	1	$2\pi X$
$2\pi$ による除算	DTPI	1	1	$X/2\pi$
並列	PARA	1	1	$(Y^{-1} + X^{-1})^{-1}$
デシベルへの変換	TODB	1	1	$10 \log_{10}  X $
デシベルからの変換	DBTO	1	1	$10^{X/10}$

## 6.13 地震に関する計算

地震のマグニチュードの換算は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ジュールから地震のマグニチュード	TOEQM	1	1	$(\log_{10}(X - 4.8))/1.5$
地震のマグニチュードからジュール	EQMT0	1	1	$10^{4.8+1.5X}$

## 6.14 健康や生活に関する計算

健康や生活に関する計算は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
不快指数	DISCOM	2	2	$0.81Y - 0.01X \times (0.99Y - 14.3) + 46.3$
BMI	BMI	2	2	$\frac{X}{(Y/100)^2}$

不快指数は、温度と湿度から、平均的に不快に感じる尺度を示すものです。Y にセルシウス度（celsius）を、X に相対湿度（%）を指定して計算します。

## Chapter 7

# 複素数の計算

## 7.1 複素数の表示

本ソフトウェアでは、複素数を次のように表示します。

モード	数式	画面表示
直交座標	$5 + 12i$	5 + i12
極座標（度数法）	$13 \angle 67 [\text{deg}]$	13 exp(+i67.d)
極座標（ラジアン）	$13 \angle 1.3 [\text{rad}]$	13 exp(+i1.3) 13 exp(+i0.41 Pi)
極座標（グラード）	$13 \angle 75 [\text{gra}]$	13 exp(+i75.g)

キーワード「Cmd EULER」または「Cmd EUL」を単体で用いると、複素数の表示形式を変更できます。実数部と虚数部の表示（直交座標）と絶対値と偏角の表示（極座標）を切り替えます。極座標表示は、オイラー表示とも呼びます。

絶対値と偏角の表示において、偏角の表示は角度モードに依存します。角度の単位を変えたい場合、キーワード「Cmd DEG」、「Cmd RAD」、「Cmd GRA」を使って、角度モードを変更します。

また、ラジアンモードのとき、さらに角度を $\pi$ ラジアンに直すことができます。 $\pi$ ラジアンモードの有効・無効を切り替えるには、「Cmd PIRAD」または「Cmd PRAD」を単体で入力します。

モード	キーワード	画面表示
オイラー表示	EULER, EUL	[Euł]
度数法モード	DEG	(Deg)
ラジアンモード	RAD	(Rad)
グラードモード	GRAD, GRA	(Gra)
$\pi$ ラジアンモード	PIRAD, PRAD	[Euł(Pi)]

また、スカラー（整数、浮動小数点数、有理数）についても、その値が0でなければ、偏角を表示します。

## 7.2 複素数の作り方

本ソフトウェアで複素数を入力するには、3通りの方法があります。

複素数の実部虚部に認められるのはスカラー（整数、浮動小数点数、有理数）のみです。

### 7.2.1 虚数を入力し、加減する

虚数を直接入力し、足し引きする方法です。では、 $2 + i3$  を作ってみましょう。

**Cmd.** 2 i3 +

### 7.2.2 実数部・虚数部を入力して複素数にする

Y に実数部、X に虚数部の順番で数をプッシュしておき、複素数を作ることができます。キーワード「**Cmd.** MKCMP」または「**Cmd.** MKC」でこの方法を使えます。

**Cmd.** 2 3

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	2
X:	Integer	:	3

**Cmd.** mkc

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Complex	:	2 + i3

### 7.2.3 絶対値と偏角から複素数にする

Y に絶対値、X に偏角の順番で数をプッシュしておき、複素数を作ることができます。キーワード「**Cmd.** MKE」でこの方法を使えます。

ただし、キーワード「**Cmd.** MKE」は角度モードに依存します。ここでは、度数法モードと仮定して、「 $1.5 \angle 30^\circ$ 」を求めます。

**Cmd.** 1.5 30

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Floating	:	1.5
X:	Integer	:	30



**Cmd** mke

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Complex	:	1.29903811 + i0.75

ラジアンによる生成は「**Cmd** MKER」、度数法による生成は「**Cmd** MKED」、グラードによる生成は「**Cmd** MKEG」です。いずれの使い方も「**Cmd** MKE」と同じです。

これらをまとめると、次のとおりになります。

機能	キーワード	R	D	演算内容
複素数の生成（直交座標）	MKCMP, MKC	2	2	$Y + iX$
複素数の生成（極座標）	MKE	2	2	$Y \angle X$
複素数の生成（極座標、ラジアン）	MKER	2	2	$Y \angle X [\text{rad}]$
複素数の生成（極座標、度数法）	MKED	2	2	$Y \angle X [\text{deg}]$
複素数の生成（極座標、グラード）	MKEG	2	2	$Y \angle X [\text{gra}]$

## 7.3 複素数の演算

実数部分、虚数部分、偏角、共軛など、複素数に特有の演算は、次のキーワードを用います。絶対値は、実数の場合と同様、キーワード「**Cmd** ABS」が使えます。

機能	キーワード	R	D	演算内容
実数部分	RE	1	1	$\text{Re}(X)$
虚数部分	IM	1	1	$\text{Im}(X)$
複素偏角	ARG	1	1	$\arg X$
複素偏角（ラジアン）	ARGR	1	1	$\arg X [\text{rad}]$
複素偏角（度数法）	ARGD	1	1	$\arg X [\text{rad}]$
複素偏角（グラード）	ARGG	1	1	$\arg X [\text{rad}]$
複素共軛	CONJ	1	1	$X^*$

計算例 1  $\arg(1 + i2)$

**Cmd** 1 2 mkc arg

計算例 2  $\text{Re}(15 \angle 32^\circ)$

**Cmd** 15 32 mked re

計算例 3  $(6 + i3)^*$

**Cmd** 6 3 mkc conj

## 7.4 複素数の分解

実数部と虚数部への分解、絶対値と偏角への分解は、次のキーワードを用います。

機能	キーワード	R	D	演算内容
実数部と虚数部	REIM	1	1	$Y \leftarrow \operatorname{Re}(X)$ $X \leftarrow \operatorname{Im}(X)$
絶対値と偏角	MAGA	1	1	$Y \leftarrow  X $ $X \leftarrow \arg X$
絶対値と偏角（ラジアン）	MAGAR	1	1	$Y \leftarrow  X $ $X \leftarrow \arg X [\text{rad}]$
絶対値と偏角（度数法）	MAGAD	1	1	$Y \leftarrow  X $ $X \leftarrow \arg X [\text{deg}]$
絶対値と偏角（グラード）	MAGAG	1	1	$Y \leftarrow  X $ $X \leftarrow \arg X [\text{gra}]$

計算例 1  $15\angle 32^\circ$  を実部虚部に分解

`15 32 mked reim`

計算例 2  $5 + i3$  の絶対値と偏角（度）

`5 3 mkc magad`

## 7.5 複素関数

本ソフトウェアは、次の関数を複素関数として使うことができます。複素関数の場合でも、キーワードは共通です。

- 平方根・立方根
- 指数・対数関数
- 三角関数
- 双曲線関数

ただし、三角関数の複素数の入力、ラジアンモードにおいてのみ有効です。

## Chapter 8

# 論理計算

### 8.1 符号なし整数・ブーリアンの表示

本ソフトウェアでは、符号なし整数とブーリアンの値を、次のように表示します。

モード	数式	画面表示
ブーリアン	TRUE	T
	FALSE	F
2 進数	255	0b11111111
8 進数	255	0377
符号つき 10 進数	255	-1
符号なし 10 進数	255	255
16 進数	255	0xFF

### 8.2 プッシュするビット長の設定

本ソフトウェアの論理計算は、他の計算と同様に計算モードを使います。

論理計算を始める前に、設定のビット長（N ビットモード）を確認しましょう。計算設定・状態表示領域に、下表の画面表示の列のいずれかが表示されます。

設定されているビット長が行いたい計算と合わない場合、ビット長を変更します。

モード	キーワード	画面表示
8 ビットモード	BYTE	(Byte)
16 ビットモード	WORD	(Word)
32 ビットモード	DWORD	(Dword)
64 ビットモード	QWORD	(Qword)

正しく設定されると、計算設定・状態表示領域の表示が変わります。スタックの表示もご確認ください。

なお、符号なし整数をプッシュする際、設定されているビット数に対してオーバーフローする値が入力されると、下位 N ビットのマスクが取られます。

### 8.3 符号なし整数の表示切り替え

進数表示の設定は、計算設定・状態表示領域で確認できます。下表の画面表示のいずれかが表示されています。

表示を切り替えたい場合、下表のキーワードを用います。

モード	キーワード	画面表示
2 進数表示	BIN	(Bin)
8 進数表示	OCT	(Oct)
符号つき 10 進数表示	SDEC	(Sdec)
符号なし 10 進数表示	UDEC	(Udec)
16 進数	HEX	(Hex)

正しく設定されると、計算設定・状態表示領域の表示が変わります。

### 8.4 符号なし整数・ブーリアンの入力

バイナリやブーリアンの計算を行うには、データを符号なし整数としてプッシュする必要があります。普通に整数を入力しても、符号なし整数の扱いにはなりません。

#### ブーリアン

真の場合「**Cmd** TRUE」「**Cmd** T」、偽の場合「**Cmd** FALSE」「**Cmd** F」と入力します。

#### 符号なし整数

「u」を前置し、直後に符号をつけずに整数を続けて入力します。

#### 2 進数

「0b」を前置し、直後に 0 と 1 による 2 進数表現を入力します。

#### 8 進数

「0o」を前置し、直後に 0～7 による 8 進数表現を入力します。

#### 16 進数

「0x」を前置し、直後に 0～9, A～F による 16 進数表現を入力します。

ただ、入力したデータは、選択されている表示モードで表示されます。

たとえば、16 進数表示モードのときに、2 進数で「0b1010」と入力すると、「0x0000000A」が表示されます。

**Cmd** 0b1010

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Dword	:	0x0000000A

バイナリやブーリアンも他の数と同様、一度に複数プッシュできます。

**Cmd** t f

#	TYPE	:	VALUE
4:		:	
Z:	Dword	:	0x0000000A
Y:	Boolean	:	T
X:	Boolean	:	F

## 8.5 基本的な論理演算

基本的な論理演算を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
論理否定 (NOT)	NOT, ~	1	1	$\overline{X}$
論理積 (AND)	AND, &	2	2	$Y \wedge X$
論理和 (OR)	OR,	2	2	$Y \vee X$
排他的論理和 (XOR)	XOR	2	2	$Y \oplus X$
否定論理積 (NAND)	NAND	2	2	$\overline{(Y \wedge X)}$
否定論理和 (NOR)	NOR	2	2	$\overline{(Y \vee X)}$

計算例 1    0x1234 & 0b0111

Cmd. 0x1234 0b0111 and

計算例 2    not(65535)

Cmd. u65535 not

## 8.6 ビットシフト

符号なし整数のシフトを行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
左シフト	SHL, <<	1	1	$X \ll 1$
論理右シフト	SHR, >>	1	1	$X \gg 1$
算術右シフト	SAR, >>>	1	1	$X \ggg 1$
N ビット左シフト	SHLC	2	2	$X \ll N$
N ビット論理右シフト	SHRC	2	2	$X \gg N$
N ビット算術右シフト	SARC	2	2	$X \ggg N$
8 ビット左シフト	SBL	1	1	$X \ll 8$
8 ビット右シフト	SBR	1	1	$X \gg 8$
4 ビット左シフト	SNL	1	1	$X \ll 4$
4 ビット右シフト	SNR	1	1	$X \gg 4$

計算例 1    0x1234 & 0b0111

Cmd. 0x1234 0b0111 and

計算例 2    not(65535)

Cmd. u65535 not

## 8.7 ローテイト

ローテイトを行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
左ローテイト	ROL	1	1	Rotate $X$ Left
右ローテイト	ROR	1	1	Rotate $X$ Right

計算例    `rol(31)`

`u31 rol`

## 8.8 その他符号なし整数に使える機能

機能	キーワード	R	D	演算内容
インクリメント	INC, ++	1	1	$X + 1$
デクリメント	DEC, --	1	1	$X - 1$
加算	ADD, +	2	2	$Y + X$
減算	SUB	2, -	2	$Y - X$
乗算	MUL	2, *	2	$Y \times X$
除算	DIV	2, /	2	$Y / X$
符号反転	NEG, PM	1	1	$-X$

ブーリアン同士の加算は排他的論理和に、乗算は論理積になります。また、ブーリアンに対してインクリメントを実行すると、必ず TRUE になります。

## 8.9 論理演算の積算

論理積や論理和についても、スタック全体を使って計算する機能があります。

機能	キーワード	R	D	演算内容
総論理積	ALLAND	$N > 1$	$N$	$x_1 \wedge x_2 \dots$
総論理和	ALLOR	$N > 1$	$N$	$x_1 \vee x_2 \dots$
総排他的論理和	ALLXOR	$N > 1$	$N$	$x_1 \oplus x_2 \dots$

# Chapter 9

## ベクトルの計算

### 9.1 ベクトルの表示

本ソフトウェアでは、ベクトルの表示を次のようにします。

モード	数式	画面表示
横 [Row]	$[1 \ 2 \ 3]$	$[1, 2, 3]$
縦 (Col)	$\begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$

### 9.2 ベクトルの作りかた

実数のベクトルはもちろん、複素数や符号なし整数などが含まれていても計算できます。ベクトルの入力には操作数が多いため、レジスタ機能の利用をおすすめします。詳しくは、11章「レジスタ機能」をお読みください。

任意の要素を含むベクトルを作るには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ベクトルの生成（横方向）	MRTUP	N	N	Push Tup.R
ベクトルの生成（縦方向）	MCTUP	N	N	Push Tup.C

任意のベクトルを作るには、3つの手順を踏みます。

#### あらかじめ要素をプッシュ

ベクトルの要素にしたい数を順番にプッシュします

#### 要素数（次元）を整数で指定

1以上の整数でベクトルの次元を指定します

#### ベクトルの生成機能を使う

縦方向または横方向で、指定された要素数にしたがってベクトルを作ります

ベクトルの要素にベクトルや行列を含むことはできません。ベクトルの要素として認められるのは、整数、浮動小数点数、有理数、複素数、ブーリアン、符号なし整数です。

では、横方向ベクトル  $[1 + i2 \ 6]$  を作ってみましょう。

## (1) 要素をプッシュ

**Cmd.** 1 2 mkc 6

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Complex	:	1 + i2
X:	Integer	:	6

## (2) 要素数（次元）をプッシュ

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:	Complex	:	1 + i2
Y:	Integer	:	6
X:	Integer	:	2

## (3) 横方向ベクトルの生成

**Cmd.** mrtup

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Tuple[Row]	:	[1 + i2, 6]

縦方向ベクトルについても、作り方は同様です。

単位ベクトルを作ることができます。

機能	キーワード	R	D	演算内容
単位ベクトルの生成（横方向）	MRUTUP	2	2	Push Tup.R
単位ベクトルの生成（縦方向）	MCUTUP	2	2	Push Tup.C

単位ベクトルを作るには、2つの整数をプッシュする必要があります。

- 要素数として整数をプッシュ
- 1にする要素の位置として整数をプッシュ（最初の要素は1番目とする）
- 単位ベクトルの生成機能を使う

では、3次元の縦方向単位ベクトル (0 1 0) を作ってみましょう。

## (1) 要素数（次元）をプッシュ

**Cmd.** 3



#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	3

(2) 1 にしたい位置の整数をプッシュ

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	3
X:	Integer	:	2

(3) 縦方向単位ベクトルの生成

**Cmd.** mcutup

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Tuple(Col)	:	(0, 1, 0)

## 9.3 ベクトルからの抽出

ベクトルから要素を抽出するには、キーワード「**Cmd.** TGET」を用います。

Y にベクトルを、X に抽出したい要素の位置（整数）をプッシュしておき、実行します。  
要素の位置は 1 から数えます。

機能	キーワード	R	D	演算内容
ベクトルからの要素の抽出	TGET	2	2	Extract

この機能では、抽出元のベクトルもドロップされます。抽出元のベクトルを残したい場合、レジスタ機能をご利用ください。「レジスタにベクトルを登録しておき、それを呼び出して抽出する」という流れになります。

レジスタの利用方法については、11 章「レジスタ機能」をお読みください。

また、ベクトルそのものを分解することもできます。

機能	キーワード	R	D	演算内容
ベクトルの要素への分解	CUT, CRAVE	1	1	

分解元のデータはドロップされ、分解された要素は順番通りにプッシュされます。

では、3 次元のベクトル (6 9 12) から、2 番目の要素を抽出してみましょう。

(0) あらかじめベクトルがある状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Tuple[Col]	:	(6, 9, 12)

(1) 抽出したい要素の位置を指定

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Tuple[Col]	:	(6, 9, 12)
X:	Integer	:	2

(2) 要素を抽出

**Cmd.** tget

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	9

## 9.4 ベクトルを含む四則演算

ベクトル同士の加減算やベクトルと数との乗除算を行う場合でも、数同士の四則演算と同じキーワードを使います。

**計算例 1**  $(3 \ 2 \ 1) + (5 \ 6 \ 9)$

**Cmd.** 3 2 1 3 mctup

**Cmd.** 5 6 9 3 mctup

**Cmd.** +

**計算例 2**  $(3 \ 2 \ 1) \times 9$

**Cmd.** 3 2 1 3 mctup 9 \*

ベクトル同士であっても、次元の違いにより、計算が定義できない場合があります。その場合、エラーが表示されます。

## 9.5 内積・外積

ベクトルの内積・外積を計算するには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
内積	INNER, DOT	2	2	$\vec{Y} \cdot \vec{X}$

次のページに続く

前ページからの続き

機能	キーワード	R	D	演算内容
外積	OUTER, CROSS	2	2	$\vec{Y} \times \vec{X}$

内積は、同じ次元数のベクトル、外積は、両方が 3 次元のベクトルの必要があります。

計算例 1 (3 2 1) · (7 8 9)

`Cmd. 3 2 1 3 mctup`

`Cmd. 7 8 9 3 mctup`

`Cmd. dot`

計算例 2 (1 2 3) × (4 5 6)

`Cmd. 1 2 3 3 mctup`

`Cmd. 4 5 6 3 mctup`

`Cmd. cross`

## 9.6 ベクトルノルム

ベクトルノルムを求めるには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ユークリッドノルム	NORM	1	1	$\sqrt{\sum_{i=1}^{\infty}  x_i ^2}$
ユークリッドノルムの 2 乗	NSQ	1	1	$\sum_{i=1}^{\infty}  x_i ^2$
p 次平均ノルム	PNORM	2	2	$(\sum_{i=1}^{\infty}  y_i ^x)^{1/x}$
最大値ノルム	MAXNORM	1	1	$\max( x_1 , \dots,  x_n )$

ユークリッドノルム、ユークリッドノルムの 2 乗、最大値ノルムは、ベクトルを 1 つ用意して利用します。たとえば、[3 5 7] のユークリッドノルムを求めるには、次のように入力します。

`Cmd. 3 5 7 3 mrtup norm`

p 次平均ノルムを求めるには、ベクトルと、次元にあたる整数をプッシュして利用します。たとえば、[7 8 9] の 3 次平均ノルムを求めるには、次のように入力します。

`Cmd. 7 8 9 3 mrtup 3 pnorm`

## 9.7 ベクトルの転置

ベクトルの転置を行うには、キーワード「`Cmd. TRANS`」を用います。

機能	キーワード	R	D	演算内容
転置	TRANS	1	1	$X^T$

このキーワードは、行列に対しても同じです。



# Chapter 10

## 行列の計算

### 10.1 行列の表示

行列は、次のように、横方向ベクトル（行ベクトル）の並びとして表示します。

モード	数式	画面表示									
-	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
1	2	3									
4	5	6									
7	8	9									

### 10.2 行列の作りかた

本ソフトウェアでは、行列の計算を行えます。実数の行列はもちろん、複素数や符号なし整数などが含まれていても計算できます。

行列の入力はベクトルと同様、操作数が多いため、レジスタ機能の利用をおすすめします。詳しくは、11 章「レジスタ機能」をお読みください。

任意の要素を含む行列を作るには、キーワード「Cmn MKMAT」を用います。

機能	キーワード	R	D	演算内容
行列の生成	MKMAT	N	N	Push Mat

任意の行列を作るには、3 つの手順を踏みます。

#### あらかじめ同方向・同次元のベクトルをプッシュ

横方向なら行ベクトルで、縦方向なら列ベクトルで行列を作ります

#### 要素数（次元）を整数で指定

1 以上の整数で束ねるベクトルの個数を指定します

#### 行列の生成機能を使う

ベクトルを束ねて行列を作ります

行列は横方向ベクトルの束として管理されています。ベクトルに含むことができないデータは、行列にも含むことができません。

では、例として、次の行列 A を入力します。2 種類の方法を示します。

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

(1-1) 横方向ベクトル 2 つをプッシュ

**Cmd.** 1 2 2 mrtup 3 4 2 mrtup

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Tuple[Row]	:	[1, 2]
X:	Tuple[Row]	:	[3, 4]

(1-2) ベクトルを束ねる個数をプッシュ

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:	Tuple[Row]	:	[1, 2]
Y:	Tuple[Row]	:	[3, 4]
X:	Integer	:	2

(1-3) 行列の生成

**Cmd.** mkmat

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Matrix	:	[[1, 2], [3, 4]]

(2-1) 縦方向ベクトル 2 つをプッシュ

**Cmd.** 1 3 2 mctup 2 4 2 mctup

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Tuple(Col)	:	(1, 3)
X:	Tuple(Col)	:	(2, 4)

(2-2) ベクトルを束ねる個数をプッシュ

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:	Tuple(Col)	:	(1, 3)
Y:	Tuple(Col)	:	(2, 4)
X:	Integer	:	2

## (2-3) 行列の生成

**Cmd.** mkmat

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Matrix	:	[[1, 2], [3, 4]]

異なる方向、または異なる次元のベクトルが含まれると、エラーが表示されます。

また、単位行列を作ることもできます。単位行列を作るには、キーワード「**Cmd.** MKUMAT」を用います。

機能	キーワード	R	D	演算内容
単位行列の生成	MKUMAT	1	1	Push Mat

次元数を指定して、単位行列の生成機能を使います。たとえば、3次元の単位行列を作るには、次のように入力します。

**Cmd.** 3 mkumat

## 10.3 行列からの抽出

行列から行ベクトルや列ベクトルまたは単一の要素を取り出すには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
行列からの単一要素の抽出	MGET	3	3	Extract
行列からの行ベクトルの抽出	MGETR	2	2	Extract
行列からの列ベクトルの抽出	MGETC	2	2	Extract

この機能では、抽出元の行列もドロップされます。抽出元の行列を残したい場合、レジスタ機能をご利用ください。「レジスタに行列を登録しておき、それを呼び出して抽出する」という流れになります。

レジスタの利用方法については、11章「レジスタ機能」をお読みください。

また、行列そのものを分解することもできます。

機能	キーワード	R	D	演算内容
行列の行ベクトルへの分解	CUT, CRAVE	1	1	

分解元のデータはドロップされ、分解された要素は順番通りにプッシュされます。次ページから、「**Cmd.** MGET」「**Cmd.** MGETR」「**Cmd.** MGETC」の使い方を説明します。

### 10.3.1 行列からの単一要素の抽出

行列から単一要素を抽出するには、キーワード「`MGET`」を用います。Zに行列を、Yに  
行の位置（整数）を、Xに列の位置（整数）をプッシュしておき、実行します。要素の位置  
は1から数えます。

では、行列  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  から、(1,2) 成分を取り出しましょう。

(0) あらかじめ行列がある状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Matrix	:	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

(1) 抽出する要素の位置を指定

`1 2`

#	TYPE	:	VALUE
4:		:	
Z:	Matrix	:	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Y:	Integer	:	1
X:	Integer	:	2

(2) 単一要素の抽出

`mget`

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	2

### 10.3.2 行列からのベクトルの抽出

行列から行ベクトル（横方向）を抽出するには、キーワード「`MGETR`」を、列ベクトル  
（縦方向）を抽出するには、キーワード「`MGETC`」を用います。

Yに行列を、Xにベクトルの位置（整数）をプッシュしておき、実行します。要素の位置  
は1から数えます。

では、行列  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  から、2 番目の列ベクトルを取り出しましょう。

(0) あらかじめ行列がある状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Matrix	:	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$



(1) 抽出する要素の位置を指定

`Cmd. 2`

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Matrix	:	[[1, 2], [3, 4]]
X:	Integer	:	2

(2) 列ベクトルの抽出

`Cmd. mgetc`

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Tuple(Col)	:	(2, 4)

## 10.4 行列を含む四則演算

行列同士の加減算や、行列とベクトル、行列と数との乗除算を行う場合でも、数同士の四則演算と同じキーワードを使います。

計算例 1  $\begin{bmatrix} 3 & 7 \\ 9 & 5 \end{bmatrix} - \begin{bmatrix} 2 & 6 \\ 2 & 4 \end{bmatrix}$

```
Cmd. 3 7 2 mrtup 9 5 2 mrtup 2 mkmat
Cmd. 2 6 2 mrtup 2 4 2 mrtup 2 mkmat
Cmd. -
```

計算例 2  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix}$

```
Cmd. 1 2 2 mrtup 3 4 2 mrtup 2 mkmat
Cmd. 5 6 2 mctup
Cmd. *
```

## 10.5 行列式と逆行列

行列式を求めるには、キーワード「`Cmd. DET`」を、逆行列を求めるには、逆数のキーワード「`Cmd. INV`」を用います。

機能	キーワード	R	D	演算内容
行列式	DET	1	1	$\det X$
逆行列	INV	1	1	$X^{-1}$

行列式・逆行列ともに、正方行列に限ります。行列式の値がゼロになる正方行列の逆行列はエラーとなります。

計算例  $\begin{bmatrix} \sqrt{2} & 1 \\ 1 & \sqrt{2} \end{bmatrix}^{-1}$

```
Cmd. 2 sqrt 1 2 mrtup 1 2 sqrt 2 mrtup 2 mkmat inv
```

## 10.6 行列の転置

行列の転置を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
転置	TRANS	1	1	$X^T$
エルミート転置	HTRANS, HCONJ	1	1	$(X^T)^*$

エルミート転置は、通常の転置を行った後、すべての成分について共軛を取ります。

## 10.7 その他の行列演算

その他の行列演算に、次のものがあります。

機能	キーワード	R	D	演算内容
トレース	TRACE	1	1	$\text{tr}(X)$

トレースは正方行列に対してのみ有効です。

# Chapter 11

## レジスタ機能

### 11.1 レジスタとは

レジスタは、スタックとは別に値を格納できる領域です。レジスタをうまく使うと、効率的に計算を進められます。レジスタ機能では、RA～RZ の 26 本のレジスタが利用できます。

HOMURA: (FD) (Rad) (Hex) (Dword) [Reg]		
Std: 9/15, Stack: 3, History: 0/2		
=====		
#	TYPE :	VALUE
RA:	Floating :	3.14159265
RB:	:	
RC:	:	
<hr/>		
Z:	Integer :	4
Y:	Integer :	3
X:	Integer :	2
-----		
STORE to selected register		
Ready to operate		

各レジスタには、1 つのデータがストア（登録）できます。一度レジスタにストアしたデータは、何度もロード（呼び出し）できます。また、レジスタの上書きやデリート（削除）も自由に行えます。レジスタはスタックから独立しており、スタックが変更されても、レジスタは影響を受けません。

また、レジスタにはいかなるデータも登録できます。整数や有理数はもちろん、ベクトルや行列、エラーや文字列さえも、1 つのレジスタに 1 つ登録できます。

なお、本章では、レジスタの機能の説明のため、次のような表を用います。

#	TYPE	:	VALUE
RA:		:	
RB:		:	
Z:		:	
Y:		:	
X:		:	

この表では、スタックの X, Y, Z と、レジスタの RA, RB を表示しています。

## 11.2 レジスタの表示切り替え

レジスタの表示・非表示を切り替えるには、キーワード「Cmd REGISTER」または「Cmd REG」を単体で入力します。レジスタ表示が有効のとき、計算設定・状態表示領域には [Reg] が、スタック表示領域の上部分にはレジスタが表示されます。確認したいレジスタが見えない場合は、レジスタのページめくり機能を用います。レジスタの表示に関するキーワードは次のとおりです。

機能	キーワード
レジスタの表示・非表示	REGISTER, REG
レジスタの次のページ	REGNEXT, RN
レジスタの前のページ	REGPREV, RP
レジスタの最初のページ	REGFIRST, RF

ページめくりに関しては、3 章「設定・表示切り替え」の 4 節「レジスタのページめくり」をお読みください。

レジスタが画面に表示されていなくても、レジスタ機能は利用できます。また、レジスタの表示を変更しても、レジスタの内容に影響しません。そのため、レジスタの表示・非表示の切り替えや、ページめくりによって、レジスタの内容がクリアされることはありません。

## 11.3 指定レジスタへのストア

指定レジスタへのストア機能は、X のデータを指定したレジスタに移動する機能です。指定レジスタへのストアは、次のキーワードを用います。

機能	キーワード	R	D
RA へのストア	STRA	1	1
RB へのストア	STRB	1	1
⋮	⋮		
RZ へのストア	STRZ	1	1

「Cmd STR?

」という形式で、この「?」にはレジスタを指定するアルファベットが入ります。では、RA に整数の 5 をストアしてみましょう。

(1) ストアしたい数をプッシュ

Cmd 5

#	TYPE	:	VALUE
RA:		:	
RB:		:	
Z:		:	
Y:		:	
X:	Integer	:	5

(2) RA にストア

Cmd. stra

#	TYPE	:	VALUE
RA:	Integer	:	5
RB:		:	
Z:		:	
Y:		:	
X:		:	

また、同じレジスタに対してストアを行うと、レジスタの内容は上書きされます。

(1) 最初の状態

#	TYPE	:	VALUE
RA:	Integer	:	5
RB:		:	
Z:		:	
Y:		:	
X:	Integer	:	7

(2) ストアしたい数をプッシュ

Cmd. 9

#	TYPE	:	VALUE
RA:	Integer	:	5
RB:		:	
Z:		:	
Y:	Integer	:	7
X:	Integer	:	9

(3) RA の上書き

Cmd. stra

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:		:	
Z:		:	
Y:		:	
X:	Integer	:	7

RB～RZ についても同様です。

## 11.4 指定レジスタのロード

指定レジスタのロード機能は、データがストアされているレジスタから、その内容をスタックにプッシュする機能です。ただし、レジスタにデータがない場合はエラーです。

なお、ロードを行ってもレジスタの中にデータは残るので、何度も呼び出しが可能です。指定レジスタのロードは、次のキーワードを用います。

機能	キーワード	R	D
RA のロード	LDRA	0	0
RB のロード	LDRB	0	0
⋮	⋮		
RZ のロード	LDRZ	0	0

「Cmn LDR?

」という形式で、この「?」にはレジスタを指定するアルファベットが入ります。では、RA と RB をロードして、加算してみましょう。

### (1) 最初の状態

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y:		:	
X:		:	

### (2) RA をロード

Cmn ldra

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y:		:	
X: Integer		:	9

### (3) RB をロード

Cmn ldrb

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y: Integer		:	9
X: Integer		:	4

(4) 加算

**Cmd.** +

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y:		:	
X:	Integer	:	13

RC～RZ についても同様です。

# 11.5 指定レジスタのデリート

指定レジスタの内容を明示的にデリートすることもできます。ただし、実行できるのは、指定したレジスタにすでにデータが登録されている場合に限りです。指定レジスタのデリートには、次のキーワードを用います。

機能	キーワード	R	D
RA のデリート	DELRA	0	0
RB のデリート	DELRB	0	0
⋮	⋮		
RZ のデリート	DELRZ	0	0

「**Cmd.** DELR?

(1) 最初の状態

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y:		:	
X:		:	

(2) RA をデリート

**Cmd.** delra

#	TYPE	:	VALUE
RA:		:	
RB:	Integer	:	4
Z:		:	
Y:		:	
X:		:	

## 11.6 レジスタ演算

レジスタ演算とは、指定レジスタと  $X$  とで計算を行い、その計算結果でそのレジスタを上書きする機能です。

レジスタ演算を行うには、次のキーワードを用います。下表のキーワード列の「?」には、レジスタに対応するアルファベットが入ります。

機能	キーワード	R	D	演算内容
レジスタインクリメント	IR?, ++R?	0	0	$R \leftarrow R + 1$
レジスタデクリメント	DR?, --R?	0	0	$R \leftarrow R - 1$
レジスタ加算	ADDR?, +R?	1	1	$R \leftarrow R + X$
レジスタ減算	SUBR?, -R?	1	1	$R \leftarrow R - X$
レジスタ乗算	MULR?, *R?	1	1	$R \leftarrow R \times X$
レジスタ除算	DIVR?, /R?	1	1	$R \leftarrow R/X$

レジスタ演算を実行すると、スタックから 1 つのデータがドロップされ、計算結果がレジスタに上書きされます。レジスタインクリメントとレジスタ加算の例を示します。

### (1) 最初の状態

#	TYPE	:	VALUE
RA:	Integer	:	9
RB:	Integer	:	4
Z:		:	
Y:		:	
X:		:	

### (2) RA をインクリメント

**Cmd.** ira

#	TYPE	:	VALUE
RA:	Integer	:	10
RB:	Integer	:	4
Z:		:	
Y:		:	
X:		:	

### (3) 小数 1.2 をプッシュ

**Cmd.** 1.2



#	TYPE	:	VALUE
RA:	Integer	:	10
RB:	Integer	:	4
Z:		:	
Y:		:	
X:	Floating	:	1.2

## (4) RB と加算

**Cmd** +rb

#	TYPE	:	VALUE
RA:	Integer	:	10
RB:	Floating	:	5.2
Z:		:	
Y:		:	
X:		:	

## 11.7 レジスタクリア

すべてのレジスタを一掃するには、レジスタクリアを実行します。レジスタクリアには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
レジスタクリア	REGCLEAR, RCLR	0	0	

指定レジスタのみを削除したい場合は、指定レジスタのデリートを実行します。

レジスタとスタックの両方をクリアするには、オールクリアが便利です。キーワード「**Cmd** AC」でオールクリアを実行できます。

## 11.8 文字列とレジスタ

レジスタにはいかなるデータも登録できます。文字列も例外ではありません。

本ソフトウェアでは、文字列を半角ダブルクォーテーションで囲うと、文字列データをスタックにプッシュできます。これをレジスタに登録することにより、レジスタ内に目印をつけることができます。また、これをマクロ機能と組み合わせると、より強力です。

詳しくは、2章「画面の見方と操作方法」の5節「数の入力方法」や、15章「その他の機能」の5節「マクロ機能」をお読みください。



## Chapter 12

# スタック操作

### 12.1 スタックの特殊操作

スタックにデータをプッシュする順番を誤った場合や、末尾以外のデータを変更したい場合、入力し直すのは面倒です。本ソフトウェアでは、スタック操作機能を多く用意してあります。本章では、スタック操作をすべて説明します。

### 12.2 基本的なスタック操作

基本的なスタック操作を行うには、次のキーワードを用います。

機能	キーワード	R	D	演算内容
ドロップ	DROP, \	1	1	
コピー※	DUP	1	1	
クリア	CLEAR, CLR	N>0	N	

※入力欄が空の状態 Enter キーを押下しても実行されます

### 12.3 スタックの順番を変更する操作

スタックの順番の操作を行う機能には、次のものがあります。

機能	キーワード	R	D	演算内容
スワップ	SWAP, \$	2	0	
末尾 3 要素回転	ROT	3	0	
末尾 3 要素逆回転	UNROT	3	0	
位置指定回転	ROLL	N	1	
位置指定逆回転	ROLLD	N	1	

それぞれの詳細は次のとおりです。

### 12.3.1 スワップ

スタックの  $Y$  と  $X$  を交換する操作です。スタックに 2 つ以上のデータが必要です。キーワードは「`Cmd` SWAP」 「`Cmd` \$」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Rational	:	9/4

(2) スワップ

`Cmd` swap

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Rational	:	9/4
X:	Integer	:	3

### 12.3.2 末尾 3 要素回転

スタックの  $Z, Y, X$  を回転する操作です。スタックに 3 つ以上のデータが必要です。並びは次のようになります。

$$\begin{pmatrix} Z \\ Y \\ X \end{pmatrix} \rightarrow \begin{pmatrix} Y \\ X \\ Z \end{pmatrix}$$

キーワードは「`Cmd` ROT」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Rational	:	9/4

(2) 末尾 3 要素回転

`Cmd` rot

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Rational	:	9/4
X:	Integer	:	3

### 12.3.3 末尾 3 要素逆回転

スタックの  $Z, Y, X$  を逆回転する操作です。スタックに 3 つ以上のデータが必要です。並びは次のように変わります。

$$\begin{pmatrix} Z \\ Y \\ X \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Z \\ Y \end{pmatrix}$$

キーワードは「**Cmd** UNROT」です。

### 12.3.4 位置指定回転

指定位置からスタックのデータの並びを回転させる操作です。指定位置のデータが末尾に移動します。位置を指定する正の整数をプッシュし、回転させます。

キーワードは「**Cmd** ROLL」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Rational	:	9/4

(2) 位置指定 (3 番目、つまり  $Z$  とする)

**Cmd** 3

#	TYPE	:	VALUE
4:	Integer	:	256
Z:	Rational	:	9/4
Y:	Integer	:	3
X:	Integer	:	3

(3) 位置指定回転

**Cmd** roll

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	3
Y:	Rational	:	9/4
X:	Integer	:	256

### 12.3.5 位置指定逆回転

指定位置からスタックのデータの並びを逆回転させる操作です。末尾のデータが指定位置に移動します。位置を指定する正の整数をプッシュし、回転させます。

キーワードは「**Cmd** ROLLD」です。

操作やスタックの動きに関しては、位置指定回転と同様です。

## 12.4 コピー・上書きをする操作

コピー・上書きをする操作には、次のものがあります。

機能	キーワード	R	D
オーバー	OVER, 0	2	0
位置指定コピー	PICK	N	0
位置指定上書き	UNPICK	N	1
末尾 2 要素コピー	XY, YX, DUP2	2	0
2 回コピー	DUPDUP	1	0
個数指定コピーおよび個数	NDUPN	1	1

それぞれの詳細は次のとおりです。

### 12.4.1 オーバー

スタックの Y をコピーし、プッシュします。データ数が 2 未満ならばエラーです。

キーワードは「`Cmd OVER`」「`Cmd 0`」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	16
X:	Integer	:	32

(2) オーバー

`Cmd 0`

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	16
Y:	Integer	:	32
X:	Integer	:	16

### 12.4.2 位置指定コピー

指定された位置のデータをコピーし、プッシュします。

キーワードは「`Cmd PICK`」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Rational	:	9/4

(2) 位置指定 (3 番目、つまり Z とする)

**Cmd.** 3

#	TYPE	:	VALUE
4:	Integer	:	256
Z:	Integer	:	3
Y:	Rational	:	9/4
X:	Integer	:	3

(3) 位置指定コピー

**Cmd.** pick

#	TYPE	:	VALUE
4:	Integer	:	256
Z:	Integer	:	3
Y:	Rational	:	9/4
X:	Integer	:	256

### 12.4.3 位置指定上書き

Y で指定されたデータで、X で指定した位置のデータを上書きします。

キーワードは「**Cmd.** UNPICK」です。

(1) 最初の状態 (2 番目、つまり Y を上書きしたい)

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	256
X:	Integer	:	3

(2) データをプッシュ

**Cmd.** 64

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Integer	:	64

(3) 位置をプッシュ

**Cmd.** 2

#	TYPE	:	VALUE
4:	Integer	:	256
Z:	Integer	:	3
Y:	Integer	:	64
X:	Integer	:	2

(4) 位置指定上書き

`Cmd` unpick

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	64
X:	Integer	:	3

#### 12.4.4 末尾 2 要素コピー

スタックの Y と X にあるデータをそれぞれコピーし、Y, X の順番にプッシュします。データ数が 2 未満ならばエラーです。

キーワードは「`Cmd` XY」「`Cmd` YX」「`Cmd` DUP2」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	16
X:	Integer	:	32

(2) 末尾 2 要素コピー

`Cmd` xy

#	TYPE	:	VALUE
4:	Integer	:	16
Z:	Integer	:	32
Y:	Integer	:	16
X:	Integer	:	32

#### 12.4.5 2 回コピー

コピー操作を単純に 2 回実行します。キーワードは「`Cmd` DUPDUP」「`Cmd` DD」です。

#### 12.4.6 個数指定コピーおよび個数

X が正の整数のとき、一旦 X をドロップし、Y のデータが X 個になるようにコピーし、X をプッシュします。キーワードは「`Cmd` NDUPN」です。

(1) 最初の状態



#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	16
X:	Integer	:	32

(2) 個数を指定

**Cmd.** 2

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	16
Y:	Integer	:	32
X:	Integer	:	2

(3) 個数指定コピーおよび個数

**Cmd.** ndupn

#	TYPE	:	VALUE
4:	Integer	:	16
Z:	Integer	:	32
Y:	Integer	:	32
X:	Integer	:	2

## 12.5 削除をする操作

削除をする操作には、次のものがあります。

機能	キーワード	R	D
2 回ドロップ	DROP2, \\\	2	2
3 回ドロップ	DROP3, \\\	3	3
N 回ドロップ	DROPN	N+1	N+1
2 番目の削除	NIP	2	2
指定位置の削除	NIPN	N	2

それぞれの詳細は次のとおりです。

### 12.5.1 2 回ドロップ

ドロップ操作を単純に 2 回実行します。キーワードは「**Cmd.** DROP2」「**Cmd.** \\\」です。

### 12.5.2 3 回ドロップ

ドロップ操作を単純に 3 回実行します。キーワードは「**Cmd.** DROP3」「**Cmd.** \\\」です。

## 12.5.3 N 回ドロップ

整数で指定した数のデータ、およびその数をドロップします。キーワードは「`Cmd DROPN`」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	256
Y:	Integer	:	3
X:	Rational	:	9/4

(2) 整数（ドロップする個数）をプッシュ

`Cmd 2`

#	TYPE	:	VALUE
4:	Integer	:	256
Z:	Integer	:	3
Y:	Rational	:	9/4
X:	Integer	:	2

(3) N 回ドロップ（今回の場合、2 + 1 回ドロップ）

`Cmd dropn`

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	256

## 12.5.4 2 番目の削除

スタックの Y を削除します。キーワードは「`Cmd NIP`」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	16
X:	Integer	:	32

(2) 2 番目の削除

`Cmd nip`

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	32

### 12.5.5 指定位置の削除

整数で指定した位置のデータを削除します。

キーワードは「Cmd NIPN」です。

(1) 最初の状態

#	TYPE	:	VALUE
4:		:	
Z:	Integer	:	64
Y:	Integer	:	16
X:	Integer	:	32

(2) 位置を指定

Cmd 3

#	TYPE	:	VALUE
4:	Integer	:	64
Z:	Integer	:	16
Y:	Integer	:	32
X:	Integer	:	3

(3) 位置指定削除

Cmd nipn

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:	Integer	:	16
X:	Integer	:	32

## 12.6 その他のスタック操作

その他のスタック操作には、次のものがあります。

機能	キーワード	R	D
スタック深さ	DEPTH	0	0

スタック深さは、現在のスタックのデータ数を整数でプッシュします。



## Chapter 13

# 単位換算機能

6 章「便利な計算機能」にも単位換算の機能があるので、併せてお読みください。

### 13.1 対応する単位の種類

本ソフトウェアの単位換算機能で対応する単位の種類は次のとおりです。

- 長さ
- 長さの逆数
- 面積
- 面積の逆数
- 体積
- 体積の逆数
- 時間
- 時間の逆数
- 質量
- 速度
- 加速度
- 力
- 圧力
- エネルギー
- 温度

単位換算に対応するデータ型はスカラー（整数、浮動小数点数、有理数）のみです。

### 13.2 単位換算機能の使い方

単位換算機能を使うには、キーワード「Cmd CONV」 「Cmd CV」を用います。また、前回と同じ単位換算を行う場合、キーワード「Cmd REC」 「Cmd Z」で単位の入力を省略できます。ただし、ソフトウェアを起動して一度でも正しい組み合わせで単位換算を実行した場合に限ります。

機能	キーワード	R	D	演算内容
単位換算	CONV, CV	1	1	Unit conversion
前回と同じ単位換算	REC, Z	1	1	Unit conversion (Redo)

スタックの末尾のデータに対して単位換算を実行できます。変換元の単位と変換先の単位をキーワードで指定して実行します。たとえば、インチからセンチメートルに換算するには次のように入力します。

Cmd conv in cm

指定された2つの単位について、単位の種類が異なる場合、エラーとなります。

### 13.3 長さの単位

単位換算機能で対応する長さの単位とキーワードを示します。

単位		キーワード	基準との比
メートル	[m]	M	1
キロメートル	[km]	KM	1 E+03
センチメートル	[cm]	CM	1 E-02
ミリメートル	[mm]	MM	1 E-03
海里 ※ 1	[nmi]	NMI	1 852
ヤード ※ 1	[yd]	YD	0.914 4
フィート ※ 1	[ft]	FT	0.304 8
インチ ※ 1	[in]	IN	0.025 4
マイル ※ 1	[mi]	MI	1 609.344
ファゾム ※ 2	[fath]	FATH	1.828 8
パイカ ※ 3	[pc]	PC	127/30000
ポイント ※ 3	[pt]	PT	127/360000
尺		SHAKU	10/33
寸		SUN	1/33
間		KEN	20/11
丈		JOU	100/33
町		CHOU	1200/11
里		RI	43200/11

※ 1 国際単位です

※ 2 英ファゾムです

※ 3 PostScript の単位です

### 13.4 長さの逆数の単位

単位換算機能で対応する長さの逆数の単位とキーワードを示します。

単位		キーワード	基準との比
毎メートル	[1/m]	/M	1
毎キロメートル	[1/km]	/KM	1 E-03
毎センチメートル	[1/cm]	/CM	1 E+02
毎ミリメートル	[1/mm]	/MM	1 E+03
毎海里	[1/nmi]	/NMI	1/1852
毎ヤード	[1/yd]	/YD	1250/1143
毎フィート	[1/ft]	/FT	1250/381

次のページに続く

前ページからの続き

単位		キーワード	基準との比
毎インチ	[1/in]	/IN	5000/127
毎マイル	[1/mi]	/MI	125/201168
毎ファゾム	[1/fath]	/FATH	625/1143
毎パイカ	[1/pc]	/PC	30000/127
毎ポイント	[1/pt]	/PT	360000/127
毎尺		/SHAKU	3.3
毎寸		/SUN	33
毎間		/KEN	0.55
毎丈		/JOU	0.33
毎町		/CHOU	11/1200
毎里		/RI	11/43200

## 13.5 面積の単位

単位換算機能で対応する面積の単位とキーワードを示します。

単位		キーワード	基準との比
平方メートル	[m <sup>2</sup> ]	M2	1
平方キロメートル	[km <sup>2</sup> ]	KM2	1 E+06
平方センチメートル	[cm <sup>2</sup> ]	CM2	1 E-04
平方ミリメートル	[mm <sup>2</sup> ]	MM2	1 E-06
アール	[a]	ARE	1 E+02
ヘクタール	[ha]	HA	1 E+04
エーカー	[ac]	ACRE	4 046.856 422 4
平方ヤード	[yd <sup>2</sup> ]	YD2	0.836 127 36
平方フィート	[ft <sup>2</sup> ]	FT2	9.290 304 E-02
平方インチ	[in <sup>2</sup> ]	IN2	6.451 6 E-04
平方マイル	[mi <sup>2</sup> ]	MI2	2 589 988.110 336
坪		TSUBO	400/121
畝		ISSE	12000/121
反		ITTAN	120000/121
町歩		CHOUBU	1200000/121

## 13.6 面積の逆数の単位

単位換算機能で対応する面積の逆数の単位とキーワードを示します。

単位		キーワード	基準との比
毎平方メートル	[1/m <sup>2</sup> ]	/M2	1
毎平方キロメートル	[1/km <sup>2</sup> ]	/KM2	1 E-06
毎平方センチメートル	[1/cm <sup>2</sup> ]	/CM2	1 E+04
毎平方ミリメートル	[1/mm <sup>2</sup> ]	/MM2	1 E+06
毎アール	[1/a]	/ARE	1 E-02
毎ヘクタール	[1/ha]	/HA	1 E-04
毎エーカー	[1/ac]	/ACRE	78125/316160658
毎平方ヤード	[1/yd <sup>2</sup> ]	/YD2	1562500/1306449
毎平方フィート	[1/ft <sup>2</sup> ]	/FT2	1562500/145161
毎平方インチ	[1/in <sup>2</sup> ]	/IN2	25000000/16129
毎平方マイル	[1/mi <sup>2</sup> ]	/MI2	15625/40468564224
毎坪		/TSUBO	121/400
毎畝		/ISSE	121/12000
毎反		/ITTAN	121/120000
毎町歩		/CHOUBU	121/1200000

## 13.7 体積の単位

単位換算機能で対応する体積の単位とキーワードを示します。

単位		キーワード	基準との比
立方メートル	[m <sup>3</sup> ]	M3	1
立方キロメートル	[km <sup>3</sup> ]	KM3	1 E+09
立方センチメートル	[cm <sup>3</sup> ]	CM3	1 E-06
立方ミリメートル	[mm <sup>3</sup> ]	MM3	1 E-09
リットル	[L]	L	1 E-03
デシリットル	[dL]	DL	1 E-04
キロリットル	[kL]	KL	1
ミリリットル	[mL]	ML	1 E-06
立方ヤード	[yd <sup>3</sup> ]	YD3	0.764 554 857 984
立方フィート	[ft <sup>3</sup> ]	FT3	0.028 316 846 592
立方インチ	[in <sup>3</sup> ]	IN3	1.638 706 4 E-05
立方マイル	[mi <sup>3</sup> ]	MI3	4 168 181 825.440 579 584
英ガロン	[gal <sub>imp</sub> ]	IMG	4.546 09 E-03
米ガロン	[gal <sub>us</sub> ]	USG	3.785 411 784 E-03
合		GOU	2401/13310000
升		SHOU	2401/1331000
斗		ITTO	2401/133100
石		KOKU	2401/13310



## 13.8 体積の逆数の単位

単位換算機能で対応する体積の逆数の単位とキーワードを示します。

単位		キーワード	基準との比
毎立方メートル	[1/m <sup>3</sup> ]	/M3	1
毎立方キロメートル	[1/km <sup>3</sup> ]	/KM3	1 E-09
毎立方センチメートル	[1/cm <sup>3</sup> ]	/CM3	1 E+06
毎立方ミリメートル	[1/mm <sup>3</sup> ]	/MM3	1 E+09
毎リットル	[1/L]	/L	1 E+03
毎デシリットル	[1/dL]	/DL	1 E+04
毎キロリットル	[1/kL]	/KL	1
毎ミリリットル	[1/mL]	/ML	1 E+06
毎立方ヤード	[1/yd <sup>3</sup> ]	/YD3	1953125000/1493271207
毎立方フィート	[1/ft <sup>3</sup> ]	/FT3	1953125000/55306341
毎立方インチ	[1/in <sup>3</sup> ]	/IN3	1250000000000/2048383
毎立方マイル	[1/mi <sup>3</sup> ]	/MI3	1953125/8140980127813632
毎英ガロン	[1/gal <sub>imp</sub> ]	/IMG	1000000000/454609
毎米ガロン	[1/gal <sub>us</sub> ]	/USG	1250000000000/473176473
毎合		/GOU	13310000/2401
毎升		/SHOU	1331000/2401
毎斗		/ITT0	133100/2401
毎石		/KOKU	13310/2401

## 13.9 時間の単位

単位換算機能で対応する時間の単位とキーワードを示します。

単位		キーワード	基準との比
秒	[s]	SEC, S	1
分	[min]	MIN	60
時	[h]	HOUR, H	3 600
日	[d]	DAY, C	86 400
週	[wk]	WEEK, WK	604 800
平年	[yr]	YEAR, YR	31 536 000
グレゴリオ平均年		GYEAR	31 556 952
ユリウス平均年		JYEAR	31 557 600

## 13.10 時間の逆数の単位

単位換算機能で対応する時間の逆数の単位とキーワードを示します。

単位		キーワード	基準との比
毎秒	[1/s]	/SEC, /S	1
毎分	[1/min]	/MIN	1/60
毎時	[1/h]	/HOUR, /H	1/3600
毎日	[1/d]	/DAY, /D	1/86400
毎週	[1/wk]	/WEEK, /WK	1/604800
毎平年	[1/yr]	/YEAR, /YR	1/31536000
毎グレゴリオ平均年		/GYEAR	1/31556952
毎ユリウス平均年		/JYEAR	1/31557600

## 13.11 質量の単位

単位換算機能で対応する質量の単位とキーワードを示します。

単位		キーワード	基準との比
キログラム	[kg]	KG	1
グラム	[g]	G	1 E-03
ミリグラム	[mg]	MG	1 E-06
メトリックトン	[t]	TON	1 E+03
ロングトン (英トン)	[lt]	LTON	1 016.046 908 8
ショートトン (米トン)	[st]	STON	907.184 74
常用オンス	[ozav]	OZ	0.028 349 523 125
常用ポンド	[lbav]	LB	0.453 592 37
貫		KAN	3.75
両		RYOU	3.75 E-02
匁		MOMME	3.75 E-03
斤		KIN	0.6

## 13.12 速度の単位

単位換算機能で対応する速度の単位とキーワードを示します。

単位		キーワード	基準との比
メートル毎秒	[m/s]	M/S	1
メートル毎分	[m/min]	M/MIN	1/60
キロメートル毎秒	[km/s]	KM/S	1 E+03
キロメートル毎時	[km/h]	KM/H, KPH	5/18
インチ毎秒	[ips]	IPS	0.025 4
フィート毎秒	[fps]	FPS	0.304 8

次のページに続く

前ページからの続き

単位		キーワード	基準との比
マイル毎時	[mph]	MPH	0.447 04
ノット（海里毎時）	[kn]	KN, KT	463/900

### 13.13 加速度の単位

単位換算機能で対応する加速度の単位とキーワードを示します。

単位		キーワード	基準との比
メートル毎秒毎秒	[m/s <sup>2</sup> ]	M/S2	1
キロメートル毎時毎秒	[km/h/s]	KM/H/S, KPH/S	5/18
ガル	[Gal]	GAL	1 E-02
インチ毎秒毎秒	[ips <sup>2</sup> ]	IPS2	0.025 4
フィート毎秒毎秒	[fps <sup>2</sup> ]	FPS2	0.304 8
マイル毎時毎秒	[mph/s]	MPH/S	0.447 04
ノット毎秒	[kn/s]	KN/S	463/900

### 13.14 力の単位

単位換算機能で対応する力の単位とキーワードを示します。

単位		キーワード	基準との比
ニュートン	[N]	NEWTON	1
ダイン	[dyn]	DYN	1 E-05
重量キログラム	[kgf]	KGF	9.806 65
重量グラム	[gf]	GF	9.806 65 E-03

### 13.15 圧力の単位

単位換算機能で対応する圧力の単位とキーワードを示します。

単位		キーワード	基準との比
パスカル	[Pa]	PA	1
ヘクトパスカル	[hPa]	HPA	1 E+02
キロパスカル	[kPa]	KPA	1 E+03
メガパスカル	[MPa]	MPA	1 E+06
バール	[bar]	BAR	1 E+05
水銀柱ミリメートル	[mmHg]	MMHG	101325/760

次のページに続く

前ページからの続き

単位		キーワード	基準との比
水銀柱インチ	[inHg]	INHG	3 386.388 64

## 13.16 エネルギーの単位

単位換算機能で対応するエネルギーの単位とキーワードを示します。

単位		キーワード	基準との比
ジュール	[J]	J	1
キロジュール	[kJ]	KJ	1 E+03
メガジュール	[MJ]	MJ	1 E+06
電子ボルト	[eV]	EV	1.602 176 620 8 E-19
キロ電子ボルト	[keV]	KEV	1.602 176 620 8 E-16
メガ電子ボルト	[MeV]	MEV	1.602 176 620 8 E-13
ギガ電子ボルト	[GeV]	GEV	1.602 176 620 8 E-10
熱力学カロリー	[cal <sub>th</sub> ]	CAL	4.184
キロカロリー	[kcal <sub>th</sub> ]	KCAL	4 184
TNT 換算トン	[t <sub>TNT</sub> ]	TTNT	4.184 E+09
キロワット時	[kWh]	KWH	3.6 E+06
英熱量 (BTU)	[Btu]	BTU	1055.06

## 13.17 温度の単位

単位換算機能で対応する温度の単位とキーワードを示します。

単位		キーワード	刻み幅比	絶対零度
ケルビン	[K]	KEL	1	0
セルシウス度	[°C]	DEGC	1	-273.15
ランキン度	[si°R]	DEGR	5/9	0
ファーレンハイト度	[si°F]	DEGF	5/9	-459.67

セルシウス度とファーレンハイト度は絶対零度の値が異なります。それぞれ、絶対零度を基準にしたケルビンやランキン度に変換してから換算します。

たとえば、セルシウス度からファーレンハイト度への変換は、次のようになります。

$$\theta[^\circ\text{C}] = (\theta + 273.15) \times \frac{9}{5} - 459.67[^\circ\text{F}]$$

## Chapter 14

# ソフトウェア内蔵定数

### 14.1 定数の入力

本ソフトウェアは、数学・科学定数を内蔵しています。定数の値をプッシュするには、対応するキーワードを入力します。

科学定数表は 2014 CODATA を元に作成しています。

### 14.2 数学定数

基本的な数学定数のキーワードは次のとおりです。

定数名	キーワード	値
円周率	PI	3.141 592 653 589 79
ネイピア数	E	2.718 281 828 459 05
オイラーの定数	EG	0.577 215 664 901 533

### 14.3 基本的な物理定数

基本的な物理定数のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
真空中の光速	$c_0$ [m/s]	LIGHT, C0	299 792 458
真空透磁率	$\mu_0$ [H/m]	MAGNETIC, MU0	1.256 637 061 436 E-06
真空誘電率	$\epsilon_0$ [F/m]	ELECTRIC, EPS0	8.854 187 817 620 E-12
真空特性インピーダンス	$Z_0$ [ $\Omega$ ]	IMPEDANCE, Z0	376.730 313 461
万有引力定数	$G_0$ [ $\text{m}^3/\text{kg}/\text{s}^2$ ]	GRAVITATION, NEWTONIAN	6.674 08 E-11
プランク定数	$h$ [J s]	PLANCK, H	6.626 070 040 E-34
換算プランク定数	[J s]	RPLANCK, HBAR	1.054 571 800 E-34

## 14.4 電磁気学

電磁気学の定数のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
電気素量	$e$ [C]	ECHARGE, EVOLT	1.602 176 620 8 E-19
磁束量子	$\Phi_0$ [Wb]	Q.FLUX	2.067 833 831 E-15
コンダクタンス量子	$G_0$ [S]	Q.CONDUCT	7.748 091 731 0 E-05
抵抗量子	$R_0$ [ $\Omega$ ]	Q.RESIST	12 906.403 727 8
ジョセフソン定数	$K_J$ [Hz/V]	JOSEPHSON	483 597.852 5 E-09
フォン・クリッツィング定数	$R_K$ [ $\Omega$ ]	KLITZING	25 812.807 455 5
ボーア磁子	$\mu_B$ [J/T]	B.MAGNETON	927.400 999 4 E-26
核磁子	$\mu_N$ [J/T]	N.MAGNETON	5.050 783 699 E-27

## 14.5 原子・核物理学

原子・核物理学の定数のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
微細構造定数	$\alpha$ [1]	FSTRUCT	7.297 352 566 4 E-03
リュードベリ定数	$R_\infty$ [ $\text{m}^{-1}$ ]	RYDBERG	10 973 731.568 508
ボーア半径	$a_0$ [m]	B.RADIUS	0.529 177 210 67 E-10
ハートリーエネルギー	$E_h$ [J]	HARTREE	4.359 744 650 E-18

電子に関する定数です。

定数名	記号と単位	キーワード	値
電子の質量	$m_e$ [kg]	E.MASS	9.109 383 56 E-31
電子のコンプトン波長	$\lambda_e$ [m]	E.COMPTON	2.426 310 236 7 E-12
古典電子半径	$r_e$ [m]	E.RADIUS	2.817 940 322 7 E-15
電子の磁気モーメント	$\mu_e$ [J/T]	E.MAGNETIC	-928.476 462 0 E-26
電子の磁気回転比	$\gamma_e$ [ $\text{s}^{-1}\text{T}^{-1}$ ]	E.GYRO	1.760 859 644 E+11

陽子に関する定数です。

定数名	記号と単位	キーワード	値
陽子の質量	$m_p$ [kg]	P.MASS	1.672 621 898 E-27
陽子のコンプトン波長	$\lambda_p$ [m]	P.COMPTON	1.321 409 853 96 E-15
陽子の磁気モーメント	$\mu_p$ [J/T]	P.MAGNETIC	1.410 606 787 3 E-26
陽子の磁気回転比	$\gamma_p$ [ $\text{s}^{-1}\text{T}^{-1}$ ]	P.GYRO	2.675 221 900 E+08

中性子に関する定数です。

定数名	記号と単位	キーワード	値
中性子の質量	$m_n$ [kg]	N.MASS	1.674 927 471 E-27
中性子のコンプトン波長	$\lambda_n$ [m]	N.COMPTON	1.319 590 904 81 E-15
中性子の磁気モーメント	$\mu_n$ [J/T]	N.MAGNETIC	-0.966 236 50 E-26
中性子の磁気回転比	$\gamma_n$ [s <sup>-1</sup> T <sup>-1</sup> ]	N.GYRO	1.832 471 72 E+08

その他素粒子に関する定数です。

定数名	記号と単位	キーワード	値
ミューオンの質量	$m_\mu$ [kg]	MU.MASS	1.883 531 594 E-28
ミューオンの磁気モーメント	$\mu_\mu$ [J/T]	MU.MAGNETIC	-4.490 448 26 E-26
タウオンの質量	$m_\tau$ [kg]	TAU.MASS	3.167 47 E-27

## 14.6 物理化学

物理化学の定数のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
ボルツマン定数	$k$ [J/K]	BOLTZMANN	1.380 648 52 E-23
アボガドロ定数	$N_A$ [mol <sup>-1</sup> ]	AVOGADRO	6.022 140 857 E+23
原子質量定数	$m_u$ [kg]	DALTON	1.660 539 040 E-27
ファラデー定数	$F$ [C/mol]	FARADAY	96 485.332 89
モル気体定数	$R$ [J K <sup>-1</sup> mol <sup>-1</sup> ]	GAS	8.314 4598
理想気体のモル体積※ (m <sup>3</sup> )	$V_m$ [m <sup>3</sup> /mol]	MOLV	22.413 962 E-03
理想気体のモル体積※ (L)	$V_m$ [L/mol]	MOLVL	22.413 962
ロシュミット数※	$n_0$ [m <sup>-3</sup> ]	LOSCHMIDT	2.686 7811 E+25

※セルシウス 0 度、標準大気圧 (273.15[K], 1[atm]) です。

熱輻射に関する定数です。

定数名	記号と単位	キーワード	値
シュテファン・ボルツマン定数	$\sigma$ [W m <sup>-2</sup> K <sup>-4</sup> ]	STEFAN	5.670 367 E-08
第1放射定数	$c_1$ [W m <sup>2</sup> ]	F.RAD	3.741 771 790 E-16
第2放射定数	$c_2$ [m K]	S.RAD	1.438 777 36 E-02

## 14.7 協定値

協定値のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
標準重力加速度	$g_n$ [m/s <sup>2</sup> ]	GRAVITY	9.806 65
標準大気圧	1atm[Pa]	ATM	10 1325
セルシウス 0 度	0°C [K]	ZEROD	273.15

## 14.8 プランク単位

プランク単位のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
プランク質量	$m_P$ [kg]	PL.MASS	2.176 470 E-08
プランクエネルギー	$E_P$ [GeV]	PL.ENERGY	1.220 910 E+19
プランク温度	$T_P$ [K]	PL.TEMP	1.416 808 E+32
プランク長	$l_P$ [m]	PL.LENGTH	1.616 229 E-35
プランク時間	$t_P$ [s]	PL.TIME	5.391 16 E-44

## 14.9 天文学

天文学で使う定数のキーワードは次のとおりです。

定数名	記号と単位	キーワード	値
天文単位	AU [m]	ASTRO	149 597 870 700
パーセク	pc [m]	PARSEC	3.085 677 581 E+16
光年	ly [m]	LYEAR	9 460 730 472 580 800



## 14.10 紙のサイズ

紙のサイズは次のとおりです。

定数名	キーワード	値	定数名	キーワード	値
A0 の短辺 (mm)	A0.S	841	A0 の長辺 (mm)	A0.L	1189
A1 の短辺 (mm)	A1.S	594	A1 の長辺 (mm)	A1.L	841
A2 の短辺 (mm)	A2.S	420	A2 の長辺 (mm)	A2.L	594
A3 の短辺 (mm)	A3.S	297	A3 の長辺 (mm)	A3.L	420
A4 の短辺 (mm)	A4.S	210	A4 の長辺 (mm)	A4.L	297
A5 の短辺 (mm)	A5.S	148	A5 の長辺 (mm)	A5.L	210
A6 の短辺 (mm)	A6.S	105	A6 の長辺 (mm)	A6.L	148
A7 の短辺 (mm)	A7.S	74	A7 の長辺 (mm)	A7.L	105
A8 の短辺 (mm)	A8.S	52	A8 の長辺 (mm)	A8.L	74
A9 の短辺 (mm)	A9.S	37	A9 の長辺 (mm)	A9.L	52
B0 の短辺 (mm)	B0.S	1030	B0 の長辺 (mm)	B0.L	1456
B1 の短辺 (mm)	B1.S	728	B1 の長辺 (mm)	B1.L	1030
B2 の短辺 (mm)	B2.S	515	B2 の長辺 (mm)	B2.L	728
B3 の短辺 (mm)	B3.S	364	B3 の長辺 (mm)	B3.L	515
B4 の短辺 (mm)	B4.S	257	B4 の長辺 (mm)	B4.L	364
B5 の短辺 (mm)	B5.S	182	B5 の長辺 (mm)	B5.L	257
B6 の短辺 (mm)	B6.S	128	B6 の長辺 (mm)	B6.L	182
B7 の短辺 (mm)	B7.S	91	B7 の長辺 (mm)	B7.L	128
B8 の短辺 (mm)	B8.S	64	B8 の長辺 (mm)	B8.L	91
B9 の短辺 (mm)	B9.S	45	B9 の長辺 (mm)	B9.L	64
letter short (mm)	LET.S	215.9	letter long (mm)	LET.L	279.4
legal short (mm)	LEG.S	215.9	legal long (mm)	LEG.L	355.6
letter short (inch)	LET.SI	8.5	letter long (inch)	LET.LI	11
legal short (inch)	LEG.SI	8.5	legal long (inch)	LEG.LI	14



## Chapter 15

# その他の機能

### 15.1 オールクリア

レジスタおよびスタックを一掃したい場合、オールクリアを使います。その他、クリア機能には、次のものがあります。

機能	キーワード
オールクリア	AC
スタッククリア	CLEAR, CLR
レジスタクリア	REGCLEAR, RCLR

これらのクリアを実行した後でも、元に戻す機能を使えます。

### 15.2 オールリセット

計算モードを初期状態にするには、キーワード「Cmd RESET」または「Cmd RST」を単体で入力します。スタック、レジスタ、履歴が初期化されます。

続いて、「Cmd YES」または「Cmd NO」の入力待ちとなるので、実行する場合は「Cmd YES」を、キャンセルする場合は「Cmd NO」を単体で入力します。

### 15.3 元に戻す・やり直し

本ソフトウェアは、スタックおよびレジスタの履歴機能があります。万が一途中で操作を誤っても、元に戻す・やり直しができます。

機能	キーワード
元に戻す	UNDO, U
やり直し	REDO, R

履歴をたどった回数や履歴数は、計算設定・状態表示領域で確認できます。詳しくは、Chapter 2「画面の見方と操作方法」の「計算設定・状態表示領域」をご確認ください。

最大履歴数の設定については、Chapter 3「設定・表示切り替え」の「コンフィグモード

での設定」に詳しい説明があります。

## 15.4 JSON 出力

キーワード「`Cmd` JSON」または「`Cmd` OUT」を単体で入力すると、スタックとレジスタの内容を JSON 形式のテキストファイルに出力できます。

機能	キーワード
JSON 出力	JSON

ファイルは実行ファイルと同じディレクトリに出力されます。ファイル名は、次の形式になります。

eckert\_YYYY\_MMDD\_HHMMSS.json

JSON 出力に成功すると、メッセージ表示領域にファイル名が表示されます。計算結果をテキストで保存したい場合にお役立てください。

## 15.5 マクロ機能

文字列からマクロを実行するには、次のキーワードを用います。

機能	キーワード	R	D
マクロ実行	RUN	1	1

マクロ機能機能は、X の文字列を読み取り、それを入力として実行する機能です。マクロ機能の利用例を示します。

(1) 文字列"2 3 +"をプッシュ

`Cmd` "2 3 +"

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	String	:	2 3 +

(2) マクロ実行

`Cmd` run

#	TYPE	:	VALUE
4:		:	
Z:		:	
Y:		:	
X:	Integer	:	5

この機能は、レジスタ機能と組み合わせて、簡単な独自関数を定義することができます。たとえば、「 $RA + \sqrt{RB \times RC}$ 」という式の文字列"ldra ldrb ldrc \* sqrt +" を RE に登録します。

次に、RA、RB、RC にそれぞれ値を入れます。RE を呼び出し、マクロ実行すると、 $RA + \sqrt{RB \times RC}$  が計算できています。これは、同じ形の計算を、数値を変えつつ何度もう行う場合に有効です。

マクロで使う文字列にマクロ実行のキーワード「`RUN`」を含めても、実行できません。これは、無限ループを防ぐための仕様です。  
また、モード変更・表示変更キーワードを含めても、実行できません。

15.6 精度の検証

精度の検証を行うための機能として、次のようなものを提供します。

機能	キーワード	R	D	演算内容
浮動小数点数の基数	RADIX	0	0	Push Int
計算機イプシロン	EPSILON	0	0	Push Flt

本ソフトウェアの計算精度を検証する場合、これらの機能をお使いください。

15.7 特殊な起動

本ソフトウェアは、コマンドライン引数を受け取って起動できます。

引数	処理内容
-d	画面クリアを行わない
-j	JSON 出力
-jd	JSON 画面表示
--	JSON 機能のための区切り

画面バッファを残したい場合「`-d`」を用います。

```
Cmd: eckert64.exe -d
```

JSON 出力および JSON 画面表示は、ともに、引数「`--`」と一緒に利用します。これ以降に記述された引数郡をコマンドとして処理し、JSON 出力または画面表示します。

例を示します。

```
Cmd: eckert64.exe -j -- 1 2 3 sum stra pi exp strz sum dup i mul 2
```

上の例で「-j」を「-jd」に変えれば、画面表示です。



## Chapter 16

# メッセージ一覧

### 16.1 エラーメッセージ

本ソフトウェアで表示されるエラーメッセージの一覧です。

メッセージ	意味
Bad argument count	引数の個数が不正
Bad argument type	不正な要素の型
Bad element	不正な要素の型
Bad matrix size	不正な行列のサイズ
Bad tuple size	不正なベクトルのサイズ
Determinant is zero	行列式がゼロ
Division by zero	ゼロ除算
Empty input	スペースのみの入力
Failed to output file	ファイル出力に失敗
Final page of register	レジスタの最後のページ
Final page of stack	スタックの最後のページ
First page of register	レジスタの最初のページ
First page of stack	スタックの最初のページ
From _____ to _____ : INVALID	(単位換算) 組み合わせが不正
Invalid conversion	不正な変換
Invalid input	不正な入力
Invalid range	不正な範囲
Invalid value	不正な値
Latest history	最新の状態
Logarithm of zero	ゼロの対数
Maximum integer	最大の整数
Minimum integer	最小の整数
Negative-th power of zero	ゼロの負数乗
No history	履歴がない
No older history	もっとも古い履歴
Not a positive integer	正の整数ではない
Registers are empty	レジスタが空

次のページに続く

前ページからの続き

メッセージ	意味
Selected register is empty	指定レジスタが空
Stack and registers are empty	スタックもレジスタも空
Stack is empty	スタックが空
Too few arguments	引数が足りない
Too large or small input	サポート外の値
Too large to operate	値が大きすぎて処理を中断
Unsupported in current version	現バージョンで非対応
Unsupported operation or notation	サポート外の入力
Zero-th power of zero	ゼロのゼロ乗

## 16.2 確認メッセージ

本ソフトウェアで表示される確認メッセージの一覧です。

メッセージ	意味
Error calculation	計算中にエラーが発生した
Floating overflow	浮動小数点オーバーフロー
Integer overflow	整数オーバーフロー
Rational overflow	有理数オーバーフロー

※オーバーフローとは、桁あふれのことです

## 16.3 入力待ち・入力確認メッセージ

入力待ち・入力確認メッセージの一覧です。

メッセージ	意味
Cancelled	キャンセルされた
Done	実行された
From _____ to _____	(単位換算) ~から~へ
Input integer	整数を入力
Maximum value set	最大値がセットされた
Minimum value set	最小値がセットされた
OK? Y/N	YES/NO のいずれかを入力
Setting completed	正常に設定された



# Chapter 17

## 技術情報

### 17.1 本ソフトウェアの型

本ソフトウェアで扱っているデータの型の詳細は、次のとおりです。

型名	説明	扱える値	種別
Integer	整数	64 ビット整数	数 (スカラー)
Floating	浮動小数点数	long double	数 (スカラー)
Rational	有理数	64 ビット整数 2 つ	数 (スカラー)
Complex	複素数	スカラー 2 つ	数
Boolean	ブール代数	True, False	バイナリ
Byte	バイト	符号なし 8 ビット整数	バイナリ
Word	ワード	符号なし 16 ビット整数	バイナリ
Dword	ダブルワード	符号なし 32 ビット整数	バイナリ
Qword	クアッドワード	符号なし 64 ビット整数	バイナリ
Tuple	ベクトル	数またはバイナリの組	組
Matrix	行列	ベクトルの組	組
Infinity	無限大	プラス、マイナス、複素	非数
String	文字列	文字列	非数
Error	エラー	文字列	非数

整数オーバーフローが検知されると、浮動小数点数にキャストするようになっています。浮動小数点数オーバーフローが検知されると、無限大に変換されるようになっています。

### 17.2 計算精度

本ソフトウェアは、エンジニアが手軽に使える電卓を目指しましたが、精度保証はしていないため、特別な精度が要求される計算には向きません。

内部精度は（あくまで参考値ですが）画面上部の表示桁数の部分に表示されています。実際の内部表現は 2 進数となっているため、浮動小数点演算において、特有の計算機誤差が見込まれます。ちなみに、計算機誤差の補正は行わない仕様となっていますので、あらかじめご了承ください。

## 17.3 数学的定義

本ソフトウェアで採用している数学的定義です。

### 17.3.1 整数剰余

負数を含む整数剰余の定義は次のとおりとします。

<i>A</i>	<i>B</i>	商	剰余
負数	負数	$(-A) \div (-B)$	$-((-A) \bmod (-B))$
負数	負数	$(-A) \div (-B)$	$-((-A) \bmod (-B))$
ゼロ	非ゼロ	0	0
正数	負数	$-(A \div (-B))$	$(-A) \bmod B$
正数	正数	$A \div B$	$A \bmod B$

### 17.3.2 負数の奇数根

負数について、立方根、5乗根などの奇数根は実数で定義しません。たとえば、 $-1$ の立方根を $-1$ とはしません。

負数の奇数根の定義は、複素数の範囲内で行い、次の式に従います。

$$\begin{aligned}\sqrt[n]{a + ib} &= \sqrt[n]{r} \exp(i\theta/n) \\ &= \sqrt[n]{r} (\cos(\theta/n) + i \sin(\theta/n))\end{aligned}$$

### 17.3.3 複素数の定義

複素数の絶対値 `abs` と偏角 `arg` を次のように定義しています。

$$\begin{aligned}\text{abs}(a + ib) &= r = \sqrt{a^2 + b^2} \\ \text{arg}(a + ib) = \theta &= \begin{cases} \text{atan}(b/a) & (a > 0) \\ \pi/2 & (a = 0, b > 0) \\ -\pi/2 & (a = 0, b < 0) \\ \pi - \text{atan}(b/a) & (a < 0, b > 0) \\ \text{atan}(b/a) - \pi & (a < 0, b < 0) \\ \text{Error} & (a = b = 0) \end{cases}\end{aligned}$$

これをもとにして、後述の複素関数を定義します。

## 17.3.4 複素関数

複素関数は次のように定義しています。

関数	定義
平方根	$\sqrt{a+ib} = \sqrt{r}(\cos(\theta/2) + i \sin(\theta/2))$
立方根	$\sqrt[3]{a+ib} = \sqrt[3]{r}(\cos(\theta/3) + i \sin(\theta/3))$
指数関数	$\exp(a+ib) = \exp(a)(\cos a + i \sin b)$
自然対数	$\ln(a+ib) = \ln r + i\theta$
冪乗	$(a+ib)^{c+id} = r^c e^{-d\theta} [\cos(c\theta + d \ln r) + i \sin(c\theta + d \ln r)]$
sin	$\sin(a+ib) = \sin a \cosh b + i \cos a \sinh b$
cos	$\cos(a+ib) = \cos a \cosh b - i \sin a \sinh b$
tan	$\tan(a+ib) = \frac{1}{2} \cdot \frac{\sin 2a}{\cos^2 a + \sinh^2 b} + i \frac{1}{2} \cdot \frac{\sinh 2b}{\cos^2 a + \sinh^2 b}$
asin	$\operatorname{asin}(z) = -i \ln(\sqrt{1-z^2} + zi)$
acos	$\operatorname{acos}(z) = -i \ln(z + i\sqrt{1-z^2})$
atan	$\operatorname{atan}(z) = \frac{i}{2} \ln \left( \frac{i+z}{i-z} \right) \quad (z \neq \pm i)$
sinh	$\sinh(a+ib) = \sinh a \cos b + i \cosh a \sin b$
cosh	$\cosh(a+ib) = \cosh a \cos b + i \sinh a \sin b$
tanh	$\tanh(a+ib) = \frac{\sinh 2a}{\cosh 2a + \cos 2b} + i \frac{\sin 2b}{\cosh 2a + \cos 2b}$
asinh	$\operatorname{asinh}(z) = \ln(z + \sqrt{z^2 + 1})$
acosh	$\operatorname{acosh}(z) = \ln(z + \sqrt{z+1}\sqrt{z-1})$
atanh	$\operatorname{atanh}(z) = \frac{1}{2} \ln \left( \frac{1+z}{1-z} \right) \quad (z \neq \pm 1)$



## Chapter 18

# トラブルシューティング

### 18.1 まったく操作がわからない

ソフトウェアを再起動し、Chapter 4「基本操作～四則演算」をお読みください。本ソフトウェアは逆ポーランド記法を採用しているため、操作には慣れが必要です。

### 18.2 値の全体を表示したい

値の表示の右端が「...」と表示される場合、キーワード「**VIEW**」(VIEW モード)で全体が見られます。VIEW モードからは Enter キーで復帰します。

### 18.3 分数や小数の表示を変えたい

有理数や浮動小数点数の表示を変更するには、次のキーワードを用います。

機能	キーワード
自動小数表示	AD
強制小数表示	FD
強制分数表示	FF
通常表示	STD
固定小数点表示	FIX
指数表示	SCI
エンジニアリング表示	ENG

より詳しくは、Chapter 3「設定・表示切り替え」をお読みください。

### 18.4 複素数の表示を変えたい

キーワード「**EUL**」で、複素数の直交座標表示と極座標表示を切り替えられます。極座標表示は、角度モードに従います。より詳しくは、Chapter 3「設定・表示切り替え」をお読みください。

## 18.5 スタックやレジスタの全要素を一望したい

JSON 出力機能により、スタックとレジスタの全要素をテキストファイルとして閲覧できます。Chapter 15「その他の機能」に詳しい説明があります。

単に特定のデータを見たい場合、スタックやレジスタのページめくりを使います。詳しくは、Chapter 3「設定・表示切り替え」をお読みください。

## 18.6 異常な計算結果が出た

ソフトウェアを再起動し、同じ操作を行って、結果をおたしかめください。なお、本ソフトウェアでは、内部的に数を 2 進数で管理しているため、特有の計算機誤差を生じる場合があります。「0.1 となるはずが、0.0999... のように表示される」という計算機誤差は仕様の範囲内ですので、あらかじめご了承ください。

### 18.6.1 キーワードを確認

スペルの似た他のキーワードを使用していませんか。呼び出す演算のキーワードが誤っていないかご確認ください。

### 18.6.2 表示モードを確認

見づらい値の表示になっていると、違和感が出るかもしれません。表示モードを変更し、結果を確認しましょう。モード変更については、Chapter 3「設定・表示切り替え」をお読みください。

### 18.6.3 角度モードを確認

計算しようとしている式の角度の単位を誤ると、期待と異なる結果が出ます。三角関数や複素偏角の計算の場合、指定されている角度モードによって、異なる関数が呼ばれます。たとえば、度数法モードの際にサインを指定すると度数法のサインが実行されますが、ラジアンモードのときはラジアンサインが実行されます。モード変更については、Chapter 3「設定・表示切り替え」をお読みください。

### 18.6.4 入力する値の大きさを確認

関数によっては、入力される値によって、著しい誤差を生じるものがあります。たとえば、三角関数は非常に大きな値の入力に対しては、信頼できない結果となります。また、複素関数が定義できる場合については、実数の範囲の定義域も確認しましょう。Chapter 17「技術情報」も併せてお読みください。


### 18.6.5 演算の順番を確認

数学的に同じ結果になる式でも、演算の順番によって、誤差を生じやすくなる場合があります。誤差を生じにくい式に変形して試しましょう。

## 18.7 エラーになって計算できない

計算しようとしているデータの型をおたしかめください。たとえば、浮動小数点数の階乗は計算できません。

### 18.7.1 型を確認

画面左側、TYPE の列で型を確認しましょう。表示されていない場合、キーワード「 TYPE」で表示できます。型によっては定義不能な演算となります。

### 18.7.2 値を確認

異常な値で計算しようとしていませんか。型が正常でも、値によっては演算を定義できない場合があります。たとえば、ゼロの対数は定義されません。

### 18.7.3 ベクトルや行列のサイズを確認

計算しようとしているベクトルや行列のサイズを確認しましょう。ベクトルの場合、向きにも注意しましょう。

### 18.7.4 エラーメッセージを確認

Chapter 16「メッセージ一覧」から、エラーメッセージの意味を確認できます。

## 18.8 バグと思われる挙動を見つけた

画面表示の乱れや、疑わしい計算結果・挙動にお気づきの場合、再現手順を添えて、ご連絡ください。数学的に不自然な仕様についても、報告していただけると助かります。

作者宛て E メールアドレス：<mailto:only.my.truth@gmail.com>