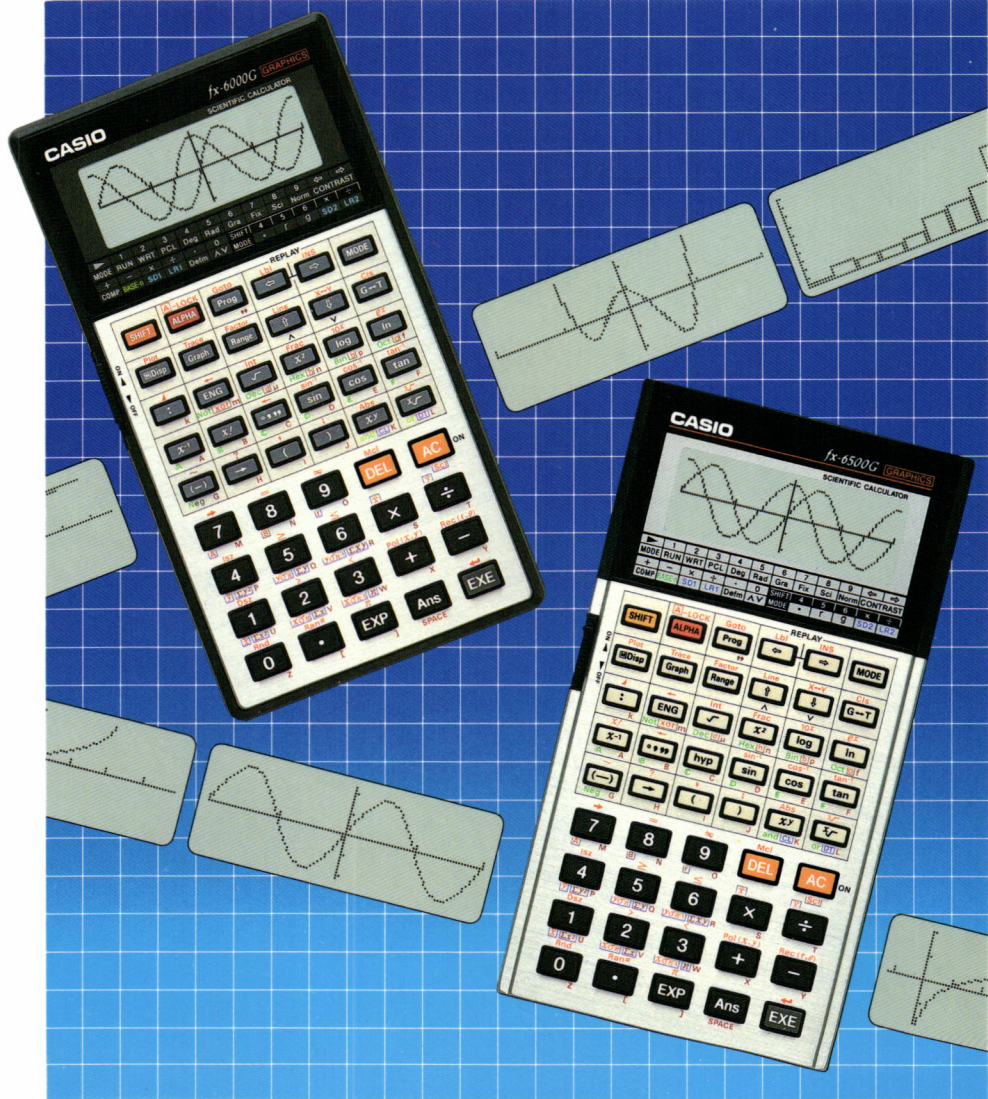


fx-6000G/fx-6500G

操作マニュアル

取扱説明書つき

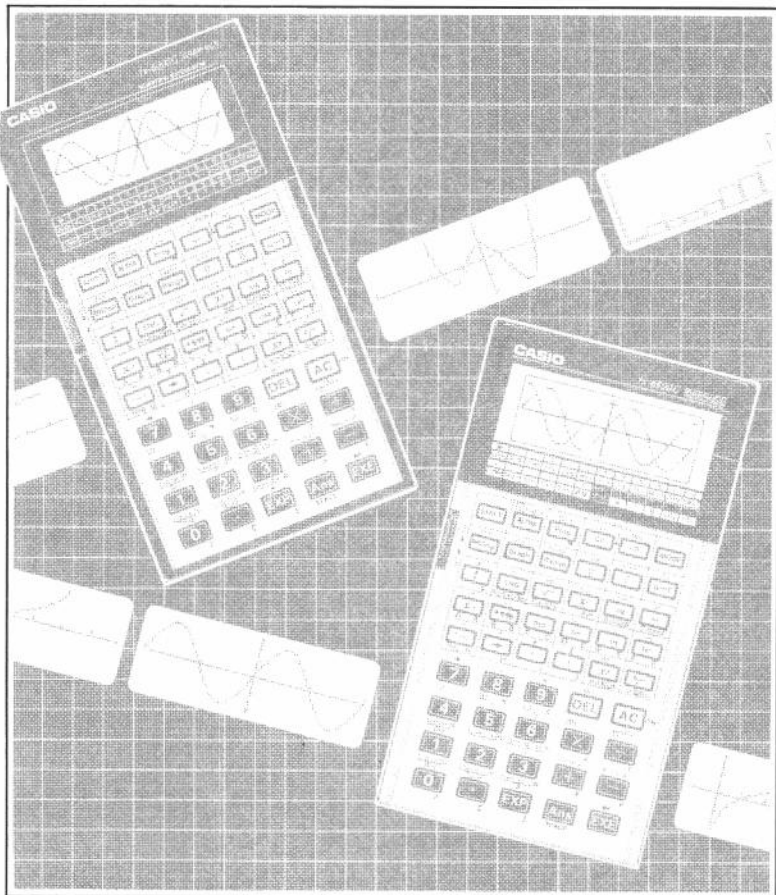


CASIO®

fx-6000G/fx-6500G

操作マニュアル

取扱説明書つき



CASIO®

- 本書の内容に関しては、将来予告なしに変更することがあります。
- 本書の内容については万全を期して作成いたしましたが、万一不審な点や誤りなど、お気づきのことがありましたらご連絡ください。
- 本書の一部または全部を無断で複製することは禁止されています。また、個人としてご利用になるほかは、著作権法上、当社に無断では使用できませんのでご注意ください。
- 本書使用による損害および逸失利益等につきましては、当社では一切その責任を負いかねますので、あらかじめご了承ください。
- 本書中のグラフィック表示は、印刷のため実物と異なることがあります。

はじめに

このたびはカシオグラフィック関数電卓をお買い上げいただきまして、誠にありがとうございます。

本機は、高級プログラム電卓に、いろいろなグラフが描けるグラフ機能を加えた、まったく新しいグラフィック電卓です。

マニュアル計算も書式通りに入力するだけで簡単に計算でき、もしキー操作をまちがっても確認や訂正がワンタッチでできるリプレイ機能を備えています。

プログラムも書式通りの入力方式で、面倒な命令も少なく、簡単な繰り返し計算から複雑な計算まで便利に使えます。

本書は

第1章 本体構成と使い方

第2章 マニュアル計算

第3章 グラフィック

第4章 プログラム計算

の4章から構成されています。

まずは第1章をお読みになり、各部の名称や使い方、ご使用上の注意を覚えてください。第2章、第3章、第4章ではそれぞれの基本的なご使用方法について例題とともに説明してありますので、よくお読みいただき、各機能を十二分にマスターの上、ご使用ください。

なお、本書は fx-6500G と fx-6000G の両機について書かれています。両機の違いは fx-6000G にはハイパボリックがなく、“x!” が **SHIFT** キーを使わずに入力できる点ですので、各項目をご覧になり、ご使用ください。

目 次

はじめに

第1章 本体構成と使い方	1
1-1 各部の名称とその説明	2
表示窓	3
電源スイッチ	3
特殊操作キー	3
置数キー(数字キー、小数点キー、指数部入力キー)	7
計算キー	7
グラフィックキー	8
関数キー	9
コントラスト調整	12
1-2 電源および電池交換について	13
電池交換の仕方	13
1-3 計算をはじめる前に	15
計算の優先順位	15
スタック数	16
計算モードについて	17
入出力桁数と演算桁数	18
桁オーバーとエラーについて	19
入力文字数	20
グラフィック表示とテキスト表示	20
表示用レジスタについて	21
訂正について	22
メモリーについて	23
メモリーの増設	24
アンサー(Ans)機能	25
オートパワーオフ(自動電源OFF)機能について	26
第2章 マニュアル計算	27
2-1 基本計算の仕方	28
加減乗除計算	28
カッコ計算	29
メモリー計算	30
小数点以下指定、有効桁数指定計算および指数表示指定	31

2-2 特別機能	33
連続演算機能	33
リプレイ機能	34
マルチステートメント機能	36
2-3 関数計算の仕方	37
角度単位変換、角度計算	37
三角関数(\sin 、 \cos 、 \tan)、逆三角関数(\sin^{-1} 、 \cos^{-1} 、 \tan^{-1})	38
対数関数(\log 、 \ln)、指数関数(10^x 、 e^x 、 x^y 、 $\sqrt[x]{y}$)	39
双曲線関数(\sinh 、 \cosh 、 \tanh)、逆双曲線関数(\sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1})	40
座標変換(Pol、Rec)	41
その他の関数($\sqrt{\quad}$ 、 x^2 、 x^{-1} 、 $x!$ 、 $\sqrt[x]{\quad}$ 、RAN#、Abs、Int、Frac)	42
2-4 2進・8進・10進・16計算の仕方	44
2進・8進・10進・16進変換	45
負数の表現	46
2進・8進・10進・16進の加減乗除算	46
論理演算	47
2-5 統計計算の仕方	48
標準偏差計算	48
回帰計算	50
直線回帰計算	51
対数回帰計算	52
指数回帰計算	53
べき乗回帰計算	54
第3章 グラフィック	55
3-1 組み込み関数グラフ	57
組み込み関数グラフの重ね描き	58
3-2 任意の関数グラフ(式を目で見る)	60
レンジ(Range)	60
任意の関数グラフを描く	65
関数グラフの重ね描き	65
トレース(Trace)機能	66
プロット(Plot)機能	69
ライン(Line)機能	71
ファクター(Factor)機能	73
3-3 グラフ関数の応用	77

3-4 1変数統計グラフ	79
1変数統計グラフの描き方	79
〈まとめ〉	82
3-5 2変数統計グラフ	83
2変数統計グラフの描き方	83
第4章 プログラム計算	85
4-1 プログラムとは?	86
計算式を求める	86
プログラミング	86
プログラムの記憶	88
プログラムの実行	89
4-2 プログラムのチェックおよび編集(訂正・追加・削除)	93
計算式を求める	93
プログラミング	93
プログラムの記憶(編集)	94
プログラムの実行	95
〈まとめ〉	97
4-3 プログラムのデバッグ(まちがいを直す)	98
エラー表示によるデバッグ	98
エラーメッセージ	98
エラーメッセージのチェックポイント	100
4-4 ステップ数のかぞえ方	101
4-5 プログラムエリアと計算モード	103
WRTモードでのプログラムエリア指定と計算モード	
(Comp、Base-n、SD、LR)指定	104
計算モードにおける注意点	105
4-6 プログラムの消し方	106
単一プログラムの消去	106
全プログラムの消去	107
4-7 便利なプログラム命令	108
ジャンプ命令	108
無条件ジャンプ	108
条件ジャンプ	110
カウントジャンプ	112
〈まとめ〉	114

サブルーチン	114
キャリッジリターン機能	117
4-8 メモリーの配列的使い方	119
配列的使い方とは?	119
配列的使い方の注意点	120
配列的使い方の応用	122
4-9 アルファ文字の表示	124
アルファ文字	124
4-10 グラフィック機能をプログラムに組む	127
ライブラリー編	129
プログラム No. 1 素因数分解	130
プログラム No. 2 最大公約数	132
プログラム No. 3 シンプソン法による定積分	134
プログラム No. 4 $\Delta \leftrightarrow Y$ 変換	136
プログラム No. 5 最小損失整合	138
プログラム No. 6 集中荷重の片持梁	140
プログラム No. 7 放物線運動	142
プログラム No. 8 正規分布	144
プログラム No. 9 円と接線	146
プログラム No.10 図形の回転	152
プログラム No.11 パラメータによるグラフの変化を見る	156
プログラム No.12 ヒステリシスループ	160
プログラム No.13 回帰曲線	164
プログラム No.14 パレード図	172
プログラムシート	176
付 録	181
マニュアル計算	182
プログラム計算	186
エラーメッセージ一覧表	188
関数桁容量(原則として)	190
規 格	192
カシオサービスセンター	194
保 証 書	195

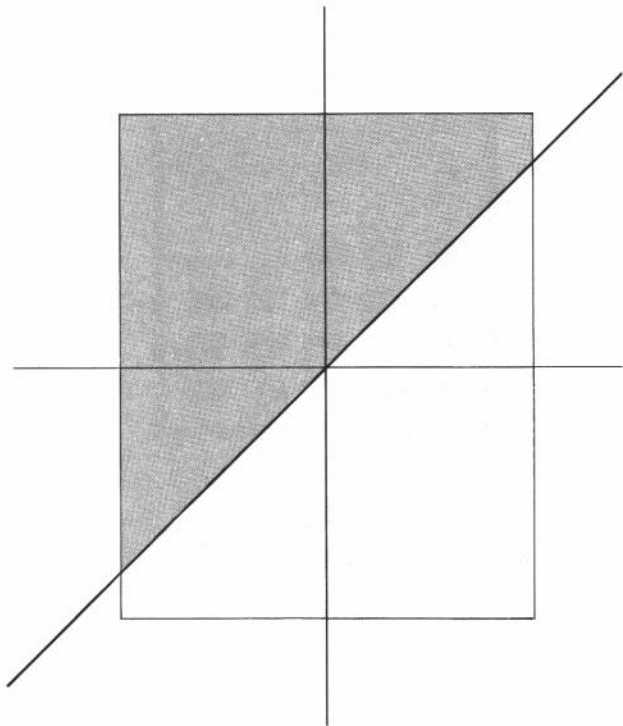
■ご使用上の注意

- 計算機は精密な電子部品で構成されていますので、絶対に分解しないでください。
また、投げたり落したり等のショックや、急激な温度変化を与えないでください。
特に、高温の所、湿気やホコリの多い所に放置したり保管することはしないでください。
なお、温度が低いときは表示の応答速度が遅くなったり、点灯しなくなることがありますが、通常の温度になると正常にもどります。
- 電池は、計算機を使わない場合でも2年に1度は交換してください。
特に消耗済みの電池を放置しておきますと、液もれをおこし、故障等の原因になりますので、計算機内には絶対に残しておかないでください。
- 計算機のお手入れは、シンナー・ベンジン等の揮発性液体をさけ、「乾いた柔らかい布」あるいは、「中性洗剤液に浸し固くしぼった布」でおふきください。

■保証・アフターサービス

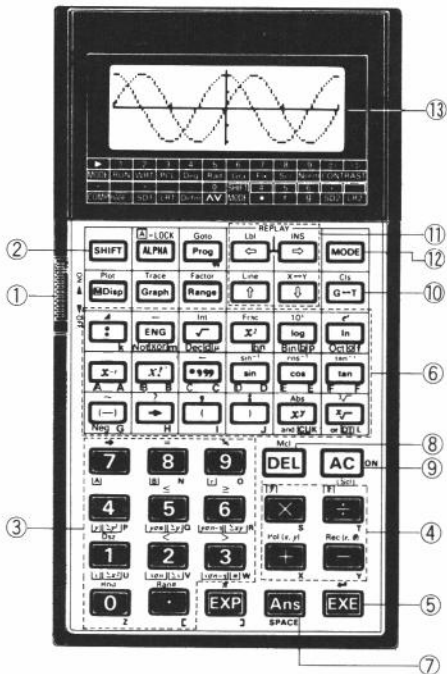
- 保証は、巻末の保証書の内容によりますので、よくお読みのうえ、記入事項を確認して、大切に保管してください。
- 万一故障したときは
①お買い上げ店 ②カシオ計算機サービスセンター
のうち、ご都合のよい所へ、必ず保証書をそえて、ご持参またはご郵送ください。
この場合、故障内容を具体的にお知らせください。
- 修理依頼される前には、この説明書をもう一度お読みになると共に、電源の状態および、プログラムミス、操作ミスがないかをよくお調べください。
- ご不明の点やご質問、お問い合わせ等は、194ページのカシオ計算機へ直接ご連絡ください。

第1章 本体構成と使い方

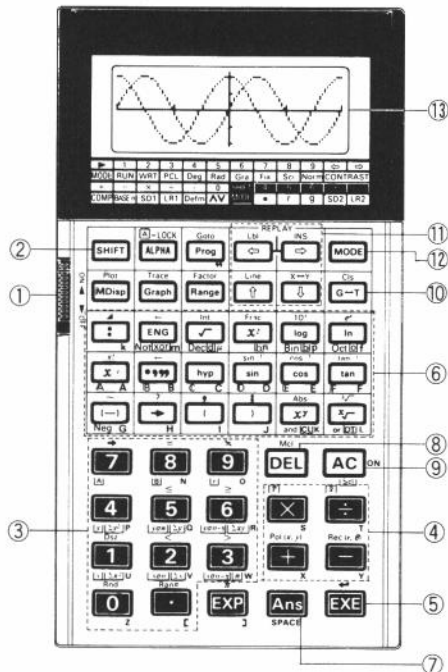


1-1 各部の名称とその説明

fx-6000G



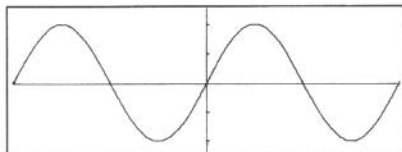
fx-6500G



- | | |
|-------------|---------------|
| ① 電源スイッチ | ⑧ デリートキー |
| ② シフトキー | ⑨ オールクリアーキー |
| ③ 置数キー | ⑩ グラフキー |
| ④ 四則計算キー | ⑪ カーソル/リプレイキー |
| ⑤ エグゼキュートキー | ⑫ モードキー |
| ⑥ 関数キー | ⑬ 表示窓 |
| ⑦ アンサーキー | |

■表示窓

***	MODE	***
RUN	:	COMP
Deg	:	Norm
Step		0



表示窓は16桁、4行の文字表示または、横95×縦32ドットのグラフを表示します。
※表示窓は横96×縦32ドットで構成されていますが、座標軸からの対称を保つため、左端1ドットは表示されません。

左の表示はシステム表示で、動作モードと演算モードの設定、角度単位、小数点以下有効桁数の設定、キー入力バッファの状態(Step)などを表示しています。
右の表示はサインカーブのグラフで、このようなグラフ表示もできます。
なお、数字の0(ゼロ)はアルファベットのO(オー)と区別するために“0”と表示します。

■電源スイッチ

ツマミを上にはスライドすると電源が入り、下にはスライドさせると電源が切れます。

■特殊操作キー

SHIFT シフトキー

キーパネル面に茶色で記されている関数命令や機能を使うとき押します。

SHIFT キーを押すと“S”が点滅し、SHIFT キーが押されたことを示します。続けて SHIFT キーを押すと“S”が消え、SHIFT キーを押す前の状態になります。

MODE モードキー

計算機の状態および角度の単位を指定するとき押します。

MODE 0 ……グラフ表示中に押しますと、縦方向(Y座標)のピッチを半分にしてグラフを表示します。このときX座標とY座標の比率は2:1となります。

また、Y座標のピッチが半分のときにもう一度 MODE 0 と押しますと、元のピッチに戻ります。

MODE 1 ……マニュアル計算やプログラムの実行が行なえます。

MODE 2 ……プログラムの書き込みやチェックが行なえます。

- MODE** **3** ……プログラムクリアが行なえます。
- MODE** **4** ……“Deg”と表示され、続けて **EXE** キーを押すと、角度の単位を「度」に指定します。
- MODE** **5** ……“Rad”と表示され、続けて **EXE** キーを押すと、角度の単位を「ラジアン」に指定します。
- MODE** **6** ……“Gra”と表示され、続けて **EXE** キーを押すと、角度の単位を「グラッド」に指定します。
- MODE** **7** ……“Fix”と表示され、続けて **0** ~ **9** **EXE** と押すことにより、小数点以下を 0 ~ 9 に指定します。
例) **MODE** **7** **3** **EXE** → 小数点以下 3 桁指定
- MODE** **8** ……“Sci”と表示され、続けて **0** ~ **9** **EXE** と押すことにより、有効桁数を 1 ~ 10 に指定します。
例) **MODE** **8** **5** **EXE** → 有効桁数 5 桁指定
- MODE** **9** ……“Norm”と表示され、続けて **EXE** キーを押すと、小数点以下や有効桁数の指定を解除します。
- MODE** **□** ……“Defm”と表示され、続けて数値を入力して **EXE** キーを押せば、使用可能なメモリー数を指定できます。
例) **MODE** **□** **10** **EXE** → メモリーを 10 個増す。
メモリー数を入力せずに **EXE** キーを押すと、現在使用可能なメモリー数と残りステップ数を表示します。(24 ページ参照)
例) **MODE** **□** **EXE**

```

**Defm**
Program : 56
Memory : 36
350 Bytes Free

```

- MODE** **田** ……四則計算や関数計算を行なう COMP モードに指定します。(プログラム実行可)
- MODE** **日** ……2 進・8 進・10 進・16 進計算や変換が行なえます。
- MODE** **×** ……標準偏差計算が行なえます。(SD 1 モード)
- MODE** **+** ……回帰計算が行なえます。(LR 1 モード)
- SHIFT** **MODE** **×** ……1 変数統計データによる棒グラフ、折れ線グラフ 正規分布曲線を描くことができます。(SD 2 モード)

SHIFT **MODE** **+**……2変数統計データによる回帰直線グラフを描くことができます。(LR2モード)

SHIFT **MODE** **4**……数値を度単位で入力するとき、数値の後に押します。

SHIFT **MODE** **5**……数値をラジアン単位で入力するとき、数値の後に押します。

SHIFT **MODE** **6**……数値をグラッド単位で入力するとき、数値の後に押します。

ALPHA アルファキー

アルファベットや特殊記号を表示させるとき押します。“**A**”が点滅し **ALPHA** キーが押されたことを示します。続けて **ALPHA** キーを押すと“**A**”が消え、**ALPHA** キーが押される前の状態に戻ります。

なお、**SHIFT** キーに続けて押すとアルファ文字入力状態がロック(固定)され、連続してアルファ文字を入力できます。

”

k	m	μ	n	ρ	f
A	B	C	D	E	F
G	H	I	J	K	L
M	N	O			
P	Q	R	S	T	
U	V	W	X	Y	
Z	[]	SPACE		

※本書ではアルファ文字を使うときのキー操作を **ALPHA** **A**、**ALPHA** **X** のように記します。

Goto **Prog** プログラムキー

プログラムを実行するとき、**Prog** **0** ~ **9** **EXE** と押します。

例) **Prog** **1** **EXE** → P1 のプログラム実行開始

SHIFT キーに続けて押しますと“Goto”を表示し、プログラム中の分岐(ジャンプ)命令となります。

Lbl **REPLAY** **INS** **Line** **X**→**Y** **↑** **↓** カーソル/リプレイキー

表示されている計算式や数値を訂正するとき、カーソル(表示窓内の点滅している“-”)を左右、上下に移動させるために押します。**←**キーはカーソルを左に、**→**キーはカーソルを右に、**↑**キーはカーソルを上、**↓**キーは

カーソルを下に移動させます。これらのキーを押し続けると表示されている計算式や数値の間を、カーソルが連続して移動します。

[⇐] キーと **[⇒]** キーは **[EXE]** キーを押した後はリプレイキーとなり、**[⇐]** キーを押すと計算式の前からカーソルが点滅し、**[⇒]** キーを押すと計算式の後からカーソルが点滅し、再び実行させたり、数値等をかえてからの実行ができます。

[MODE] **[0]** により $x \cdot y$ 座標のピッチが1対1のとき、**[↑]** キーと **[↓]** キーはグラフ表示中、グラフを y 座標方向に3分割し、上・中・下と切り換えて表示させます。(57ページ参照)

[SHIFT] キーに続けて押しますと、**[Lbl]** (**[⇐]**) キーはプログラム中のラベル入力に、**[INS]** (**[⇒]**) キーは数値等を挿入するときに使います。**[Line]** (**[↑]**) キーは折れ線グラフや回帰直線を描くときに、**[x→y]** (**[x→y]**) キーはグラフのトレース中の X 座標値、 Y 座標値の表示切り替えのために使います。

表示の濃度を調整するとき、**[MODE]** キーに続けて押します。(12ページ)

[^{Mc}DEL] デリートキー

カーソルと文字が点滅している箇所を削除したいとき押します。

[SHIFT] **[Mc]** **[EXE]** と押しますと、メモリー内をクリアーします。

[AC] オールクリアーキー

表示されている計算式や数値(テキスト表示)を全て消したいときに押します。(入力バッファの内容も全てクリアーされます)

また、エラーチェック“**〇〇ERROR**”表示のとき、エラー状態を解除するために押します。

オートパワーオフ(26ページ参照)が働いて、表示が消えているときに電源を入れるために押します。

[EXE] エクゼキュートキー

計算の答を求めるときやグラフを描くとき押します。

プログラム計算などで、データを入力するときの最後や、計算結果が表示されているときに、次の実行に移すために押します。

[Ans] アンサーキー

[EXE] キーが押され実行された演算結果を呼び戻します。

プログラムが実行された場合は、最後の演算結果を呼び戻します。

■ ① ~ ⑨ ◻ [EXP] 置数キー(数字キー、小数点キー、指数部入力キー)

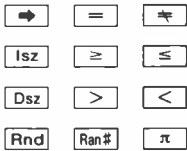
置数(計算機に数値を入れること)のとき、数の上桁より順に入れます。

◻ は小数点の位置で押します。

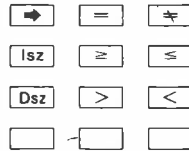
指数部を置数するときは1.23 [EXP] 6 (=1.23×10⁶)と入力します。

モードにより、[SHIFT] キーとの組み合わせで次のように表示されます。

COMP モード ([MODE] ⊕)

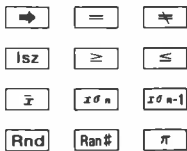


Base-n モード ([MODE] ⊖)



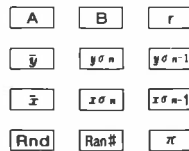
COMP モードとは異なり、Pol(、Rec(、Rnd、Ran#、πの関数は使えません。

SD モード ([MODE] ⊗)



標準偏差用の関数が使えます。

LR モード ([MODE] ⊕)



2変数統計用の関数が使えます。

■ 計算キー

⊕ ⊖ ⊗ ⊕ 四則計算キー

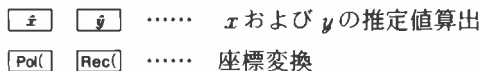
加減乗除計算をするとき、書式通りにそれぞれの位置で押します。

モードにより [SHIFT] キーとの組み合わせで、次のように表示されます。

COMP モードおよび SD モード



LR モード



■グラフィックキー

いろいろなグラフを描くときに使います。詳しい使い方は55ページをご覧ください。

なお、Base-n モードではグラフィックキーは使えません。

モード表示キー/プロットキー

- 動作モードや演算モード、角度単位や丸めなどの設定状態を確認したいとき押します。押している間だけ設定状態を表示します。
- [SHIFT]** キーに続けて押しますと、グラフィック画面上に点を描く命令となります。

グラフキー/トレースキー

- グラフを描くとき、計算式の前に押します。“Graph Y=”と表示します。
- [SHIFT]** キーに続けて押しますと、描いたグラフの上をトレース(跡をたどる)し、 x または y 座標の値を表示します。

レンジキー/ファクターキー

- グラフの範囲や大きさを確認したり、設定したりするとき押します。
- [SHIFT]** キーに続けて押しますと、レンジの上限・下限の内容を変化させ、レンジの拡大・縮小ができます。

グラフ・テキスト切替キー/クリアースクリーンキー

- グラフィック画面とテキスト画面(20ページ参照)を切り替えるとき押します。
- [SHIFT]** **[Cis]** **[EXE]** と押しますと、グラフィック画面のグラフをクリアーします。テキスト画面はクリアーしません。

■関数キー

関数計算を行なうときに押します。モードや **[SHIFT]** キーとの組み合わせによりいろいろな使い方がありますので、よく覚えてください。

マルチステートメント/表示キー

- プログラム計算や連続計算のとき、計算式や命令の区切りとして押します。
：(コロン)で続けることをマルチステートメント(36ページ参照)と言います。
- [SHIFT]** キーに続けて押しますと、プログラム計算や連続計算のときの結果表示(出力)のために押します。

ENG エンジニアリングキー

- [EXE]** キーにより求められた計算結果を指数表示およびその指数部が3の倍数になるように変換したいとき押します。(3の倍数： $10^3 = \overset{\text{キロ}}{\text{K}}$ 、 $10^6 = \overset{\text{メガ}}{\text{M}}$ 、 $10^9 = \overset{\text{ギガ}}{\text{G}}$ 、 $10^{-3} = \overset{\text{ミリ}}{\text{m}}$ 、 $10^{-6} = \overset{\text{マイクロ}}{\mu}$ 、 $10^{-9} = \overset{\text{ナノ}}{\text{n}}$ 、 $10^{-12} = \overset{\text{ピコ}}{\text{p}}$)
- Base-n モードでは論理演算の否定を求めるとき、数値の前に押します。
- Base-n モードで排他的論理和を求めるとき **[SHIFT]** キーに続けて押します。

√ ルート/インテジャーキー

- 数値の平方根を求めるとき、数値の前に押します。
- [SHIFT]** キーに続けて押しますと、数値の整数部だけを取り出します。
- Base-n モードでは **[Dec]** **[EXE]** と押すことにより、10進数による計算モードの指定となります。
- Base-n モードで **[SHIFT]** キーに続けて押しますと、次の数値を10進数に指定します。

x² 2乗/フラクシオンキー

- 数値の2乗(自乗)を求めるとき、数値の後に押します。
- [SHIFT]** キーに続けて押しますと、数値の小数部だけを取り出します。
- Base-n モードでは **[Hex]** **[EXE]** と押すことにより、16進数による計算モードの指定となります。
- Base-n モードで **[SHIFT]** キーに続けて押しますと、次の数値を16進数に指定します。



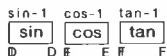
ログ/10の x 乗キー

- 数値の常用対数(10を底とする対数)を求めるとき、数値の前に押します。
- 10の数値乗を求めるとき、数値の前に **SHIFT** **10^x** と押します。
- Base-n モードでは **Bin** **EXE** と押すことにより、2進数による計算モードの指定となります。
- Base-n モードで **SHIFT** キーに続けて押しますと、次の数値を2進数に指定します。



エルエヌ/ e の x 乗キー

- 数値の自然対数(e を底とする対数)を求めるとき、数値の前に押します。
- e の数値乗を求めるとき、数値の前に **SHIFT** **e^x** と押します。
- Base-n モードでは **Oct** **EXE** と押すことにより、8進数による計算モードの指定となります。
- Base-n モードで **SHIFT** キーに続けて押しますと、次の数値を8進数に指定します。



三角関数/逆三角関数キー

- 数値の三角関数(sin, cos, tan)を求めるとき、数値の前にそれぞれ押します。
- 数値の逆三角関数(\sin^{-1} , \cos^{-1} , \tan^{-1})を求めるとき、数値の前に **SHIFT** **\sin^{-1}** のように押します。
- Base-n モードでは16進数の D、E、F (=13₁₀、14₁₀、15₁₀)を置数するとき押します。



マイナスキー

- 負数を置数するとき、数値の前に押します。
例) $-123 \rightarrow$ **(-)** **1** **2** **3**
- 複数のメモリーに同じ数値を代入するとき、**SHIFT** キーに続けて押します。
例) 456をメモリー A~F に代入する \rightarrow **4** **5** **6** **→** **ALPHA** **A** **SHIFT** **~** **ALPHA** **F** **EXE**
- Base-n モードでは数値の負数を求めるとき、数値の前に押します。負数は置数の2の補数をとります。

代入キー

- 演算結果等をメモリーに代入するとき、メモリーの前に押します。

例) $12+45=$ の答えをメモリーAに代入する→**11** **2** **+** **4** **5** **→** **ALPHA** **A** **EXE**

- プログラム計算や連続計算の中で、数値を入力したいところで **SHIFT** **?** と押します。

カッコキー

- 計算式の開きカッコの位置で **[** キーを、閉じカッコの位置で **]** キーを押します。

- 座標変換や統計計算などの引数を区切るための、(カンマ)や;(セミコロン)を入力するとき **SHIFT** キーに続けて押します。

べき乗/絶対値キー

- x (任意の数)の y (任意の数)乗を求めるとき、 x と y の間で押します。
- 数値の絶対値を求めるとき、数値の前に **SHIFT** **Abs** と押します。
- Base-n モードでは論理積を求めるとき押します。
- SD モードおよび LR モードでは入力したデータを削除するとき押します。(**SHIFT** キーに続けて押すと x が入力できます)

べき乗根/立方根キー

- y の x 乗根を求めるとき、 x と y の間で押します。
- 数値の立方根を求めるとき、数値の前に **SHIFT** **$\sqrt[3]{\quad}$** と押します。
- Base-n モードでは論理和を求めるとき押します。
- SD モードおよび LR モードではデータ入力キーとして押します。(**SHIFT** キーに続けて押すと $\sqrt{\quad}$ が入力できます)

<fx-6500G をお使いの方は>

逆数/階乗キー

- 数値の逆数を求めるとき、数値の後に押します。
- 数値の階乗を求めるとき、数値の後に **SHIFT** **$x!$** と押します。
- Base-n モードでは16進数の A (=10₁₀) を置数するとき押します。

度分秒キー (10進数 ↔ 60進数変換キー)

- 度・分・秒(時・分・秒)のような60進数を置数するとき押します。

例) $78^{\circ}45'12'' \rightarrow 78$ **[0.]** **45** **[0.]** **12** **[0.]**

- 10進数の数値を度分秒(時分秒)で表示させたいとき、**SHIFT** キーに続いて押します。
- Base-n モードでは16進数の B (=11₁₀) を置数するとき押します。

$\boxed{\text{hyp}}$ ハイバボリックキー

- 数値の双曲線関数 (\sinh 、 \cosh 、 \tanh) を求めるとき、数値の前に $\boxed{\text{hyp}}$ $\boxed{\sin}$ のように押します。
- 数値の逆双曲線関数 (\sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1}) を求めるとき、数値の前に $\boxed{\text{SHIFT}}$ $\boxed{\text{hyp}}$ $\boxed{\sin^{-1}}$ のように押します。 ($\boxed{\text{hyp}}$ $\boxed{\text{SHIFT}}$ $\boxed{\sin^{-1}}$ のように押しても求められます)
- Base-n モードでは16進数の C (=12₁₀) を置数するとき押します。

〈fx-6000G をお使いの方は〉

$\boxed{x^{-1}}$ 逆数キー

- 数値の逆数を求めるとき、数値の後に押します。
- Base-n モードでは16進数の A (=10₁₀) を置数するとき押します。

$\boxed{x^{\square}}$ 階乗キー

- 数値の階乗を求めるとき、数値の後に押します。
- Base-n モードでは16進数の B (=11₁₀) を置数するとき押します。

$\boxed{\circ\ \dots\ \circ}$ 度分秒キー (10進数 \leftrightarrow 60進数変換キー)

- 度・分・秒 (時・分・秒) のような60進数を置数するとき押します。
例) 78°45'12" → 78 $\boxed{\circ\ \dots\ \circ}$ 45 $\boxed{\circ\ \dots\ \circ}$ 12 $\boxed{\circ\ \dots\ \circ}$
- 10進数の数値を度分秒 (時分秒) で表示させたいとき、 $\boxed{\text{SHIFT}}$ キーに続いて押します。
- Base-n モードでは16進数の C (=12₁₀) を置数するとき押します。

■コントラスト調整

$\boxed{\text{MODE}}$ キーに続けて $\boxed{\leftarrow}$ 、 $\boxed{\rightarrow}$ キーを押しますと、表示の濃度を調整することができます。 $\boxed{\leftarrow}$ キーを押すと表示は薄くなり、 $\boxed{\rightarrow}$ キーを押すと濃くなります。 $\boxed{\leftarrow}$ 、 $\boxed{\rightarrow}$ キーとも押し続けることにより、続けて薄くまたは濃くなります。なお、 $\boxed{\text{MODE}}$ 、 $\boxed{\leftarrow}$ 、 $\boxed{\rightarrow}$ (および $\boxed{\uparrow}$ $\boxed{\downarrow}$) キー以外のキーを押すと、コントラスト調整は解除されます。

※コントラスト調整を最も濃くしてもまだ表示が薄いときは、電池の消耗が考えられますので、なるべく早く電池を交換してください。

※ $\boxed{\text{Range}}$ キーを押して表示されるレンジ表示中 (60ページ) では、コントラスト調整はできません。

1-2 電源および電池交換について

本機は、リチウム電池《CR-2032C》3個を電源として使用します。

電池が消耗しますと、表示窓の数字や文字がうすくなり、大変見にくくなります。コントラスト調整(12ページ参照)しても表示が薄いときは、電池が消耗していますので、電池交換を行なってください。なお、電池は3個ともいっしょに交換してください。

※電池は2年以上使用した場合、液もれをおこす危険がありますので、使わない場合でも2年に1度は必ず電池交換をしてください。

※本機に組み込まれている電池はモニター用の電池ですので、所定の時間に満たないうちに消耗することがあります。

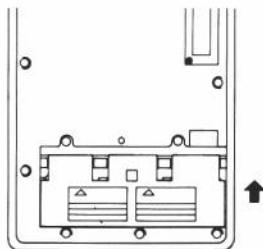
※電池交換を行ないますと、記憶されているプログラムやデータは消えますので、交換後もプログラムやデータをお使いになる場合には、コーディングシート等へ書き止めておくことをおすすめします。

※電池交換の際はカシオ指定の電池をお使い下さい。

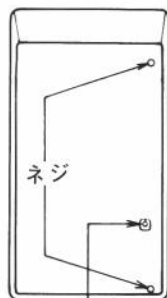
■電池交換の仕方

- ①電源スイッチを切ってから、裏面にあるネジをドライバーではずし、裏ブタをとります。
- ②電池押え板を矢印方向にスライドしてはずします。
- ③古い電池を3個とも取り出します。

(電池ボックスを下に向けて軽く)
たたけば簡単にはずれます。)

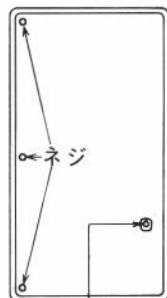


fx-6000G



オールリセットボタン

fx-6500G

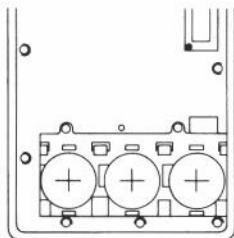


オールリセットボタン

④新しい電池の表面を乾いた布でよくふき、⊕側を上にして入れます。

⑤電池押え板をはずすときは逆にスライドさせてとめ、裏ボタンもネジ止めします。

※消耗済みの電池は絶対に火中に投下しないでください。破裂することがあり非常に危険です。



【ご注意】

交換前の電池容量が残っている場合は、電池交換を手早く行なえばメモリー内容、プログラム内容が消えたり、変ったりすることはありませんが、必ずそれぞれの内容をご確認ください。

また、電池容量が低下している場合や、電池を長くはずしておきますと、メモリー内容やプログラム内容が消えたり変化していることがありますので、このようなときには電池交換後本体裏面のオールリセットボタンを先のとがった細い棒などで押してください。全てのメモリーやプログラムがクリアーされます。

また、表示のコントラストも自動的に中位置にセットされますので、見やすい濃度に調整してください。

★電池交換後にオールリセットボタンを押しても効かない場合には、電池を取り出してから、3分以上放置してから電池を入れ直してください。

電池は、幼児の手のとどかないところに保管してください。

万一、飲み込んだ場合にはただちに医師と相談してください。

1-3 計算をはじめる前に

■計算の優先順位

計算には「優先順位」という規則があり、たし算・ひき算よりかけ算・わり算の方を先に計算することになっています。また、関数の方が優先され、関数の中でも前置関数より後置関数の方が優先されます。本機はこの優先順位を計算機自身が自動的に判別します。優先順位は次のようになっています。

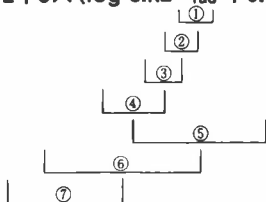
- ① 座標変換 $\text{Pol}(x, y)$ 、 $\text{Rec}(r, \theta)$
- ② 後置関数 x^2 、 x^{-1} 、 $x!$ 、 $^\circ$ 、 $^\circ$ 、 $^\circ$ 、 $^\circ$
- ③ べき乗・べき乗根 x^y 、 $\sqrt[y]{x}$
- ④ π やメモリの直前の省略乗算 2π 、 $4R$ 等
- ⑤ 前置関数 $\sqrt{\quad}$ 、 $\sqrt[3]{\quad}$ 、 \log 、 10^x 、 \ln 、 e^x 、 \sin 、 \cos 、 \tan 、 \sin^{-1} 、 \cos^{-1} 、 \tan^{-1} 、 \sinh 、 \cosh 、 \tanh 、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} 、 $(-)$ 、 Abs 、 Int 、 Frac 、 h 、 d 、 b 、 o 、 Neg 、 Not
※ \sinh 、 \cosh 、 \tanh 、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} はfx-6500Gのみです。
- ⑥ 前置関数やカッコの前の省略乗算 $3\sin 5$ 、 $6\sqrt{7}$ 、 $2\sin 30\cos 60$ 等
- ⑦ \times 、 \div
- ⑧ $+$ 、 $-$
- ⑨ and
- ⑩ or、xor
- ⑪ 関係演算子 $<$ 、 $>$ 、 $=$ 、 \neq 、 \leq 、 \geq

※同順位の関数が連続しているときは右側から左側へ $(e^{\ln\sqrt{120}} \rightarrow e^x \{\ln(\sqrt{120})\})$ 、
他は左側から右側へ実行されます。

※複合関数は右側から左側へ $(\sin \cos^{-1} 0.6 \rightarrow \sin(\cos^{-1} 0.6))$ 実行されます。

※カッコが使用された場合は、カッコ内が最優先されます。

例 $2+3 \times (\log \sin 2\pi^{\text{rad}} + 6.8) = 22.07101691$



■スタック数

本機には優先順位の低い計算数値や計算命令(関数等)を一時的に記憶するスタックと呼ばれるメモリーがあります。

数値用のスタックは8段、命令用のスタックは20段まであります。このスタック以上に複雑な計算式を実行しますとスタックエラー(Stk ERROR)となります。

例 スタックの数え方

$$2 \times ((3 + 4 \times (5 + 4) \div 3) \div 5) + 8 =$$

① ② ③ ④ ⑤

1 2 3 4 5 6 7

数値用スタック

①	2
②	3
③	4
④	5
⑤	4
⋮	

命令用スタック

1	×
2	(
3	(
4	+
5	×
6	(
7	+
⋮	

※計算は優先順位の高い順に実行され、スタックの中から消されます。

■計算モードについて

本機はマニュアル計算を行ったりプログラムを記憶させたりするモードや、一般計算、統計計算を行なうモードなど、いろいろなモードがあります。このモードは計算に合わせて切替えてください。

●動作モード

動作モードは全部で3つあります。

①RUN(ラン)モード

マニュアル計算やプログラムの実行、グラフ表示を行なうモードで、関数計算を含めて計算はこのモードで行ないます。

②WRT(ライト)モード

プログラムを記憶させたり、編集を行なうモードです。(第4章参照)

③PCL(プログラムクリアー)モード

記憶させたプログラムを消去するモードです。(第4章参照)

●演算モード

演算モードは全部で6つあり、計算の種類により使いわけます。

①Compモード

関数計算を含む一般の計算を行ないます。

②Base-nモード

2進・8進・10進・16進の変換および計算、論理演算を行なうモードです。(44ページ参照) 関数計算およびグラフ表示は行なえません。

③SD1モード

標準偏差計算(1変数統計)を行なうモードです。(48ページ参照)

④SD2モード

1変数統計データによる棒グラフ、折れ線グラフ、正規分布曲線を描くモードです。(79ページ参照)

⑤LR1モード

回帰計算(2変数統計)を行なうモードです。(50ページ参照)

⑥LR2モード

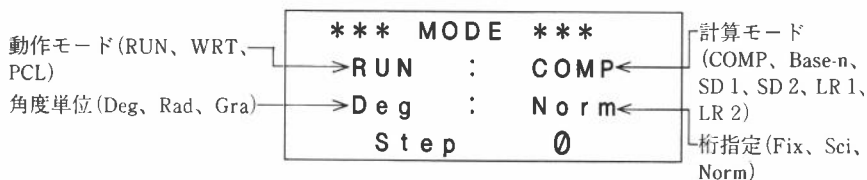
2変数統計データによる回帰直線を描くモードです。(83ページ参照)

以上のようなモードがありますので、モードを確認しながら計算を始めてください。

〈注 意〉

電源スイッチを OFF にしたとき(オートパワーオフを含む)、動作モードは解除され、次の電源スイッチ ON で RUN モードに設定されますが、演算モードおよび小数点以下指定(MODE 7 n)、有効桁数指定(MODE 8 n)、角度単位の指定(Deg、Rad、Gra)は記憶されています。

電源スイッチを ON にすると、設定されているモードの状態が表示されますので、使いたいモードであるか確認して、計算を行なってください。



■入出力桁数と演算桁数

本機の入出力できる範囲(入出力桁数)は仮数部10桁、指数部2桁ですが、内部では仮数部13桁、指数部2桁で計算を行なっています。

例 $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE]

4 2 8 5 7 . 1 4 2 8 6
0 . 1 4 2 8 5 7 1 4

3 [EXP] 5 [÷] 7 [÷] 42857 [EXE]

計算結果が 10^{10} 以上および 10^{-2} 未満のときは、自動的に指数表示になります。

例 $123456789 \times 9638 =$

123456789 [×] 9638 [EXE]

1 . 1 8 9 8 7 6 5 3 2 E + 1 2

↑
仮数部

↑
指数部

なお、一度計算を終らせて表示させますと仮数部10桁に丸められますので、表示数値に続けて計算させた場合は仮数部10桁が計算数値となります。

例 $3 \times 10^5 \div 7 =$

3 [EXP] 5 [÷] 7 [EXE]

4 2 8 5 7 . 1 4 2 8 6
0 . 1 4 2 8 6

[÷] 42857 [EXE]

※メモリー内にも仮数部13桁、指数部2桁が記憶できます。

■桁オーバーとエラーについて

計算機が計算範囲をこえて使用されたり、誤った入力を行いますと表示窓に“○○ ERROR”とエラーメッセージを表示して、以後の計算ができなくなります。

これが〈エラーチェック機能〉で、次の場合にそうなります。

(1)計算途中または答え、もしくはメモリー内の数値が $\pm 9.999999999 \times 10^{99}$ をこえたとき。

(2)関数計算において、200ページの被演算数の範囲をこえて計算しようとしたとき。

(3)統計計算で、適当でない操作が行なわれたとき。

(例 $n=0$ で x や $x \sigma n$ を求めようとしたとき)

(4)数値用スタックや演算用スタックをこえて計算しようとしたとき。

(例 □ キーを続けて19回押し、2田3田4の計算をしたとき)

(5)メモリーを増設していないのに、Z[2]のようなメモリーを使ったとき。

(メモリーの増設については24ページを参照してください)

(6)書式上誤った入力をして [EXE] キーを押したとき。

(例 5田田3 [EXE] と操作したとき)

(7)引数を必要とする命令や関数で、引数の値が適当でないとき。

(例 Sci、Fix、で0～9以外の値を入力したとき)

以上のようなときに次のようなエラーメッセージを表示します。

(1)～(3)のときは“Ma ERROR”

(4)のときは“Stk ERROR”

(5)のときは“Mem ERROR”

(6)のときは“Syn ERROR”

(7)のときは“Arg ERROR”

エラーメッセージにはこの他に、“Ne ERROR”(ネスティングエラー)と“Go ERROR”(ゴーエラー)がありますが、これらのエラーは主にプログラムを使用中におきますので、98ページまたは188ページの「エラーメッセージ一覧表」をご覧ください。

■入力文字数

本機は計算を行なうために127ステップのエリアがあります。これは一度に計算できる数値や計算命令を覚えておくエリアです。

この127ステップとは1機能1ステップで数え、数字や \oplus 、 \ominus 、 \otimes 、 \oplus キーなどは1つのキー操作で1ステップとなりますが、 $\boxed{\text{SHIFT}} \boxed{\sin^{-1}}$ のように2つのキーを操作しても、機能的には1機能ですので1ステップと数えます。

このステップの確認はカーソルでできます。 $\boxed{\leftarrow}$ キーや $\boxed{\rightarrow}$ キーを押すことによりカーソルが1ステップずつ移動します。

入力文字は127ステップまでしか入力できませんので、通常カーソルは“—”の点滅となっていますが、122ステップ目の入力をするるとカーソルが“—”から“■”の点滅に変わります。もし計算をしていてカーソルが“■”になったときは区切りの良いところで一度計算を終らせ、続けて計算を始めてください。

※数値や計算命令を入力していきますと、表示窓の左から入力されますが、計算の答えは右づめで表示されます。

■グラフィック表示とテキスト表示

本機はグラフを表示するグラフィック表示と計算式や命令を表示するテキスト表示があり、各々は独立して記憶されます。

このグラフィック表示とテキスト表示の切り替えは $\boxed{\text{G}\leftrightarrow\text{T}}$ キーを押すことにより行なえます。 $\boxed{\text{G}\leftrightarrow\text{T}}$ キーを押すごとに“グラフィック→テキスト→グラフィック→……”と切り替わります。

各々の表示をクリアーする操作も異なります。

グラフィック表示 : $\boxed{\text{SHIFT}} \boxed{\text{Cls}} \boxed{\text{EXE}}$

テキスト表示 : $\boxed{\text{AC}}$

なお、 $\boxed{\text{AC}}$ キーを押すと、グラフィック表示のときでも、表示がクリアーされたテキスト表示に切り替わります。

■表示用レジスタについて

本機はテキスト表示画面とグラフィック表示画面を記憶しておく表示用レジスタを別々に持っています。

この2つのレジスタは各々に関係のある操作(計算や **AC**) を押したときのテキスト表示やグラフを描いたり **SHIFT** **Cls** **EXE** と押し **G→T** キーを押したときのグラフィック表示)以外では変更されません。特に、テキストモードでは2進・8進・10進・16進数変換や、小数点以下・有効桁数指定のときなどに便利のように、前回の計算結果を記憶しておき、演算そのものに影響を与えない命令が実行されたときに前回の計算結果を表示します。

前回の計算結果を表示する命令

- Lbl○
- Deg
- Prog○
- Dsz○
- Rad
- Isz○
- Gra
- Mcl
- Fix○
- Hex
- Sci○
- Dec
- Norm
- Bin
- Rnd
- Oct
- Scl

※以上の命令を **EXE** で実行したり、▲を付けて停止したときに表示します。

例 123×456の計算後、グラフィック表示をクリアーする。

※グラフィック表示のClsは計算には関係ないため、前回の計算結果を表示します。

AC 123 **×** 456 **EXE**

SHIFT **Cls** **EXE**

123×456	56088.
123×456	56088.
Cls	56088.

このような計算結果の表示は**AC**を押すか、一度電源がオフになると(オートパワーオフを含む)、0になります。

■訂正について

- 計算式を入力しているときに押しまちがいに気づいたときは \leftarrow 、 \rightarrow キーを使ってまちがった箇所カーソルを合わせ、正しいキーを押します。

例 123+456=の123を122と押してしまった。

① ② ②	1 2 2 _
\leftarrow	1 2 2
③	1 2 3 _

例 sin 60 を cos 60 と押してしまった。

cos ⑥ ①	c o s 6 0 _
\leftarrow \leftarrow \leftarrow	c o s 6 0
sin	s i n 6 0

※このように、いくつか前の訂正後、入力終了していればそのまま EXE キーを押して答を求めますが、さらに計算が続く場合には \rightarrow キーを押して入力している計算式の次にカーソルを進め、入力が続けます。

- まちがって不要なキーを押してしまったときには \leftarrow 、 \rightarrow キーで不要な入力の箇所カーソルを合わせて DEL キーを押します。 DEL キーは1回押すごとに1命令(1ステップ)ずつ削除されます。

例 369×2を369××2と押してしまった。

③ ⑥ ⑨ \times \times ②	3 6 9 \times \times 2 _
\leftarrow \leftarrow DEL	3 6 9 \times 2

入力した計算式の途中で挿入をしたいときには \leftarrow 、 \rightarrow キーで挿入したい箇所の次にカーソルを合わせて SHIFT INS と押します。 SHIFT INS と押すと、挿入状態となり、続けて挿入ができます。

例 2.36²を sin 2.36²としたい。

② \cdot ③ ⑥ x^2	2 . 3 6 ² _
\leftarrow \leftarrow \leftarrow \leftarrow \leftarrow	2 . 3 6 ²
SHIFT INS	2 . 3 6 ²
sin	s i n 2 . 3 6 ²

※ SHIFT INS と押しますと、挿入位置の文字と“ I ”が交互に点滅します。この“ I ”の部分に文字や命令が挿入されていき、 \leftarrow \rightarrow \uparrow \downarrow や AC キーが押されるまで挿入状態となります。なお、“ I ”の点滅は ALPHA キーが押されたアルファモードでは“ L ”に、 SHIFT キーが押されたシフトインモードでは“ I ”となります。

■メモリーについて

本機で使えるメモリーは標準で26個あります。このメモリーはアルファベットのAからZまでの名前で呼ばれ、仮数部13桁、指数部2桁までの数値を記憶します。

例 Aメモリーに123.45を記憶する。

123.45 \rightarrow ALPHA A

123.45 \rightarrow A__
123.45

EXE

メモリーへの代入には \rightarrow キーを使い、メモリー名を右側に書きます。

例 BメモリーにAメモリー+78.9を記憶する。

ALPHA A + 78.9 \rightarrow ALPHA B

A+78.9 \rightarrow B__
202.35

EXE

例 Bメモリーに74.12を加える。

ALPHA B + 74.12 \rightarrow ALPHA B

B+74.12 \rightarrow B__
276.47

EXE

●メモリー内容の確認は、メモリー名を押してから [EXE] キーを押します。

ALPHA A EXE

123.45

●メモリー内容をクリアー(0にする)するには、次のように操作します。

例 Aメモリーだけをクリアーする。

0 \rightarrow ALPHA A EXE

0.

例 全メモリーをクリアーする。

SHIFT MCl

M c l __
0.

EXE

直前に表示されていた内容を表示します

●複数のメモリーに同じ数値を記憶させるには [SHIFT] [~] を使います。

例 AからJまでの10個のメモリーに10を記憶する、

10 \rightarrow ALPHA A SHIFT ~ ALPHA J

10 \rightarrow A~J__
10.

EXE

■メモリーの増設

メモリーは標準状態で26個ありますが、プログラムを記憶するステップと呼ばれるエリアをメモリーに変換して増やすことができます。このメモリーの増設は、1メモリーにつき8ステップの換算で行なわれます。

※プログラムステップ数については101ページをご覧ください。

メモリー数	26	27	28	36	76	86
ステップ数	486	478	470	402	86	6

メモリーの増設は1個単位で、最大60個増やして86個にすることができます。

メモリーの増設は **MODE** □ **増設個数** **EXE** と操作します。

例 メモリーを30個増設して56個とします。

MODE □ 30

Defm 30

EXE

```

**Defm**
Program : 0
Memory : 56
246 Bytes Free
    
```

使用済みのプログラムステップ数
メモリー数
現在の残りステップ数

使用済みのプログラムステップ数とメモリー数と残りステップ数が表示されますが、残りステップ数は現在未使用のエリアを表示しますので、プログラムを記憶させている量により変わります。

現在メモリー数がいくつになっているかを確認するときは、**MODE** □ **EXE** と操作します。

MODE □ **EXE**

```

**Defm**
Program : 0
Memory : 56
246 Bytes Free
    
```

また、メモリー数を初期の26個に戻すには増設個数0を入力します。

MODE □ 0 **EXE**

```

**Defm**
Program : 0
Memory : 26
486 Bytes Free
    
```

※増設できるメモリー数は最大60個ですが、すでにプログラムが記憶され、増設したい数ほどステップ数が残っていない場合はエラーとなり増設できません。

※この **MODE** \square 個数はプログラムとして記憶させることもできます。

●増設メモリーの使い方

増設されたメモリーはZ[1]、Z[2]、……として使われます。使い方は他のメモリーと同じですが、呼び名としてZの後に[,]で順番を示す数字とともに使われます。([と])は **ALPHA** \square 、**ALPHA** **EXP** と押す)

たとえば5個のメモリーを増設したときはZ[1]~Z[5]が使えます。

このような使い方は配列を使うときと同じで、名前の他に添字と呼ばれる番号をつけて使います。この配列としての使い方については、プログラム編の119ページをご覧ください。

■アンサー (Ans) 機能

本機には直前の計算結果を記憶しておくアンサー機能があります。この機能は数値や数式等を入力後 **EXE** キーを押した結果(数式であれば答え)を記憶します。そして呼び出しは **Ans** キーで行ないます。

Ans キーを押しますと“Ans”と表示され、そのまま計算式に使うことができます。

※Ans はメモリーと同じ扱いになりますので、以後 Ans メモリーと呼びます。

例 $123+456=579$

$789-579=210$

1 **2** **3** **+** **4** **5** **6** **EXE**

123+456

579.

7 **8** **9** **-** **Ans** **EXE**

789-Ans

210.

Ans メモリーには仮数部13桁、指数部2桁を記憶でき、**AC** キーを押しても電源をオフにしても消えません。新たに **EXE** キーが押され、演算が実行されたときに新たな数値が記憶されます。

なお、他のメモリーへの記憶操作では **EXE** キーを押しても Ans メモリーに記憶されません。

例 78+56=134の計算後、123をAメモリーに記憶させる。

7 8 + 5 6 EXE

Ans EXE ...Ansメモリーの
確認

1 2 3 → ALPHA A

EXE

Ans EXE

78+56	134.
Ans	134.
123→A	123.
Ans	134.

Ansメモリーは他のメモリーと同様な使い方ができますので、計算式中に用いたり、“Ans”の直前の“×”を省略できます。

例 15×3=45

78×45-23=3487

1 5 × 3 EXE

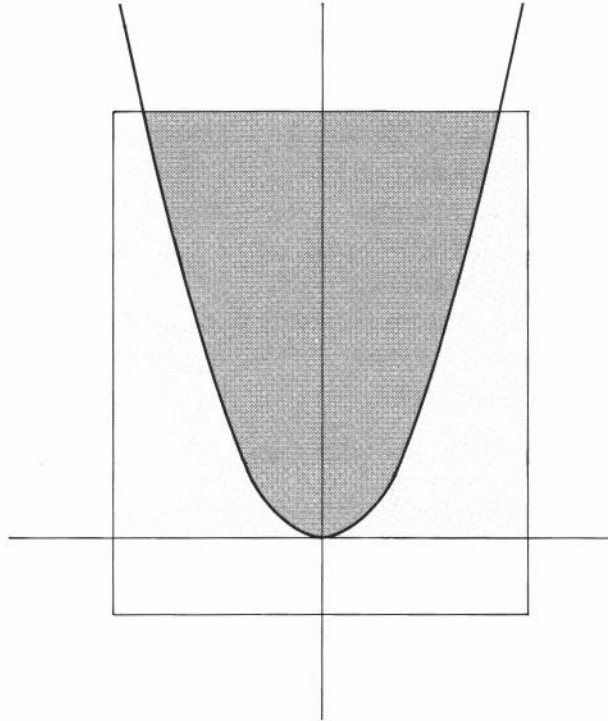
7 8 Ans = 2 3 EXE

15×3	45.
78Ans-23	3487.

■オートパワーオフ(自動電源OFF)機能について

スイッチの切り忘れによるムダな電力消費を防ぐ自動節電機能で、操作完了後(プログラム計算中を除く)や表示・入力待ちの状態ですら約6分後、自動的に電源オフになります。この場合は、電源スイッチを入れ直すか、**AC**キーを押せば、再び電源オンになります。(電源オフになってもメモリー内の数値およびプログラム、計算モードは変わりません)

第2章 マニュアル計算



2-1 基本計算の仕方

※ここからは、表示は答のみ記します。

■加減乗除計算

- 加減乗除計算は、数式どおりにキーを押します。
- 負数は数値の前に $[-]$ キーを押します。

例 題	操 作	表 示
$23+4.5-53=-25.5$	23 $+$ 4.5 $-$ 53 $[\text{EXE}]$	-25.5
$56 \times (-12) \div (-2.5) = 268.8$	56 \times $[-]$ 12 $+$ $[-]$ 2.5 $[\text{EXE}]$	268.8
$12369 \times 7532 \times 74103 = 6.903680613 \times 10^{12}$ (6903680613000)	12369 \times 7532 \times 74103 $[\text{EXE}]$	6.903680613 $_{\text{E}}$ +12
※答えが 10^{10} (100億)以上および 10^{-2} (0.01)未満のときは指数表示になります。		
$(4.5 \times 10^{75}) \times (-2.3 \times 10^{-79}) = -1.035 \times 10^{-3}$ (-0.001035)	4.5 $[\text{EXP}]$ 75 \times $[-]$ 2.3 $[\text{EXP}]$ $[-]$ 79 $[\text{EXE}]$	-1.035 $_{\text{E}}$ -03
$(1 \times 10^5) \div 7 = 14285.71429$	1 $[\text{EXP}]$ 5 \div 7 $[\text{EXE}]$	14285.71429
$(1 \times 10^5) \div 7 - 14285 = 0.7142857$	1 $[\text{EXP}]$ 5 \div 7 $-$ 14285 $[\text{EXE}]$	0.71428571
※内部演算は仮数部13桁で計算し、答えは仮数部11桁目を四捨五入して10桁まで表示します。		

- 四則の混合計算は、加減より乗除優先で計算されます。

例 題	操 作	表 示
$3+5 \times 6=33$	3 $+$ 5 \times 6 $[\text{EXE}]$	33.
$7 \times 8 - 4 \times 5=36$	7 \times 8 $-$ 4 \times 5 $[\text{EXE}]$	36.
$1+2-3 \times 4 \div 5+6=6.6$	1 $+$ 2 $-$ 3 \times 4 \div 5 $+$ 6 $[\text{EXE}]$	6.6

■カッコ計算

例 題	操 作	表 示
$100 - (2 + 3) \times 4 = 80$	100 \square (\square 2 \square + \square 3 \square) \square \times 4 \square [EXE]	80.
$2 + 3 \times (4 + 5) = 29$ ※ [EXE] を押す直前の閉じカッコは何重であってもキー操作をはぶくことができます。	2 \square + 3 \square \times (\square 4 \square + \square 5 \square) [EXE]	29.
$(7 - 2) \times (8 + 5) = 65$ ※ カッコの直前の \square は省略できます。	(\square 7 \square - \square 2 \square) \square (\square 8 \square + \square 5 \square) [EXE]	65.
$10 - \{2 + 7 \times (3 + 6)\} = -55$ ※ 以下、本書では省略せずに記します。	10 \square - { \square 2 \square + 7 \square (\square 3 \square + \square 6 \square) [EXE]	-55.
$\frac{2 \times 3 + 4}{5} = (2 \times 3 + 4) \div 5 = 2$	(\square 2 \square \times 3 \square + \square 4 \square) \square \div 5 [EXE]	2.
$\frac{5 \times 6 + 6 \times 8}{15 \times 4 + 12 \times 3} = 0.8125$	(\square 5 \square \times 6 \square + \square 6 \square \times 8 \square) \square \div (\square 15 \square \times 4 \square + \square 12 \square \times 3 \square) [EXE]	0.8125
$(1.2 \times 10^{19}) - \{(2.5 \times 10^{20}) \times \frac{3}{100}\} = 4.5 \times 10^{18}$	1.2 [EXP] 19 \square - { (\square 2.5 [EXP] 20 \square \times 3 \square \div 100 \square) [EXE]	4.5 E+18
$\frac{6}{4 \times 5} = 0.3$ ※ 上例は、6 \square \div 4 \square \div 5 [EXE] としても同様です。	6 \square \div (\square 4 \square \times 5 \square) [EXE]	0.3

■メモリー計算

- 各メモリー内の数値は電源スイッチを切っても消えませんが、**SHIFT** **MCl** **EXE**と操作すると全て消えます。

例 題	操 作	表 示
9.874×7=69.118	9.874 → ALPHA A EXE	9.874
9.874×12=118.488	ALPHA A × 7 EXE	69.118
9.874×26=256.724	ALPHA A × 12 EXE	118.488
9.874×29=286.346	ALPHA A × 26 EXE	256.724
	ALPHA A × 29 EXE	286.346
<p>※メモリーに数値を入れておくには → キーを使用します。(従ってあらかじめメモリーをクリアしておく必要はありません)</p>		
23+9=32	23 + 9 → ALPHA B EXE	32.
53-6=47	53 - 6 EXE	47.
-)45×2=90	ALPHA B + Ans → ALPHA B	
99÷3=33	EXE	79.
合計 22	45 × 2 EXE	90.
	ALPHA B - Ans → ALPHA B	
	EXE	-11.
	99 + 3 EXE	33.
	ALPHA B + Ans → ALPHA B	
	EXE	22.
12×(2.3+3.4)-5=63.4	2.3 + 3.4 → ALPHA G EXE	5.7
30×(2.3+3.4+4.5)-15	12 × ALPHA G - 5 EXE	63.4
×4.5=238.5	4.5 → ALPHA H EXE	4.5
	30 × ↑ ALPHA G + ALPHA H	
	↓ - 15 ALPHA H EXE	238.5
<p>※メモリー名の直前の × キーは省略できます。</p>		

- ★メモリー計算で使うアルファベットは、**ALPHA** キーに続けて、各々のアルファベットが右下に書かれているキーを押します。(5ページ参照)

■小数点以下指定、有効桁数指定計算および指数表示指定

- 小数点以下指定(Fix)は **MODE** **7** n (n は 0 ~ 9) **EXE** と押すことにより小数点以下の桁数を指定します。
- 有効桁数指定(Sci)は **MODE** **8** n (n は 0 ~ 9、ただし 0 は10桁指定) **EXE** と押すことにより有効桁数を指定します。
- 指数表示指定は **ENG** または **SHIFT** **ENG** と押すことにより、表示されている数値を指数部が 3 の倍数となる指数表示に変換します。
- 小数点以下・有効桁数指定を行なったときは、他の指定または **MODE** **9** **EXE** と押すまで解除されません。(電源オフでもモード切り換えでも解除されません)
- 小数点以下・有効桁数指定を行なっても、内部演算は仮数部13桁を、表示数値は仮数部10桁を記憶しています。この数値を指定された桁と同じにするには **SHIFT** **Rnd** **EXE** と操作します。

例 題	操 作	表 示
100÷6=16.66666666...	100 ÷ 6 EXE	16.66666667
	MODE 7 4 EXE (小数点以下4桁指定)	16.6667
	MODE 9 EXE (指定解除)	16.66666667
	MODE 8 5 EXE (有効桁数5桁指定)	1.6667E+01
	MODE 9 EXE (指定解除)	16.66666667
※指定を行なうと、指定桁の下1桁目が四捨五入されて表示されます。		

$200 \div 7 \times 14 = 400$	MODE 7 3 EXE (小数点以下 3桁指定)	16. 667
(表示10桁で計算を続ける)	$200 \div 7$ EXE	28. 571
	☒	28. 57142857 X _
	14 EXE	400. 000
	同じ計算を指定桁で計算する と	
	$200 \div 7$ EXE	28. 571
	(内部切捨て) SHIFT Rnd EXE	28. 571
	☒	28. 571 X _
	14 EXE	399. 994
	MODE 9 EXE (指定解除)	399. 994
$123\text{m} \times 456 = 56088\text{m}$ $= 56.088\text{km}$	123 ☒ 456 EXE	56088.
	ENG	56. 088 E +03
$78\text{g} \times 0.96 = 74.88\text{g}$ $= 0.07488\text{kg}$	78 ☒ 0.96 EXE	74. 88
	SHIFT ENG	0. 07488 E +03

2-2 特別機能

■連続演算機能

EXE キーにより演算を終了させた結果に対して、さらに計算を続けることができます。この場合は表示桁の仮数部10桁で計算が行なわれます。

例 $3 \times 4 = 12$ 続けて $\div 3.14 =$

3 \times 4 **EXE**

(続けて) \div 3.14 **EXE**

3×4	12.
12.÷3.14	3.821656051

例 $1 \div 3 \times 3 =$ の計算を行なう。

1 \div 3 \times 3 **EXE**

1 \div 3 **EXE**

(続けて) \times 3 **EXE**

1÷3×3	1.
1÷3	0.3333333333
0.3333333333×3	0.9999999999

この機能はメモリーへの記憶 (\rightarrow) および後置関数 (x^2 、 x^{-1} 、 $x!$: 42ページ参照)、 $+$ 、 $-$ 、 x^y 、 $\sqrt{\quad}$ 、 $\circ\circ$ についても使えます。

例 $12 \times 45 = 540$ の結果を C メモリーに記憶する。

12 \times 45 **EXE**

(続けて) \rightarrow **ALPHA** **C**

EXE

12×45	540.
540.→C	540.

例 78÷6=13の結果を2乗する。(42ページ参照)

78 ÷ 6 [EXE]

78 ÷ 6	13.
13. ²	169.

(続けて) [x²] [EXE]

■リプレイ機能

●計算式を記憶する機能で、演算終了後 [↔] キーまたは [⇐] キーを押すことにより実行した式を表示します。

[↔] キーを押しますと式を表示し、カーソルを先頭に位置します。

[⇐] キーを押しますと式を表示し、カーソルを最後の次に位置します。

以後、[→] キーや [⇐] キー、[↑] キーや [↓] キーによりカーソルを進め、式をチェックしたり、数値や計算命令を変更して計算を行なうことができます。

例

123 × 456 [EXE]

123 × 456	56088.
-----------	--------

[↔]

<u>1</u> 23 × 456	
-------------------	--

※一度表示をクリアしてから表示します。

[EXE]

123 × 456	56088.
-----------	--------

[↔]

123 × 456	_
-----------	---

例 4.12 × 3.58 + 6.4 = 21.1496

4.12 × 3.58 - 7.1 = 7.6496

4.12 × 3.58 + 6.4 [EXE]

4.12 × 3.58 + 6.4	21.1496
-------------------	---------

[↔]

4.12 × 3.58 + 6.4	_
-------------------	---

[↔] [↔] [↔] [↔]

4.12 × 3.58 + 6.4	
-------------------	--

[=] 7.1 [EXE]

4.12 × 3.58 - 7.1	7.6496
-------------------	--------

- 演算実行時にエラーが生じた場合、**[AC]** キーを押せば全てクリアーされ、最初から計算式を入力することができますが、**[⇐]** キーまたは **[⇒]** キーを押すことによりエラー状態が解除され、エラーが生じた箇所にカーソルを位置させ、表示します。

例 14÷10×2.3をまちがえて14÷0×2.3と入力してしまった。

14 **[⇐]** 0 **[⇐]** 2.3 **[EXE]**

14÷0×2.3
Ma ERROR
Step 4

[⇐] (または **[⇒]**)

14÷0×2.3

↑ ここでエラーが発生しました

[⇐] **[SHIFT]** **[INS]** 1 **[EXE]**

14÷10×2.3
3.22

※リプレイ機能により記憶される文字数は入力文字数(20ページ参照)と同じ127ステップです。

※リプレイ機能は **[AC]** キーを押したり、電源オフ、モード切り換えによりクリアーされます。

■マルチステートメント機能

- プログラム計算で使われるマルチステートメント機能(“:”コロンで式および文を区切る)をマニュアル計算でも行なうことができます。
- このマルチステートメント機能とは、“:”により式を区切り、続けて複数の計算を行なわせます。
- マルチステートメントで入力された式は **[EXE]** キーを押すことにより先頭から順に実行されます。
- マルチステートメントで続けて実行される途中の結果を見たいときは“:”のかわりに“▲” (**[SHIFT]** **[▲]**)を入力します。

例 $6.9 \times 123 = 848.7$
 $123 \div 3.2 = 38.4375$

123 **[→]** **[ALPHA]** **[A]** **[.]** 6.9 **[×]**
[ALPHA] **[A]** **[SHIFT]** **[▲]**
[ALPHA] **[A]** **[÷]** 3.2 **[EXE]**

```

123→A:6.9×A▲
A÷3.2
                        848.7
                    - Disp -
    
```

↑“▲”により表示中点灯

[EXE]

```

123→A:6.9×A▲
A÷3.2
                        848.7
                    38.4375
    
```

※計算式の最後には“▲”を入力しなくても、結果が表示されます。

※マルチステートメント文での連続演算はできません。

$123 \times 456 : +5$
 不可

2-3 関数計算の仕方

■角度単位変換、角度計算

- 角度単位は **MODE** [4] ~ [6] **EXE** と操作することにより基本となる角度単位をそれぞれ度・ラジアン・グラッドに設定します。
- **SHIFT** **MODE** [4] ~ [6] と操作することにより、直前の数値を度・ラジアン・グラッドに指定します。
- 基本となる角度単位を一度設定しますと、他の設定を行なわない限り、電源オフでも変更されません。
- 角度単位の確認は **Disp** を押すことにより行なえます。

例 題	操 作	表 示
4.25rad を度に変換	MODE [4] EXE 4.25 SHIFT MODE [5] EXE	243.5070629
1.23gra をラジアンに変換	MODE [5] EXE 1.23 SHIFT MODE [6] EXE	0.01932079482
7.89° をグラッドに変換	MODE [6] EXE 7.89 SHIFT MODE [4] EXE	8.766666667
答えは度で求める 47.3°+82.5rad=	MODE [4] EXE 47.3 + 82.5 SHIFT MODE [5] EXE	4774.20181
12.4°+8.3rad-1.8gra=	12.4 + 8.3 SHIFT MODE [5] = 1.8 SHIFT MODE [6] EXE	486.33497
答えはラジアンで求める 24°6'31"+85.34rad=	24 ° ' " 6 ° ' " 31 ° ' " SHIFT MODE [4] + 85.34 EXE	85.76077464
答えはグラッドで求める 36.9°+41.2rad=	MODE [6] EXE 36.9 SHIFT MODE [4] + 41.2 SHIFT MODE [5] EXE	2663.873462

■三角関数(sin、cos、tan)、逆三角関数(\sin^{-1} 、 \cos^{-1} 、 \tan^{-1})

●三角・逆三角関数の計算を行なうときは、角度単位を確実に設定してください。

例 題	操 作	表 示
$\sin 63^{\circ} 52' 41'' = 0.897859012$	MODE 4 EXE sin 63 52 41 EXE	0.897859012
$\cos\left(\frac{\pi}{3}\text{rad}\right) = 0.5$	MODE 5 EXE cos (SHIFT π \div 3) EXE	0.5
$\tan(-35\text{gra}) =$ -0.6128007881	MODE 6 EXE tan (-) 35 EXE	-0.6128007881
$2 \cdot \sin 45^{\circ} \times \cos 65^{\circ} =$ 0.5976724775	MODE 4 EXE 2 \times sin 45 \times cos 65 EXE ↑ 省略可	0.5976724775
$\sin^{-1} 0.5 = 30^{\circ}$ ($\sin x = 0.5$ の x を求める)	SHIFT \sin^{-1} 0.5 EXE ↑ .5でも可	30.
$\cos^{-1} \frac{\sqrt{2}}{2} =$ 0.7853981634 rad $= \frac{\pi}{4} \text{ rad}$	MODE 5 EXE SHIFT \cos^{-1} ($\sqrt{\quad}$ 2 \div 2) EXE \div SHIFT π EXE	0.7853981634 0.25
$\tan^{-1} 0.741 = 36.53844577^{\circ}$ $= 36^{\circ} 32' 18.4''$	MODE 4 EXE SHIFT \tan^{-1} 0.741 EXE SHIFT 0.00	36.53844577 36° 32' 18.4''
<p>※度・分・秒の各桁の合計が11桁をこえた場合は上位(度および分)を優先して表示し、それ以下は表示されませんが計算機内には10進数として全部残っています。</p>		
$2.5 \times (\sin^{-1} 0.8 - \cos^{-1} 0.9)$ $= 68^{\circ} 13' 13.53''$	2.5 \times (SHIFT \sin^{-1} 0.8 - SHIFT \cos^{-1} 0.9) EXE SHIFT 0.00	68° 13' 13.53''
$\sin 18^{\circ} \times \cos 0.25 \text{ rad} =$ 0.2994104044	sin 18 \times cos 0.25 SHIFT MODE 5 EXE	0.2994104044
<p>※上例はラジアン単位で、sin 18 SHIFT MODE 4 \times cos 0.25 EXE と操作しても同様です。</p>		

■対数関数(log、ln)、指数関数(10^x 、 e^x 、 x^y 、 $\sqrt[x]{\quad}$)

例 題	操 作	表 示
$\log 1.23(\log_{10} 1.23)=$ 0.08990511144	$\boxed{\log} 1.23 \boxed{\text{EXE}}$	0.08990511144
$\ln 90(=\log_e 90)=$ 4.49980967	$\boxed{\ln} 90 \boxed{\text{EXE}}$	4.49980967
$\log 456 \div \ln 456=$ 0.4342944819 (log/ln 比=定数 M)	$\boxed{\log} 456 \boxed{\div} \boxed{\ln} 456 \boxed{\text{EXE}}$	0.4342944819
$10^{1.23}=16.98243652$ (常用対数1.23の真数を求める)	$\boxed{\text{SHIFT}} \boxed{10^x} 1.23 \boxed{\text{EXE}}$	16.98243652
$e^{4.5}=90.0171313$ (自然対数4.5の真数を求める)	$\boxed{\text{SHIFT}} \boxed{e^x} 4.5 \boxed{\text{EXE}}$	90.0171313
$10^4 \cdot e^{-4} + 1.2 \cdot 10^{2.3}=$ 422.5878667	$\boxed{\text{SHIFT}} \boxed{10^x} 4 \boxed{\times} \boxed{\text{SHIFT}} \boxed{e^x}$ $\boxed{(-)} 4 \boxed{+} 1.2 \boxed{\times} \boxed{\text{SHIFT}} \boxed{10^x}$ 2.3 $\boxed{\text{EXE}}$	422.5878667
$5.6^{2.3}=52.58143837$	5.6 $\boxed{x^y}$ 2.3 $\boxed{\text{EXE}}$	52.58143837
$\sqrt[7]{123}(=123^{\frac{1}{7}})=$ 1.988647795	7 $\boxed{\sqrt[x]{\quad}}$ 123 $\boxed{\text{EXE}}$	1.988647795
$(78-23)^{-12}=$ 1.305111829 $\times 10^{-21}$	$\boxed{[]} 78 \boxed{[-]} 23 \boxed{[]} \boxed{x^y} \boxed{(-)} 12$ $\boxed{\text{EXE}}$	1.305111829 $\text{E}-21$
$2+3 \times \sqrt[3]{64}-4=10$ * x^y および $\sqrt[x]{\quad}$ は \times 、 \div より優先して計算されます。	2 $\boxed{+}$ 3 $\boxed{\times}$ 3 $\boxed{\sqrt[x]{\quad}}$ 64 $\boxed{[-]} 4 \boxed{\text{EXE}}$	10.
$2 \times 3.4^{(5+6.7)}=3306232.001$	2 $\boxed{\times}$ 3.4 $\boxed{x^y}$ $\boxed{[]} 5 \boxed{+} 6.7 \boxed{[]}$ $\boxed{\text{EXE}}$	3306232.001

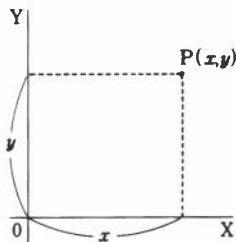
■双曲線関数(sinh、cosh、tanh)、逆双曲線関数(sinh⁻¹、cosh⁻¹、tanh⁻¹)

※fx-6500Gのみできます。

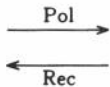
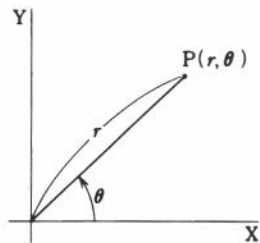
例 題	操 作	表 示
sinh 3.6=18.28545536	hyp sin 3.6 EXE	18.28545536
cosh 1.23=1.856761057	hyp cos 1.23 EXE	1.856761057
tanh 2.5=0.9866142982	hyp tan 2.5 EXE	0.9866142982
cosh 1.5 - sinh 1.5 = 0.2231301601 = e ^{-1.5} (cosh x ± sinh x = e ^{±x} の証明)	hyp cos 1.5 - hyp sin 1.5 EXE (続けて) In Ans EXE	0.2231301601 -1.5
sinh ⁻¹ 30=4.094622224	SHIFT hyp sin⁻¹ 30 EXE	4.094622224
cosh ⁻¹ ($\frac{20}{15}$) = 0.7953654612	SHIFT hyp cos⁻¹ (20 ÷ 15) EXE	0.7953654612
tanh 4x=0.88のとき x は ? $x = \frac{\tanh^{-1}0.88}{4} =$ 0.3439419141	SHIFT hyp tan⁻¹ 0.88 ÷ 4 EXE	0.3439419141
sinh ⁻¹ 2 × cosh ⁻¹ 1.5 = 1.389388923	SHIFT hyp sin⁻¹ 2 × SHIFT hyp cos⁻¹ 1.5 EXE	1.389388923
sinh ⁻¹ ($\frac{2}{3}$) + tanh ⁻¹ ($\frac{4}{5}$) = 1.723757406	SHIFT hyp sin⁻¹ (2 ÷ 3) + SHIFT hyp tan⁻¹ (4 ÷ 5) EXE	1.723757406

■座標変換(Pol、Rec)

●直交座標(Rectangular)



●極座標(Polar)



- 計算された結果はIメモリーとJメモリーに格納されます。(Iメモリーの内容は表示されます)

$$\text{Pol} \rightarrow I=r, J=\theta \quad \text{Rec} \rightarrow I=x, J=y$$

- Polで計算される θ は $-180^\circ < \theta \leq 180^\circ$ の範囲内に求められます。
(ラジアンもグラッドも同様範囲内です)

例	題	操	作	表	示
$x=14, y=20.7$ のとき	r および θ は?	MODE 4 EXE	SHIFT Pol 14 SHIFT . 20.7 EXE	24.98979792 (r)	
		EXE	(続けて) ALPHA J EXE SHIFT	55°55'42.2" (θ)	
$x=7.5, y=-10$ のとき	r および θ radは?	MODE 5 EXE	SHIFT Pol 7.5 SHIFT . (-) 10 EXE	12.5 (r)	
		EXE	(続けて) ALPHA J EXE	-0.927295218 (θ)	
$r=25, \theta=56^\circ$ のとき	x および y は?	MODE 4 EXE	SHIFT Rec 25 SHIFT . 56 EXE	13.97982259 (x)	
		EXE	(続けて) ALPHA J EXE	20.72593931 (y)	
$r=4.5, \theta=\frac{2}{3}\pi$ radのとき	x および y は?	MODE 5 EXE	SHIFT Rec 4.5 SHIFT . 2 + 3 SHIFT π EXE	-2.25 (x)	
		EXE	(続けて) ALPHA J EXE	3.897114317 (y)	

■その他の関数($\sqrt{\quad}$ 、 x^2 、 x^{-1} 、 $x!$ 、 $\sqrt[3]{\quad}$ 、Ran#、Abs、Int、Frac)

※ $x!$ はfx-6500Gでは $\boxed{\text{SHIFT}} \boxed{x!}$ キーに続けて $\boxed{x!}$ キーを押し、fx-6000Gでは直接 $\boxed{x!}$ キーを押します。

例 題	操 作	表 示
$\sqrt{2} + \sqrt{5} = 3.65028154$	$\boxed{\sqrt{\quad}} \boxed{2} \boxed{+} \boxed{\sqrt{\quad}} \boxed{5} \boxed{\text{EXE}}$	3. 65028154
$2^2 + 3^2 + 4^2 + 5^2 = 54$	$2 \boxed{x^2} \boxed{+} 3 \boxed{x^2} \boxed{+} 4 \boxed{x^2} \boxed{+}$ $5 \boxed{x^2} \boxed{\text{EXE}}$	54.
$\frac{1}{\frac{1}{3} - \frac{1}{4}} = 12$	$\boxed{[]} \boxed{3} \boxed{x^{-1}} \boxed{-} 4 \boxed{x^{-1}} \boxed{[]} \boxed{x^{-1}}$ $\boxed{\text{EXE}}$	12.
$8! (=1 \times 2 \times 3 \times \dots \times 8) = 40320$	<fx-6500G> $8 \boxed{\text{SHIFT}} \boxed{x!} \boxed{\text{EXE}}$	40320.
	<fx-6000G> $8 \boxed{x!} \boxed{\text{EXE}}$	
$\sqrt[3]{36 \times 42 \times 49} = 42$	$\boxed{\text{SHIFT}} \boxed{\sqrt[3]{\quad}} \boxed{[]} \boxed{36} \boxed{\times} \boxed{42} \boxed{\times} \boxed{49}$ $\boxed{[]} \boxed{\text{EXE}}$	42.
乱数発生(0.000~0.999の疑似乱数)	$\boxed{\text{SHIFT}} \boxed{\text{Ran\#}} \boxed{\text{EXE}}$	(例) 0.792
$\sqrt{13^2 - 5^2} + \sqrt{3^2 + 4^2} = 17$	$\boxed{\sqrt{\quad}} \boxed{[]} \boxed{13} \boxed{x^2} \boxed{-} 5 \boxed{x^2} \boxed{[]}$ $\boxed{+} \boxed{\sqrt{\quad}} \boxed{[]} \boxed{3} \boxed{x^2} \boxed{+} 4 \boxed{x^2}$ $\boxed{[]} \boxed{\text{EXE}}$	17.
$\sqrt{1 - \sin^2 40^\circ} = 0.766044431$ $= \cos 40^\circ$	$\boxed{\text{MODE}} \boxed{4} \boxed{\text{EXE}}$ $\boxed{\sqrt{\quad}} \boxed{[]} \boxed{1} \boxed{-} \boxed{[]} \boxed{\sin} \boxed{40} \boxed{[]}$ $\boxed{x^2} \boxed{[]} \boxed{\text{EXE}}$	0.766044431
($\cos \theta = \sqrt{1 - \sin^2 \theta}$ の証明)	(続けて) $\boxed{\text{SHIFT}} \boxed{\cos^{-1}} \boxed{\text{Ans}}$ $\boxed{\text{EXE}}$	40.
$\frac{1}{2!} + \frac{1}{4!} + \frac{1}{6!} + \frac{1}{8!} = 0.5430803571$	<fx-6500G> $2 \boxed{\text{SHIFT}} \boxed{x!} \boxed{x^{-1}} \boxed{+} 4 \boxed{\text{SHIFT}}$ $\boxed{x!} \boxed{x^{-1}} \boxed{+} 6 \boxed{\text{SHIFT}} \boxed{x!}$ $\boxed{x^{-1}} \boxed{+} 8 \boxed{\text{SHIFT}} \boxed{x!} \boxed{x^{-1}}$ $\boxed{\text{EXE}}$	0.5430803571

	<fx-6000G>	
	2 $x!$ x^{-1} + 4 $x!$ x^{-1} +	
	6 $x!$ x^{-1} + 8 $x!$ x^{-1}	
	EXE	
$\frac{3}{4}$ の常用対数の絶対値は?	SHIFT Abs log (3 + 4)	
$ \log \frac{3}{4} = 0.1249387366$	EXE	0.1249387366
$\frac{7800}{96}$ の整数部は……81	SHIFT int (7800 + 96)	
	EXE	81.
$\frac{7800}{96}$ の小数部は……0.25	SHIFT Frac (7800 + 96)	
	EXE	0.25
2512549139÷2141	2512549139 + 2141 EXE	1173540.
は割り切れるか?	SHIFT Frac (2512549139 +	
	2141) EXE	0.999532

2-4 2進・8進・10進・16進計算の仕方

- 2進・8進・10進・16進の計算および変換、論理演算はBase-nモード(**MODE** **□**と押す)で行ないます。
- 基本となる進数(2、8、10、16)は **Bin** **EXE**、**Oct** **EXE**、**Dec** **EXE**、**Hex** **EXE**と操作して設定します。
- 個々の数値の進数を指定する場合は、数値の直前にb、o、d、h (**SHIFT** **□**、**SHIFT** **□**、**SHIFT** **□**、**SHIFT** **□**と押す)を入力します。
- Base-nモードでは一般関数計算はできません。
- Base-nモードでは整数のみが扱え、演算結果に小数を含む場合は小数部が切り捨てられます。
- 演算は8進・10進・16進数は32ビットまで、2進数は16ビットまで扱えます。

2進数	16桁まで
8進数	11桁まで
10進数	10桁まで
16進数	8桁まで
- 取り扱う数は0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、Fですが、各進数における有効数値以外を使用するときはb、o、d、hの指定をしないと実行時にエラーとなります。

有効数値	2進数	0、1
	8進数	0、1、2、3、4、5、6、7
	10進数	0、1、2、3、4、5、6、7、8、9
	16進数	0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F
- 2進・8進・16進数では、負数は2の補数をとります。
- 16進数のA、B、C、D、E、Fはアルファベット文字と区別するため、**A**、**IB**、**C**、**ID**、**IE**、**IF**と表示されます。

●計算範囲(Base-n モード時)

- 2進数 正：1111111111111111 $\geq x \geq 0$
 負：1111111111111111 $\geq x \geq 1000000000000000$
- 8進数 正：1777777777 $\geq x \geq 0$
 負：3777777777 $\geq x \geq 20000000000$
- 10進数 正：2147483647 $\geq x \geq 0$
 負：-1 $\geq x \geq -2147483648$
- 16進数 正：7FFFFFFF $\geq x \geq 0$
 負：FFFFFFFF $\geq x \geq 80000000$

■ 2進・8進・10進・16進変換

例 題	操 作	表 示
2A ₁₆ 、274 ₈ は10進数ではいくつか？	MODE <input type="checkbox"/> Dec <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> h 2A <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> o 274 <input type="checkbox"/> EXE	42. 188.
123 ₁₀ 、1010 ₂ は16進数ではいくつか？	Hex <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> d 123 <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> b 1010 <input type="checkbox"/> EXE	0000007B 0000000A
15 ₁₆ 、1100 ₂ は8進数ではいくつか？	Oct <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> h 15 <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> b 1100 <input type="checkbox"/> EXE	0000000025 0000000014
36 ₁₀ 、3B7 ₁₆ は2進数ではいくつか？	Bin <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> d 36 <input type="checkbox"/> EXE SHIFT <input type="checkbox"/> h 3B7 <input type="checkbox"/> EXE	0000000000100100 0000001110110111

■負数の表現

例 題	操 作	表 示
110010 ₂ の負数は？	<input type="checkbox"/> Bin <input type="checkbox"/> EXE <input type="checkbox"/> Neg 110010 <input type="checkbox"/> EXE	1111111111001110
72 ₈ の負数は？	<input type="checkbox"/> Oct <input type="checkbox"/> EXE <input type="checkbox"/> Neg 72 <input type="checkbox"/> EXE	3777777706
3A ₁₆ の負数は？	<input type="checkbox"/> Hex <input type="checkbox"/> EXE <input type="checkbox"/> Neg 3A <input type="checkbox"/> EXE	FFFFFFFC6

■2進・8進・10進・16進の加減乗除算

例 題	操 作	表 示
10111 ₂ +11010 ₂ =110001 ₂	<input type="checkbox"/> Bin <input type="checkbox"/> EXE 10111 <input type="checkbox"/> + 11010 <input type="checkbox"/> EXE	000000000110001
B47 ₁₆ -DF ₁₆ =A68 ₁₆	<input type="checkbox"/> Hex <input type="checkbox"/> EXE B47 <input type="checkbox"/> - DF <input type="checkbox"/> EXE	00000A68
123 ₈ ×ABC ₁₆ =37AF ₁₆ =228084 ₁₀	<input type="checkbox"/> SHIFT <input type="checkbox"/> 0 123 <input type="checkbox"/> × ABC <input type="checkbox"/> EXE <input type="checkbox"/> Dec <input type="checkbox"/> EXE	00037AF4 228084
1F2D ₁₆ -100 ₁₀ =7881 ₁₀ =1EC9 ₁₆	<input type="checkbox"/> SHIFT <input type="checkbox"/> h 1F2D <input type="checkbox"/> - 100 <input type="checkbox"/> EXE <input type="checkbox"/> Hex <input type="checkbox"/> EXE	7881 00001EC9
7654 ₈ ÷12 ₁₀ =334.3333333 ₁₀ =516 ₈	<input type="checkbox"/> Dec <input type="checkbox"/> EXE <input type="checkbox"/> SHIFT <input type="checkbox"/> 0 7654 <input type="checkbox"/> ÷ 12 <input type="checkbox"/> EXE <input type="checkbox"/> Oct <input type="checkbox"/> EXE	334 0000000516
1234+1EF ₁₆ ÷24 ₈ =2352 ₈ =1258 ₁₀	<input type="checkbox"/> SHIFT <input type="checkbox"/> d 1234 <input type="checkbox"/> + <input type="checkbox"/> SHIFT <input type="checkbox"/> h 1EF <input type="checkbox"/> ÷ 24 <input type="checkbox"/> EXE <input type="checkbox"/> Dec <input type="checkbox"/> EXE	0000002352 1258

※計算結果は小数点以下が切り捨てられて表示されます。

※混合計算は加減より乗除優先で計算されます。

■論理演算

論理演算は論理積(and)、論理和(or)、排他的論理和(xor)、否定(Not)について行なわれます。

例 題	操 作	表 示
$19_{16} \text{ AND } 1A_{16} = 18_{16}$	Hex EXE 19 and 1A EXE	00000018
$1110_2 \text{ AND } 36_8 = 1110_2$	Bin EXE 1110 and SHIFT O 36 EXE	0000000000001110
$23_8 \text{ OR } 61_8 = 63_8$	Oct EXE 23 or 61 EXE	000000000063
$120_{16} \text{ OR } 1101_2 = 12D_{16}$	Hex EXE 120 or SHIFT b 1101 EXE	0000012D
$1010_2 \text{ AND } (A_{16} \text{ OR } 7_{16}) = 1010_2$	Bin EXE 1010 and (SHIFT h A or SHIFT h 7) EXE	0000000000001010
$5_{16} \text{ XOR } 3_{16} = 6_{16}$	Hex EXE 5 SHIFT xor 3 EXE	00000006
$42_{10} \text{ XOR } B_{16} = 33_{10}$	Dec EXE 42 SHIFT xor SHIFT xor SHIFT h B EXE	33
1234 ₈ の否定	Oct EXE Not 1234 EXE	37777776543
2FFFD ₁₆ の否定	Hex EXE Not 2FFFD EXE	FFD00012

2-5 統計計算の仕方

■標準偏差計算

- **[MODE]** **[x]** と押し SD 1 モードで行ないます。
- 計算を始める前に **[SHIFT]** **[Sci]** **[EXE]** と押し集計用メモリーをクリアしてください。
- 個々のデータは〈データ **[DT]** (**[x]** キー)〉と押します。
- 同一データが複数のときは、**[DT]** キーをその回数だけ続けて押すか、または〈データ **[SHIFT]** **[n]** 個数 **[DT]**〉と押します。
- 標準偏差

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} = \sqrt{\frac{\sum x^2 - (\sum x)^2/n}{n}} \quad \left[\begin{array}{l} \text{有限母集団全部のデータを使い} \\ \text{その集団の標準偏差を求める} \end{array} \right]$$

$$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{\sum x^2 - (\sum x)^2/n}{n-1}} \quad \left[\begin{array}{l} \text{集団中のサンプルデータを使い} \\ \text{その集団の標準偏差を推定する} \end{array} \right]$$

- 平均

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{\sum x}{n}$$

※ n 、 $\sum x$ 、 $\sum x^2$ はそれぞれメモリーのW、V、Uに格納されますので、**[ALPHA]** **[W]** **[EXE]** のようにして答えを求めます。

例 題	操 作	表 示		
データ55、54、51、55、53、53、54、52の場合	[MODE] [X] [SHIFT] [Sci] [EXE] (メモリーク リアー) 55 [DT] 54 [DT] 51 [DT] 55 [DT] 53 [DT] [DT] 54 [DT] 52 [DT]	52.		
※答えは、どの順にキーを押してもかまいません。	(標準偏差 σ_n は)			
	[SHIFT] [σ_n] [EXE]	1. 316956719		
	(標準偏差 σ_{n-1} は)			
	[SHIFT] [σ_{n-1}] [EXE]	1. 407885953		
	(平均 \bar{x} は)	53. 375		
	(データ n は)	8.		
	(総和 Σx は)	427.		
	(2乗和 Σx^2 は)			
	[ALPHA] [U] [EXE]	22805.		
前例データの不偏分散および各データと平均との差は	(続けて) [SHIFT] [σ_{n-1}] [x^2] [EXE]	1. 982142857		
	55 [] [SHIFT] [] [EXE]	1. 625		
	54 [] [SHIFT] [] [EXE]	0. 625		
	51 [] [SHIFT] [] [EXE]	-2. 375		
	∴	∴		
下表 \bar{x} および σ_{n-1} は?	[SHIFT] [Sci] [EXE]			
階級No.	階級値	度数	110 [SHIFT] [] 10 [DT]	110.
1	110	10	130 [SHIFT] [] 31 [DT]	130.
2	130	31	150 [SHIFT] [] 24 [DT]	150.
3	150	24	170 [DT] [DT]	170.
4	170	2	190 [DT] [DT] [DT]	190.
5	190	3	[ALPHA] [W] [EXE]	70.
			[SHIFT] [] [EXE]	137. 7142857
			[SHIFT] [σ_{n-1}] [EXE]	18. 42898069

☆ミス入力したデータの削除、訂正 I (正しくは51 **[DT]** と操作)

- ①50 **[DT]** →続けて **[CL]** (**[x]** キー)、その後正しく操作
- ②何個か前の49 **[DT]** →49**[CL]**、その後正しく操作

☆ミス入力したデータの削除、訂正 II (正しくは130 **[SHIFT]** **[↓]** 31 **[DT]** と操作)

- ①120 **[SHIFT]** **[↓]** →**[AC]** 後正しく操作
- ②120 **[SHIFT]** **[↓]** 31→**[AC]** 後正しく操作
- ③120 **[SHIFT]** **[↓]** 30 **[DT]** →**[CL]**、その後正しく操作
- ④何行か前の120 **[SHIFT]** **[↓]** 30 **[DT]** →120 **[SHIFT]** **[↓]** 30 **[CL]**、その後正しく操作

■回帰計算

- **[MODE]** **[+]** と押し LR1 モードで行ないます。
- 計算を始める前に **[SHIFT]** **[Sci]** **[EXE]** と押して集計用のメモリーをクリアーしてください。
- 個々のデータは $\langle x$ データ **[SHIFT]** **[↓]** y データ **[DT]** と押します。
- 同一データが複数のときは、**[DT]** キーをその回数だけ続けて押すか、または $\langle x$ データ **[SHIFT]** **[↓]** y データ **[SHIFT]** **[↓]** 個数 **[DT]** を押します。
- x データだけが同じときは **[SHIFT]** **[↓]** y データ **[DT]** または **[SHIFT]** **[↓]** y データ **[SHIFT]** **[↓]** 個数 **[DT]** と押します。
- y データだけが同じときは $\langle x$ データ **[DT]** または $\langle x$ データ **[SHIFT]** **[↓]** 個数 **[DT]** と押します。
- 回帰式は $y = A + Bx$ で、定数項 A 、回帰係数 B は次式により計算されます。

回帰式の回帰係数

回帰式の定数項

$$B = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{n \cdot \sum x^2 - (\sum x)^2}$$

$$A = \frac{\sum y - B \cdot \sum x}{n}$$

- 回帰式をもとにした推定値 \hat{x} 、 \hat{y} の計算ができます。
- 入力したデータ対の相関係数 r は次式により計算されます。

$$r = \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\{n \cdot \sum x^2 - (\sum x)^2\} \{n \cdot \sum y^2 - (\sum y)^2\}}}$$

※ n 、 $\sum x$ 、 $\sum x^2$ 、 $\sum xy$ 、 $\sum y$ 、 $\sum y^2$ はそれぞれメモリーの W 、 V 、 U 、 R 、 Q 、 P に格納されますので、**[ALPHA]** **[W]** **[EXE]** のようにして答えを求めます。

◆直線回帰計算

例 題	操 作	表 示												
● 棒鋼の温度と長さ	MODE $\left[\frac{\square}{\square} \right]$													
<table border="1" data-bbox="133 290 380 514"> <thead> <tr> <th>温 度</th> <th>長さの測定値</th> </tr> </thead> <tbody> <tr><td>10℃</td><td>1003mm</td></tr> <tr><td>15</td><td>1005</td></tr> <tr><td>20</td><td>1010</td></tr> <tr><td>25</td><td>1011</td></tr> <tr><td>30</td><td>1014</td></tr> </tbody> </table>	温 度	長さの測定値	10℃	1003mm	15	1005	20	1010	25	1011	30	1014	SHIFT Sci EXE (メモリーク リアー) 10 SHIFT $\left[\frac{\square}{\square} \right]$ 1003 DT 15 SHIFT $\left[\frac{\square}{\square} \right]$ 1005 DT 20 SHIFT $\left[\frac{\square}{\square} \right]$ 1010 DT 25 SHIFT $\left[\frac{\square}{\square} \right]$ 1011 DT 30 SHIFT $\left[\frac{\square}{\square} \right]$ 1014 DT	10. 15. 20. 25. 30.
温 度	長さの測定値													
10℃	1003mm													
15	1005													
20	1010													
25	1011													
30	1014													
この表より回帰式と相関係数を求め、回帰式をもとに、温度18℃のときの長さおよび1000mmの温度を推定する。	(定数項 A は) SHIFT A EXE	997.4												
また決定係数(r^2)および共分散($\frac{\sum xy - n \cdot \bar{x} \cdot \bar{y}}{n-1}$)を計算する。	(回帰係数 B は) SHIFT B EXE	0.56												
	(相関係数 r は) SHIFT $\left[\frac{\square}{\square} \right]$ EXE	0.9826073689												
	(18℃のときの長さは) 18 SHIFT $\left[\frac{\square}{\square} \right]$ EXE	1007.48												
	(1000mmのときの温度は) 1000 SHIFT $\left[\frac{\square}{\square} \right]$ EXE	4.642857142												
	(決定係数は) SHIFT $\left[\frac{\square}{\square} \right]$ x^2 EXE	0.9655172414												
	(共分散は) $\left[\left[\frac{\square}{\square} \right] \right]$ ALPHA R $\left[\frac{\square}{\square} \right]$													
	ALPHA W $\left[\frac{\square}{\square} \right]$ SHIFT $\left[\frac{\square}{\square} \right]$ $\left[\frac{\square}{\square} \right]$ SHIFT													
	$\left[\frac{\square}{\square} \right]$ $\left[\frac{\square}{\square} \right]$ $\left[\frac{\square}{\square} \right]$ ALPHA W $\left[\frac{\square}{\square} \right]$ $\left[\frac{\square}{\square} \right]$ $\left[\frac{\square}{\square} \right]$													
	EXE	35.												

☆ミス入力したデータの削除・訂正(正しくは10 **SHIFT** $\left[\frac{\square}{\square} \right]$ 1003 **DT**)と操作)

- ①11 **SHIFT** $\left[\frac{\square}{\square} \right]$ 1003 \rightarrow **AC** 後正しく操作
- ②11 **SHIFT** $\left[\frac{\square}{\square} \right]$ 1003 **DT** \rightarrow 続けて **CL**、その後正しく操作
- ③何行か前の11 **SHIFT** $\left[\frac{\square}{\square} \right]$ 1003 **DT** \rightarrow 11 **SHIFT** $\left[\frac{\square}{\square} \right]$ 1003 **CL** その後正しく操作

◆対数回帰計算

●回帰式は $y=A+B\cdot\ln x$ で、 x データは x の対数(\ln)を入力し、 y データは直線回帰と同様に入力します。

●回帰係数の求め方や訂正方法は直線回帰と同様な操作で求められますが、推定値 \hat{y} は $\boxed{\ln} \boxed{x} \boxed{\text{SHIFT}} \boxed{\hat{y}} \boxed{\text{EXE}}$ で、推定値 \hat{x} は $\boxed{y} \boxed{\text{SHIFT}} \boxed{\hat{x}} \boxed{\text{EXE}} \boxed{\text{SHIFT}} \boxed{e^x} \boxed{\text{Ans}} \boxed{\text{EXE}}$ で求められます。

なお、 Σx には $\Sigma \ln x$ 、 Σx^2 には $\Sigma (\ln x)^2$ 、 Σxy には $\Sigma \ln x \cdot y$ が求められています。

例	題	操 作	表 示												
<table border="1"> <thead> <tr> <th>x_i</th> <th>y_i</th> </tr> </thead> <tbody> <tr> <td>29</td> <td>1.6</td> </tr> <tr> <td>50</td> <td>23.5</td> </tr> <tr> <td>74</td> <td>38.0</td> </tr> <tr> <td>103</td> <td>46.4</td> </tr> <tr> <td>118</td> <td>48.9</td> </tr> </tbody> </table>	x_i	y_i	29	1.6	50	23.5	74	38.0	103	46.4	118	48.9		$\boxed{\text{MODE}} \boxed{+}$ $\boxed{\text{SHIFT}} \boxed{\text{Sci}} \boxed{\text{EXE}}$ $\boxed{\ln} \boxed{29} \boxed{\text{SHIFT}} \boxed{\downarrow} \boxed{1.6} \boxed{\text{DT}}$ $\boxed{\ln} \boxed{50} \boxed{\text{SHIFT}} \boxed{\downarrow} \boxed{23.5} \boxed{\text{DT}}$ $\boxed{\ln} \boxed{74} \boxed{\text{SHIFT}} \boxed{\downarrow} \boxed{38.0} \boxed{\text{DT}}$ $\boxed{\ln} \boxed{103} \boxed{\text{SHIFT}} \boxed{\downarrow} \boxed{46.4} \boxed{\text{DT}}$ $\boxed{\ln} \boxed{118} \boxed{\text{SHIFT}} \boxed{\downarrow} \boxed{48.9} \boxed{\text{DT}}$ (定数項 A は) $\boxed{\text{SHIFT}} \boxed{\text{A}} \boxed{\text{EXE}}$ (係数 B は) $\boxed{\text{SHIFT}} \boxed{\text{B}} \boxed{\text{EXE}}$ (相関係数 r は) $\boxed{\text{SHIFT}} \boxed{\text{r}} \boxed{\text{EXE}}$ ($x_i=80$ のとき \hat{y} は) $\boxed{\ln} \boxed{80} \boxed{\text{SHIFT}} \boxed{\hat{y}} \boxed{\text{EXE}}$ ($y_i=73$ のとき \hat{x} は) $\boxed{73} \boxed{\text{SHIFT}}$ $\boxed{\hat{x}} \boxed{\text{EXE}} \boxed{\text{SHIFT}} \boxed{e^x} \boxed{\text{Ans}} \boxed{\text{EXE}}$	3. 36729583 3. 912023005 4. 304065093 4. 634728988 4. 770684624 -111. 1283976 34. 0201475 0. 9940139466 37. 94879482 224. 1541313
x_i	y_i														
29	1.6														
50	23.5														
74	38.0														
103	46.4														
118	48.9														

上記データを対数回帰して回帰式および相関係数を求める。

また、回帰式より $x_i=80$ および $y_i=73$ のときの \hat{y} 、 \hat{x} をそれぞれ推定する。

◆指数回帰計算

- 回帰式は $y = A \cdot e^{Bx}$ ($\ln y = \ln A + Bx$) で、 y データは y の対数 (\ln) を入力し、 x データは直線回帰と同様に入力します。
- 訂正方法は直線回帰と同様の操作ですが、定数項 A は $\text{SHIFT} \text{ } e^x \text{ } \text{SHIFT} \text{ } [A] \text{ } \text{EXE}$ で、推定値 \hat{y} は $x \text{ } \text{SHIFT} \text{ } [D] \text{ } \text{EXE} \text{ } \text{SHIFT} \text{ } \text{ } e^x \text{ } \text{Ans} \text{ } \text{EXE}$ で、推定値 \hat{x} は $\ln y \text{ } \text{SHIFT} \text{ } [D] \text{ } \text{EXE}$ で求められます。なお、 Σy には $\Sigma \ln y$ 、 Σy^2 には $\Sigma (\ln y)^2$ 、 Σxy には $\Sigma x \cdot \ln y$ が求められています。

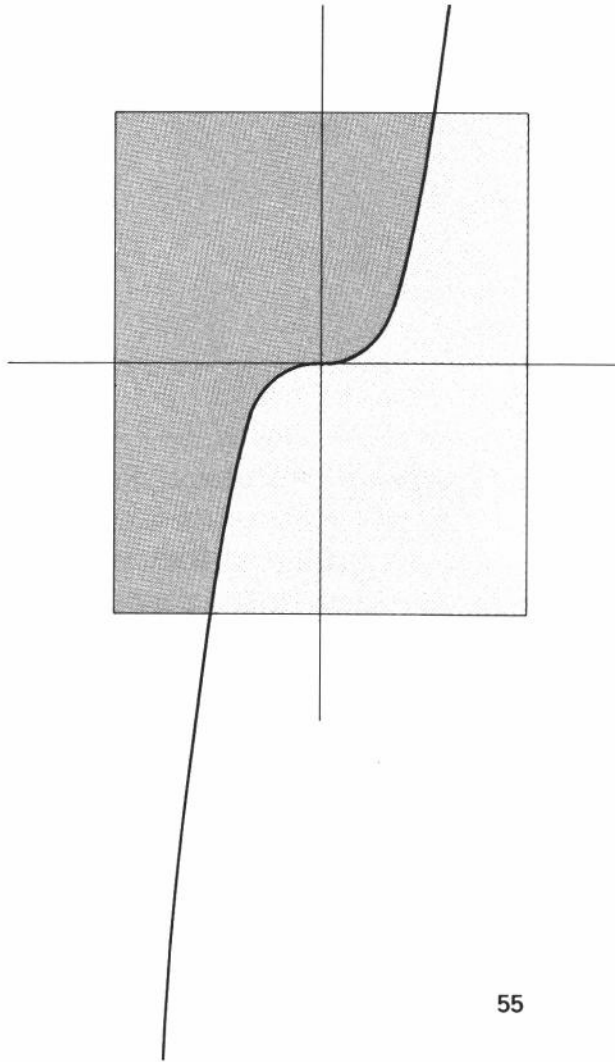
例	題	操	作	表	示												
	<table border="1"> <thead> <tr> <th>x_i</th> <th>y_i</th> </tr> </thead> <tbody> <tr> <td>6.9</td> <td>21.4</td> </tr> <tr> <td>12.9</td> <td>15.7</td> </tr> <tr> <td>19.8</td> <td>12.1</td> </tr> <tr> <td>26.7</td> <td>8.5</td> </tr> <tr> <td>35.1</td> <td>5.2</td> </tr> </tbody> </table>	x_i	y_i	6.9	21.4	12.9	15.7	19.8	12.1	26.7	8.5	35.1	5.2				
x_i	y_i																
6.9	21.4																
12.9	15.7																
19.8	12.1																
26.7	8.5																
35.1	5.2																
			$\text{MODE} \text{ } \div$ $\text{SHIFT} \text{ } [\text{Sci}] \text{ } \text{EXE}$ 6.9 $\text{SHIFT} \text{ } [.] \text{ } \ln \text{ } 21.4 \text{ } \text{DT}$ 12.9 $\text{SHIFT} \text{ } [.] \text{ } \ln \text{ } 15.7 \text{ } \text{DT}$ 19.8 $\text{SHIFT} \text{ } [.] \text{ } \ln \text{ } 12.1 \text{ } \text{DT}$ 26.7 $\text{SHIFT} \text{ } [.] \text{ } \ln \text{ } 8.5 \text{ } \text{DT}$ 35.1 $\text{SHIFT} \text{ } [.] \text{ } \ln \text{ } 5.2 \text{ } \text{DT}$		6.9 12.9 19.8 26.7 35.1												
	上記データを指数回帰して回帰式および相関係数を求める。		(定数項 A は) $\text{SHIFT} \text{ } \text{ } e^x \text{ } \text{SHIFT} \text{ } [A] \text{ } \text{EXE}$		30.49758743												
	また、回帰式より $x_i=16$ 、および $y_i=20$ のときの \hat{y} 、 \hat{x} をそれぞれ推定する。		(係数 B は) $\text{SHIFT} \text{ } [B] \text{ } \text{EXE}$ (相関係数 r は) $\text{SHIFT} \text{ } [r] \text{ } \text{EXE}$ ($x_i=16$ のとき \hat{y} は) 16 $\text{SHIFT} \text{ } [D] \text{ } \text{EXE} \text{ } \text{SHIFT} \text{ } \text{ } e^x$ $\text{Ans} \text{ } \text{EXE}$ ($y_i=20$ のとき \hat{x} は) $\ln \text{ } 20 \text{ } \text{SHIFT} \text{ } [D] \text{ } \text{EXE}$		-0.04920370831 -0.997247352 13.87971539 8.574868046												

◆べき乗回帰計算

- 回帰式は $y = A \cdot x^B (\ln y = \ln A + B \ln x)$ で、データは $x \cdot y$ とともに対数 (ln) を入力します。
- 訂正方法は直線回帰と同様の操作ですが、定数項 A は $\text{[SHIFT] [e^x] [SHIFT] [A] [EXE]}$ で、推定値 \hat{y} は $\text{[ln] } x \text{ [SHIFT] [y] [EXE] [SHIFT] [e^x] [Ans] [EXE]}$ で、 \hat{x} は $\text{[ln] } y \text{ [SHIFT] [x] [EXE] [SHIFT] [e^x] [Ans] [EXE]}$ で求められます。なお、 Σx には $\Sigma \ln x$ 、 Σx^2 には $\Sigma (\ln x)^2$ 、 Σy には $\Sigma \ln y$ 、 Σy^2 には $\Sigma (\ln y)^2$ 、 Σxy には $\Sigma \ln x \cdot \ln y$ が求められています。

例	題	操	作	表	示												
	<table border="1"> <thead> <tr> <th>x_i</th> <th>y_i</th> </tr> </thead> <tbody> <tr> <td>28</td> <td>2410</td> </tr> <tr> <td>30</td> <td>3033</td> </tr> <tr> <td>33</td> <td>3895</td> </tr> <tr> <td>35</td> <td>4491</td> </tr> <tr> <td>38</td> <td>5717</td> </tr> </tbody> </table>	x_i	y_i	28	2410	30	3033	33	3895	35	4491	38	5717				
x_i	y_i																
28	2410																
30	3033																
33	3895																
35	4491																
38	5717																
		[MODE] [÷] $\text{[SHIFT] [Sci] [EXE]}$ $\text{[ln] } 28 \text{ [SHIFT] [r] [ln] } 2410$ [DT] $\text{[ln] } 30 \text{ [SHIFT] [r] [ln] } 3033$ [DT] $\text{[ln] } 33 \text{ [SHIFT] [r] [ln] } 3895$ [DT] $\text{[ln] } 35 \text{ [SHIFT] [r] [ln] } 4491$ [DT] $\text{[ln] } 38 \text{ [SHIFT] [r] [ln] } 5717$ [DT]			<p>3. 33220451</p> <p>3. 401197382</p> <p>3. 496507561</p> <p>3. 555348061</p> <p>3. 63758616</p>												
	<p>上記データをべき乗回帰して、回帰式および相関係数を求める。</p> <p>また回帰式より $x_i=40$ および $y_i=1000$ のときの \hat{y}、\hat{x} をそれぞれ推定する。</p>	<p>(定数項 A は)</p> $\text{[SHIFT] [e^x] [SHIFT] [A] [EXE]}$ <p>(係数 B は)</p> [SHIFT] [B] [EXE] <p>(相関係数 r は)</p> [SHIFT] [r] [EXE] <p>($x_i=40$ のとき \hat{y} は)</p> $\text{[ln] } 40 \text{ [SHIFT] [y] [EXE] [SHIFT] [e^x] [Ans] [EXE]}$ <p>($y_i=1000$ のとき \hat{x} は)</p> $\text{[ln] } 1000 \text{ [SHIFT] [x] [EXE] [SHIFT] [e^x] [Ans] [EXE]}$			<p>0. 2388010724</p> <p>2. 771866153</p> <p>0. 9989062542</p> <p>6587. 675458</p> <p>20. 2622568</p>												

第3章 グラフィック



本機の特長であるグラフ機能は、大型表示の95×32ドット(最上部は未使用)をフルに使い、色々な関数グラフや統計グラフが簡単に、素早く描けます。描けるグラフは組み込み関数グラフの他に、“y=”の形で表わされる関数式のグラフが任意に描け、式を目で見ることができます。

なお、グラフ命令はマニュアルでもプログラム中に書き込んでも使えますが、ここではマニュアル中心に説明します。

プログラム中での使い方はマニュアルと同様で、詳しくは127ページをご覧ください。

※**ALPHA** キーの後のキーは、アルファベットを表記したキーマークとなっており、キーの右下にある赤色の文字を参照して押してください。

例) **Graph** **sin** **ALPHA** **X**
↳ **田** キーです。

3-1 組み込み関数グラフ

関数グラフを描くときは、RUN モードの COMP モードにしてください。SD モード、LR モードも描けますが、一部の関数が使えなくなります。(Base-n モードでは使えません)

組み込み関数グラフは全部で fx-6500G は20種類、fx-6000G は14種類あり、各々の基本的関数グラフを自動的に描きます。

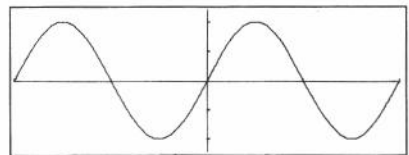
- sin
- cos
- tan
- \sin^{-1}
- \cos^{-1}
- \tan^{-1}
- sinh
- cosh
- tanh
- \sinh^{-1}
- \cosh^{-1}
- \tanh^{-1}
- $\sqrt{\quad}$
- x^2
- log
- ln
- 10^x
- e^x
- x^{-1}
- $\sqrt[3]{\quad}$

※sinh、cosh、tanh、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} は fx-6500G のみできます。

なお、組み込み関数を実行しますとレンジ(グラフの範囲、60ページ参照)は自動的に最適なレンジに設定され、グラフを分割して表示します。なお、以前のグラフ表示はクリアーされます。

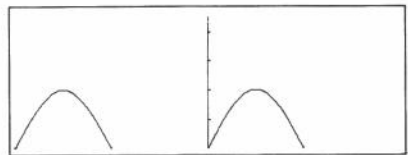
例 サインカーブ

MODE \oplus
Graph sin EXE

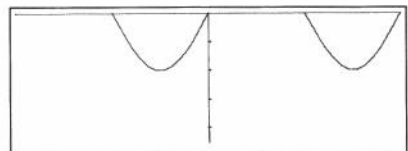


まず、グラフの中心を表示しますので、このときに \uparrow キーまたは \downarrow を押すことにより、上半分または下半分を分割して表示します。

\uparrow

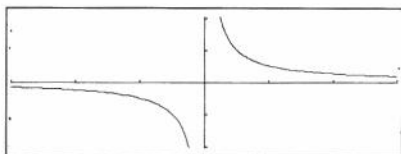


\downarrow \downarrow

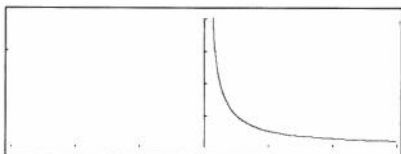


例 $y = \frac{1}{x}$ のグラフ

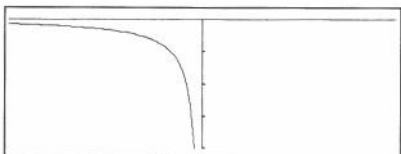
Graph x^{-1} EXE



↑



↓ ↓



※この機能は任意の関数グラフでも、MODE \square と押すことにより行なえます。

■組み込み関数グラフの重ね描き

組み込み関数グラフを使って、2種類以上のグラフを重ねて描くことができます。このとき、最初のグラフでレンジが自動的に設定されますので、2つ目以降は、最初のグラフのレンジに合わせた範囲に描かれます。

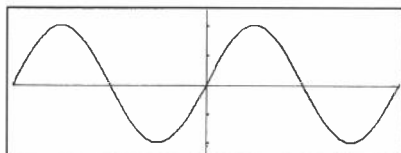
操作方法は、最初のグラフは前出の例と同じに、“Graph [関数キー] EXE”としますが、次のグラフから、“Graph [関数キー] ALPHA X EXE”と操作して、変数Xを使います。

この[関数キー]の後に ALPHA X を入力することにより、レンジを変更せず、表示をクリアすることなく任意のグラフを描くことができます。(詳しくは65ページ参照)

例 $y = \sin x$ のグラフに、 $y = \cos x$ のグラフを重ねる。

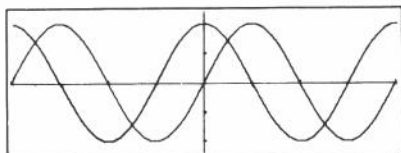
まず、 $y = \sin x$ のグラフを描きます。

Graph sin EXE

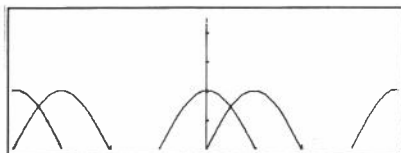


次に、レンジをかえずに $y = \cos x$ のグラフを描きます。

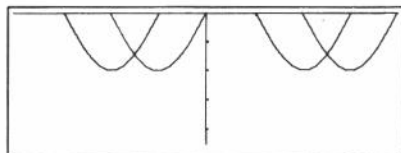
Graph cos ALPHA X EXE



↑



↓ ↓



〈注 意〉

組み込み関数グラフはマルチステートメント(36ページ)での使用、およびプログラムに組みこむことはできません。

3-2 任意の関数グラフ(式を目で見る)

任意の関数グラフとは、前出の組み込み関数を単独で使うだけでなく、その関数を組み合わせて使ったり、“ $y = 2x^2 + 3x - 5$ ”のような式をグラフにして、**目で見る**ことができます。

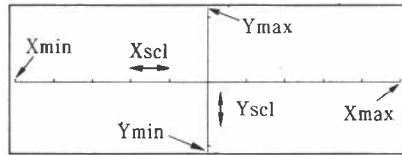
また、任意の関数グラフでは、**レンジ**と呼ばれるグラフを描く範囲が組み込み関数グラフと違って自動的に設定されませんので、レンジの範囲外のグラフを描こうとしても、表示されません。

■レンジ(Range)

レンジとは、 x 軸・ y 軸の範囲および両軸に刻まれる目盛りの間隔で、この内容を **Range** キーを押すことにより、設定したり確認することができます。

●レンジの内容

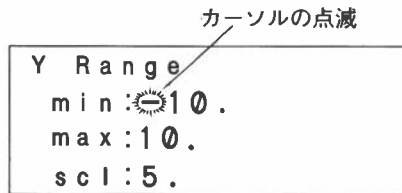
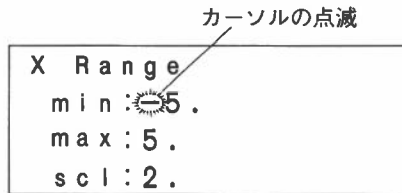
レンジの内容はXmin(x 軸の最小値)、Xmax(x 軸の最大値)、Xscl(x 軸の目盛の間隔)、Ymin(y 軸の最小値)、Ymax(y 軸の最大値)、Yscl(y 軸の目盛の間隔)です。



●レンジ画面

Range キーを押すと、レンジ内容を右のようにXレンジとYレンジに分けて表示します。

カーソル()と交互に点滅している所を設定し直すことができます。



※レンジ内容により、この表示と同じにならないこともあります。

●レンジの設定

レンジを設定するには、現在カーソルが点滅している位置から順に Xmin→Xmax→Xscl→Ymin→Ymax→Yscl と値を入力していきます。

値の入力はカーソルの点滅位置で置数し、**[EXE]** キーを押します。このとき、直接置数して **[EXE]** キーを押すと入力された値に変更されますが、**[↔]** キーにより表示されている値の2桁目以降にカーソルを合わせて置数すると、この部分のみの値が変更されます。

では、上記内容を次のように設定してみます。

Xmin	→	0	Ymin	→	-5
Xmax	→	5	Ymax	→	15
Xscl	→	1	Yscl	→	5

①Xmin を 0 にするので、0 を入力。

0 **[EXE]**

```
X Range
min: 0
max: 5
scl: 2
```

②Xmax はそのままですので、**[EXE]** キーだけを押します。

[EXE]

(**[↔]** キーでも可)

```
X Range
min: 0
max: 5
scl: 2
```

③Xscl は 1 にするので、1 を入力。

1 **[EXE]**

(このときレンジ設定の表示が
Yレンジにかかります。)

```
Y Range
min: 10
max: 10
scl: 5
```

④Ymin は-5 ですので、1桁 $\left[\rightleftharpoons \right]$ キーで送ってから5を入力。

$\left[\rightleftharpoons \right]$ 5 $\left[\text{EXE} \right]$

```
Y Range
min : -5
max : 10.
scl : 5.
```

⑤Ymax は15 ですので、1桁 $\left[\rightleftharpoons \right]$ キーで送ってから5を入力。

$\left[\rightleftharpoons \right]$ 5 $\left[\text{EXE} \right]$

```
Y Range
min : -5
max : 15
scl : 5.
```

⑥Yscl は5 ですので、そのまま $\left[\text{EXE} \right]$ キーだけを押します。

$\left[\text{EXE} \right]$

全ての設定が終了すると $\left[\text{Range} \right]$ キーを押す前の表示に戻ります。

レンジ内容が正しく設定されたかは、もう一度 $\left[\text{Range} \right]$ キーを押して確認できます。

$\left[\text{Range} \right]$

```
X Range
min : 0.
max : 5.
scl : 1.
```

$\left[\text{Range} \right]$

※続けて $\left[\text{Range} \right]$ キーを押すとYレンジを表示
します。

```
Y Range
min : 5.
max : 15.
scl : 5.
```

このレンジ内容表示中に部分的に設定したいときや、 $\left[\text{EXE} \right]$ キーを押すすぎて通りすぎてしまった部分に戻るときなどは、 $\left[\updownarrow \right]$ ・ $\left[\leftarrow \rightarrow \right]$ キーで自由にカーソルを上下することができます。(このとき、値は変わりません。)なお、 $\left[\updownarrow \right]$ ・ $\left[\leftarrow \rightarrow \right]$ キーによりカーソルを上下したときは、上はXminまで、下はYsclまでカーソルが移動し、それ以上先に進んだり、別の表示になることはありません。

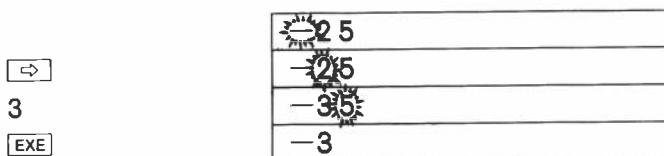
レンジ内容表示中に元の表示に戻すには、Yレンジ表示中にもう一度 **Range** キーを押します。

※レンジの入力範囲は、 $-9.9999\text{E}+98 \sim 9.9999\text{E}+98$ です。

※レンジ内容表示中は、**0** ~ **9**、**.**、**EXP**、**(-)** および **←**、**→**、**↑**、**↓**、**Range** キー以外の入力はできません。(負符号は **(-)** キーを使い“(-) 5”のように押します)

※レンジ内容の設定で数値を入力する際、新たな数値の入力はカーソルが先頭(左端)にあるときに入力してください。**←**・**→** キーでカーソルを移動すると、カーソル位置から後(右側)は入力されず、またカーソルより前(左側)はそのまま入力されます。

例



※入力できる数値は有効桁数9桁までで、 10^{-2} 未満または 10^9 以上は仮数部6桁(負符号を含む)指数部2桁にして表示します。

※不適当な入力(計算範囲外や負符号のみ等)は、前回の数値を変更することはありません。(一時的に表示されるだけです)

※Xscl・Ysclとも、値を0とすると目盛りは打ちません。

※レンジの範囲で、min 値より max 値が小さい場合は、軸の方向が逆になります。

例 X min : 5

X max : -5



※レンジの範囲で、min 値と max 値が等しい場合、グラフを描いたときにエラー (Ma ERROR) となります。

※レンジの設定により座標軸が表示できない場合は、y 軸の代りに表示窓の左端か右端に、x 軸の代りに表示窓の上端か下端に目盛が表示されます。(原点に近い側の枠に目盛をとります)

※レンジ内容が変更(再設定)された場合は、グラフ表示はクリアーされ、設定されている座標軸のみ表示されます。

※レンジ内容により、目盛の間隔が一定とならない場合があります。

※レンジの範囲を広くとりすぎると、不連続グラフを結んで実際とは異なる表示をすることがあります。

※変曲点の近辺で急激に変化するようなグラフでは、変曲点が表示し切れない場合があります。

※レンジ内容により、レンジオーバーとなる目盛を打たせると、Ma ERROR となります。

例

Xmin 9.ε99

Xmax 9.9ε99

Xscl 1.ε99 ⇒ 目盛を計算すると計算範囲外となる。

⋮

※レンジの範囲が極端にせまい場合、Ma ERROR となることがあります。

●レンジのリセット

レンジ内容表示中に **SHIFT** **DEL** キーを押すと、レンジ内容がリセットされ、初期設定になります。

Range (レンジ内容表示中は省略)

SHIFT **DEL**

```
X Range
min:-4.7
max:4.7
scl:1.
```

Range

```
Y Range
min:-3.1
max:3.1
scl:1.
```

〈参考〉

プログラム中にレンジ設定を組み込むには

Range Xmin 値, Xmax 値, Xscl 値, Ymin 値, Ymax 値, Yscl 値
と書き込みます。

Range の後のデータは 6 個以内で、6 個未満のときは先頭から順に設定します。

■任意の関数グラフを描く

任意の関数グラフはレンジを設定した後、**Graph** キーに続いて関数値(計算式)を入力することにより、簡単に描けます。

まずは、“ $y = 2x^2 + 3x - 4$ ”のグラフを描いてみます。

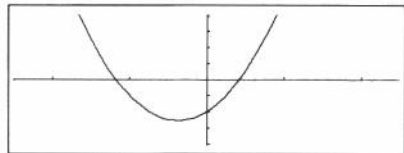
レンジは次の内容に設定します。

X Range
 min : -5.
 max : 5.
 scl : 2.

Y Range
 min : -8.
 max : 8.
 scl : 2.

Graph キーに続けて関数式を入力します。

Graph 2 **ALPHA** **X** x^2 + 3 **ALPHA** **X** -
 4 **EXE**



このようにすれば、関数式をすぐに目で見ることができます。

また、グラフを描いた直後に **←** **→** キーを押して、リプレイ機能で数値をかえてグラフを描くことができます。(74ページ参照)

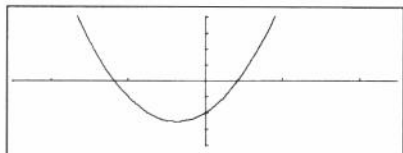
■関数グラフの重ね描き

2種類以上の関数式によるグラフを描くことにより、その交点の位置や解の数がすぐにわかるようになります。

例 前出の“ $y = 2x^2 + 3x - 4$ ”のグラフと“ $y = 2x + 3$ ”のグラフの交点を見ましょう。

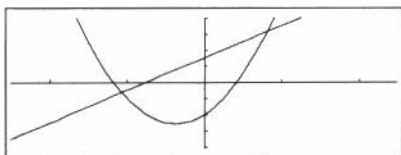
まず、“ $y = 2x^2 + 3x - 4$ ”のグラフを描きますので、最初にグラフィック画面をクリアしてから行ないます。

SHIFT **Cls** **EXE**
Graph 2 **ALPHA** **X** x^2 + 3 **ALPHA** **X** -
 4 **EXE**



次に、この上から、“ $y = 2x + 3$ ”のグラフを重ねます。

Graph 2 ALPHA X + 3 EXE



このように、2種類のグラフを重ねると、交点が2つあることがすぐにわかります。

この2つの交点のだいたいの座標を求めるには、次の項目で説明するトレース(Trace)機能を使うとすぐに求められます。

※グラフの重ね描きで組み込み関数を使うときなど、必ず関数に変数X(ALPHA X)を入れてください。変数Xを入れない場合は、組み込み関数ではグラフィック表示をクリアしてから新たに描きなおします。

■トレース(Trace)機能

グラフ上のポインター(点滅している点)をカーソルキー(← →)で動かし、 x 座標、 y 座標を読み取ることができます。

トレース機能は、グラフ表示直後に [SHIFT] [Trace] と押すことにより、グラフ上の左端にポインターが点滅し、最下段に x 座標値($x = \dots$)が表示されます。

点滅しているポインターは [←] または [→] キーにより移動し、そのときの x 座標値を表示します。 x 座標値表示中に y 座標値を表示させたいときは [SHIFT] [X↔Y] と押します。[SHIFT] [X↔Y] と押すごとに x 座標値 → y 座標値 → x 座標値 → …… と交互に表示します。

例 “ $y = x^2 - 3$ ”と“ $y = -x + 2$ ”の2つのグラフの交点を求める。

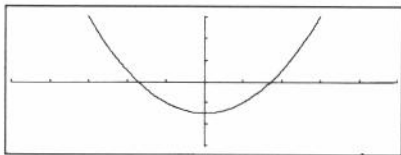
レンジは次の内容で行ないます。

X Range
min: -5.
max: 5.
scl: 1.

Y Range
min: -6.
max: 6.
scl: 2.

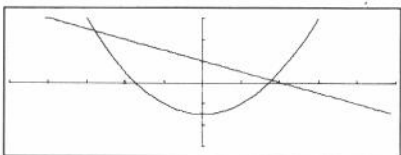
まず、“ $y = x^2 - 3$ ”のグラフを描きます。

Graph ALPHA X x^2 - 3 EXE



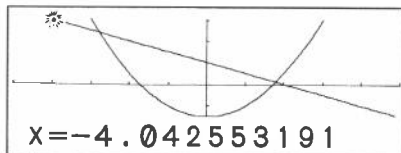
次に“ $y = -x + 2$ ”のグラフを描きます。

Graph (-) ALPHA X + 2 EXE



では、この直線グラフをトレースしてみます。

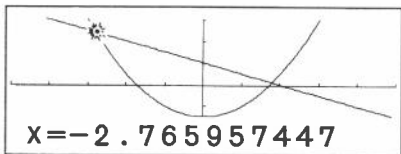
SHIFT Trace



↑ x 座標値を表示

グラフ上の左端で点滅している点がポインターで、 \leftarrow キーを押すとグラフにそって移動します。 \leftarrow キーは1回押すとポインターが1ポイントごとに進み、押し続けると連続して進みます。

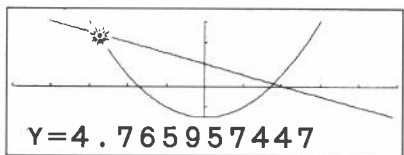
\leftarrow ~ (押し続ける)



ポインターを2つのグラフの交点に合うまで \leftarrow キーを押し続けます。

交点にポインターが合ったときの x 座標を読みます。y 座標の値は SHIFT X \leftrightarrow Y と押して求めます。

SHIFT X \leftrightarrow Y

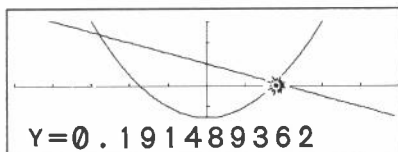


この結果、第1の交点の座標が $x = -2.765957447$ 、 $y = 4.765957447$ であることがわかります。

※ポインターは、 $\left[\rightarrow \right]$ キーまたは $\left[\leftarrow \right]$ キーを押すごとに任意の刻み幅で移動します。この刻み幅は、表示画面のドット上を移動しますので、一定値で移動はしません。よって、交点の $x \cdot y$ 座標値は概算値となります。

同様に、 $\left[\rightarrow \right]$ キーをポインターが次の交点に進むまで押します。

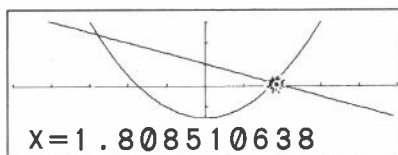
$\left[\rightarrow \right]$ ~



このときの x 座標値は $\left[\text{SHIFT} \right] \left[X \leftrightarrow Y \right]$ で表示されます。

$\left[\text{SHIFT} \right] \left[X \leftrightarrow Y \right]$ (座標値クリアー)

$\left[\text{SHIFT} \right] \left[X \leftrightarrow Y \right]$



表示窓の下に表示される座標値は、 $\left[\text{SHIFT} \right] \left[X \leftrightarrow Y \right]$ と押すごとに X 座標 \rightarrow Y 座標 \rightarrow 消去 \rightarrow X 座標 \rightarrow Y 座標……と切り換わります。

なお、座標値を消去しているときにトレース機能を実行しても、ポインターの点滅だけで座標値は表示されません。このときは、もう一度 $\left[\text{SHIFT} \right] \left[X \leftrightarrow Y \right]$ と押すと x 座標値を表示します。

このように、描いたグラフの上をトレースしていけば、そのときの $x \cdot y$ 座標値(概算値)がすぐにわかります。

※トレース機能は、グラフを描いた直後のみ実行でき、途中で計算や他の操作 ($\left[\text{M} \right] \left[\text{Disp} \right]$ $\left[\text{Range} \right]$ 、 $\left[\text{G} \rightarrow \text{T} \right]$ は除く)を行なった後では実行できません。

※最下段に表示される $x \cdot y$ 座標値は仮数部10桁または仮数部5桁、指数部2桁となります。

※トレース機能はプログラム中に書き込んで使うことはできません。

※“▲”により停止中、トレース機能を実行することができます。

■プロット (Plot) 機能

プロット機能とは、グラフィック表示画面の任意の座標に点を表示させます。この表示された点をカーソルキー(◀ ▶ ▲ ▼)で上下・左右に移動させ、表示されているグラフの座標値を読み取ったり、2点を表示させて直線で結んだり(ライン機能、71ページ)することができます。

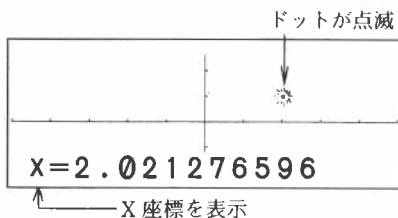
プロット機能は **SHIFT** **Plot** と押し、“Plot”の後に“ x 座標、 y 座標”を指定します。

例 次の座標中に、 $x = 2$ 、 $y = 2$ の点を打つ。

X Range
min: -5.
max: 5.
scl: 1.

Y Range
min: -5.
max: 5.
scl: 2.

SHIFT **Plot** 2 **SHIFT** 2 **EXE**

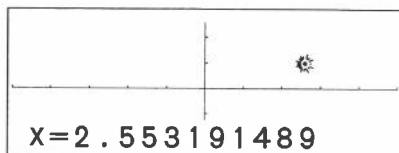


点滅しているポインタの位置が、指定された座標です。

※表示されている $x \cdot y$ 座標値は、表示画面のドット密度により指定された座標値の最も近い値を判断して表示しますので、指定値の近似値となります。

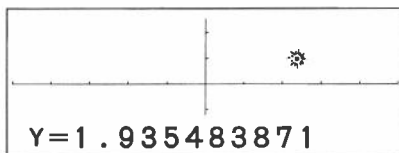
この点滅しているポインタは、カーソルキー(◀ ▶ ▲ ▼)により上下・左右に移動させて、座標値を読み取ることができます。

◀ ▶ ▶ ▶ ▶



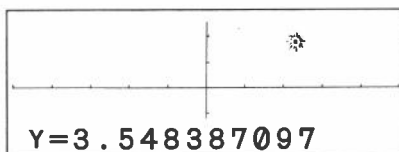
このときの y 座標値は？

SHIFT X \leftrightarrow Y



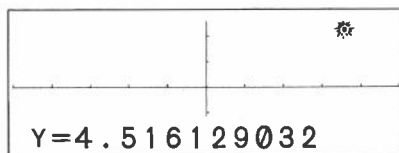
↑ ~ ↑

10回押す



続けてプロット機能により新しい座標値を入力すると、現在の点滅しているポインターを点灯させたまま、新たなポインターが点滅します。

SHIFT Plot 3.5 SHIFT □ 4.5 EXE



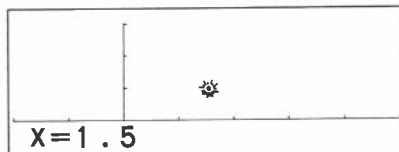
また、 $x \cdot y$ 座標を指定せずにプロット機能を実行すると、画面中央にポインターが点滅します。

レンジ内容は次のように設定します。

X Range
min : -2 .
max : 5 .
scl : 1 .

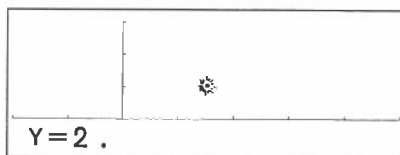
Y Range
min : -2 .
max : 6 .
scl : 2 .

SHIFT Plot EXE



このときの y 座標は？

SHIFT X \leftrightarrow Y



※レンジで設定された範囲外の点を描こうとしたときは、無視されてポインターは点滅しません。

※プロット機能により点滅しているポインターの $x \cdot y$ 座標は、XメモリーとYメモリーに記憶されます。

※プロット機能により点滅しているポインターは、次のプロットにより新たなポインターを点滅させたときには、点滅しない点として表示されます。

※表示窓の下に表示される座標値は、SHIFT X \leftrightarrow Y と押すごとに X座標 \rightarrow Y座標 \rightarrow 消去 \rightarrow X座標 \rightarrow Y座標……と切り換わります。

なお、座標値を消去しているときに新たなポインターをプロットしても、ポインターの点滅だけで座標値は表示されません。このときは、もう一度 SHIFT X \leftrightarrow Y と押すと x 座標値を表示します。

■ライン(Line)機能

ライン機能は、プロット機能により点灯された2点(点滅しているポインターも含む)を直線で結ぶ機能です。

この機能により、座標上のグラフに任意の線を描き入れて、グラフを見やすく、座標を読みやすくなります。

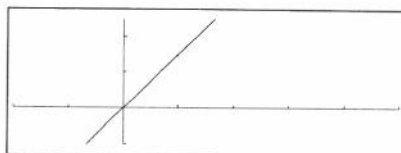
例 “ $y = 3x$ ”のグラフの $(x, y) = (1, 0)$ の点から x 軸・ y 軸に垂線を引く。
レンジは次のように設定します。

X Range
min : -2 .
max : 5 .
scl : 1 .

Y Range
min : -2 .
max : 5 .
scl : 2 .

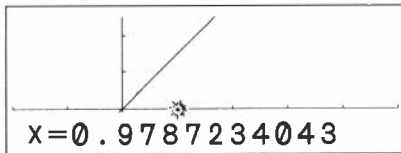
まず、グラフィック画面をクリアーしてから “ $y = 3x$ ”のグラフを描きます。

SHIFT Cls EXE
Graph 3 ALPHA X EXE



次に、 $x = 1$ 、 $y = 0$ の点をプロット機能で打ちます。

SHIFT Plot 1 SHIFT , 0 EXE

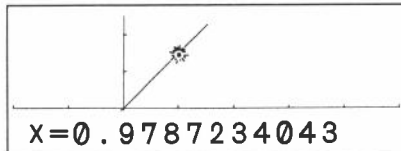


続けて、 $x = 1$ 、 $y = 0$ の点をもう一度打ち、カーソルキー(⇐)で“ $y = 3x$ ”のグラフと交わる点へポインターを移動させます。

SHIFT Plot 1 SHIFT , 0 EXE

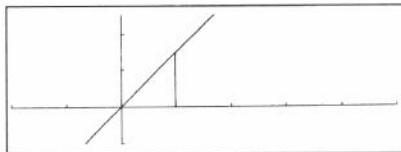
⇐ ~ ⇐

($y = 3x$ のグラフ上にポインターを移動)



ライン機能により線を引きます。

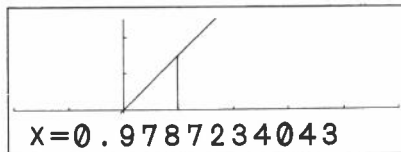
SHIFT Line EXE



次に、グラフ上の交点から y 軸上に垂線を引きますので、もう一度グラフ上の交点にポインターを打ち、カーソルキーで y 軸上に移動させて線を引きます。グラフ上の交点の $x \cdot y$ 座標はプロット機能により、X メモリーと Y メモリーに記憶されていますので、このまま“Plot X, Y”を実行します。

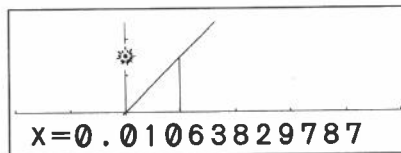
SHIFT Plot ALPHA X SHIFT , ALPHA Y

EXE

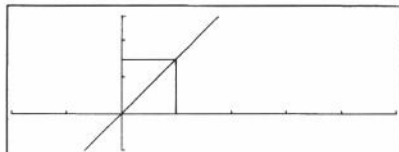


⇐ ~ ⇐

(y 軸上にポインターを移動)



SHIFT Line EXE



※ライン機能により引ける線は、プロット機能により点滅しているポインターと前回ポインターのあった点の間だけです。

■ファクター (Factor) 機能

ファクター機能は、プロット機能やトレース機能により点滅しているポインターを中心として、レンジ内容を拡大、縮小します。

拡大とは、レンジ内容の min 値と max 値を $\frac{1}{n}$ 倍にします。縮小とは、レンジ内容の min 値と max 値間の値を n 倍にします。

●使い方

SHIFT Factor m SHIFT \square n EXE ……ポインターを中心に、 x 方向を m 倍、 y 方向を n 倍に拡大します。

SHIFT Factor n EXE ……ポインターを中心に、 x 方向・ y 方向とも n 倍に拡大します。

なお、ファクター機能を実行すると、レンジ内容が変更されるため、グラフィック画面はクリアされます。

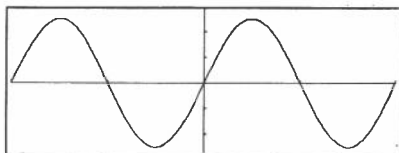
例 次のレンジ内容設定後、“ $y = \sin x$ ”のグラフを座標軸の交点を中心に1.5倍にする。

X Range
min : -360.
max : 360.
scl : 180.

Y Range
min : -1.
max : 1.
scl : 0.4

レンジ内容設定後、“ $y = \sin x$ ”のグラフを描いてみます。

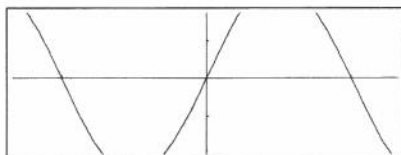
Graph sin ALPHA X EXE



このグラフを座標軸の交点を中心に拡大しますので、プロット機能でポインターを点滅させた後に、ファクター機能で1.5倍にします。

SHIFT Plot [] SHIFT Factor 1.5 []
Graph sin ALPHA X EXE

※マルチステートメント機能を使って一度にグラフを描きます。



このときのレンジ内容を見てみましょう。

```
X Range
  min: -240.
  max: 240.
  scl: 180.
```

```
Y Range
  min: -0.66666667
  max: 0.666666666
  scl: 0.4
```

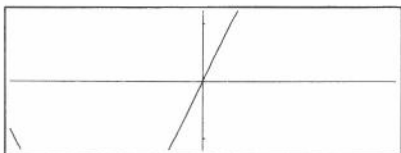
このように、 $x \cdot y$ 方向の範囲が $\frac{1}{1.5}$ になっています。

もう一度グラフを1.5倍にしてみましょう。

今度は新たに命令を入力する必要はなく、ただEXEキーを押すだけでかまいません。

これは、前回の操作をマルチステートメント機能で描いているので、今度はリプレイ機能が使えます。

EXE



これでは少し大きすぎて、画面からだいぶはみ出しています。

今度は縮小してみましょう。

現在表示されているグラフを半分の大きさ($\frac{1}{2}$ 倍)にしてみます。

半分の大きさの指定は、リプレイ機能を利用して1.5倍を、0.5倍に訂正します。

⇐

```
Plot : Factor 1.5
: Graph Y=sin X
```

⇒ ⇒ ⇒

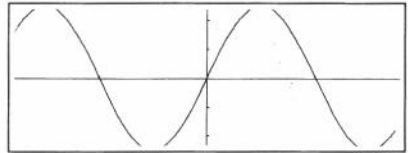
Plot : Factor 1.5
: Graph Y=sin X

0

Plot : Factor 0.5
: Graph Y=sin X

これで半分になる指定が終わりました。
では、実行してみましょう。

EXE



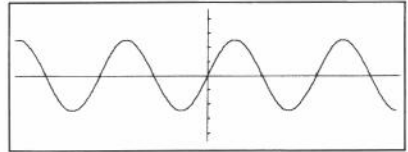
このときのレンジ内容を見てみましょう。

X Range
min : -320.
max : 320.
scl : 180.

Y Range
min : -0.88888889
max : 0.88888889
scl : 0.4

続けて、もう半分にしてみます。

EXE



今度は、 x 座標を2倍に、 y 座標を1.5倍にしてみます。

⇒

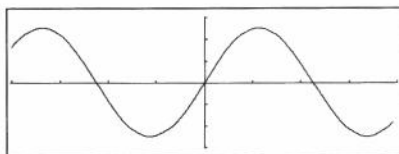
Plot : Factor 0.5
: Graph Y=sin X

⇨ ⇨ ⇨ [SHIFT] [INS]
2 [SHIFT] [↓]
[SHIFT] [INS] 1

Plot : Factor 2, 1
_5: Graph $Y = \sin X$

では、実行してみます。

[EXE]



このように、ファクター機能を使えば、グラフを拡大したり縮小したりすることができます。

なお、この例では画面の中央を中心にして拡大・縮小を行いましたが、ポインターを任意の所に点滅させれば、そのポインターを中心に拡大・縮小ができます。

また、ポインターを出現させないでファクター機能を実行すると、自動的に画面の中心を基準として拡大・縮小することができます。

3-3 グラフ関数の応用

複雑な方程式も、グラフ化することにより、式を目で見ることができます。

ここでは、複雑な方程式をグラフ化してみましょう。

例1) $y = x^3 - 9x^2 + 27x + 50$ の3次方程式をグラフにする。

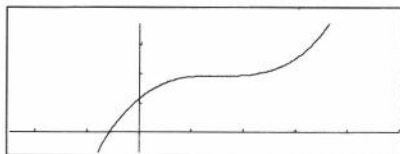
レンジ内容は次のように設定します。

X Range
min: -5.
max: 10.
scl: 2.

Y Range
min: -30.
max: 150.
scl: 40.

操 作

SHIFT C|s EXE
Graph ALPHA X x^y 3 □ 9 ALPHA X
 x^2 + 27 ALPHA X + 50 EXE



例2) $y = x^6 + 4x^5 - 54x^4 - 160x^3 + 641x^2 + 828x - 1260$ の6次方程式をグラフにする。

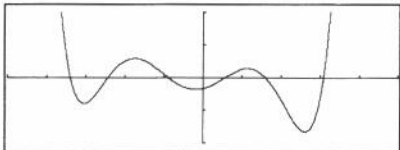
レンジ内容は次のように設定します。

X Range
min: -10.
max: 10.
scl: 2.

Y Range
min: -8000.
max: 8000.
scl: 4000.

操 作

SHIFT C|s EXE
Graph ALPHA X x^y 6 + 4 ALPHA X
 x^y 5 □ 54 ALPHA X x^y 4 □ 160
ALPHA X x^y 3 + 641 ALPHA X x^2
+ 828 ALPHA X □ 1260 EXE



例3) $y = x^4 + 4x^3 - 36x^2 - 160x + 300$ の方程式の極大・極小を求める。

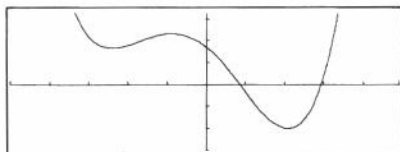
※グラフ化すれば、式を微分し、増減表を作らなくてもすぐわかります。

レンジ内容は次のように設定します。

X Range
min: -10.
max: 10.
scl: 2.

Y Range
min: -600.
max: 600.
scl: 200.

SHIFT Cls EXE
 Graph ALPHA X x^y 4 + 4 ALPHA X
 x^y 3 = 36 ALPHA X x^2 = 160
 ALPHA X + 300 EXE



例4) $y = x^3 - 3x^2 - 6x - 16$ のグラフと $y = 3x - 11$ のグラフが接点を持つかを見る。

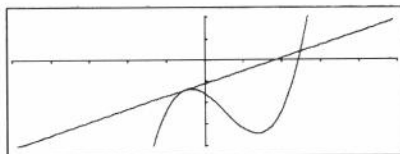
レンジは次のように設定します。

X Range
min: -10.
max: 10.
scl: 2.

Y Range
min: -40.
max: 20.
scl: 10.

操作

SHIFT Cls EXE
 Graph ALPHA X x^y 3 = 3 ALPHA X
 x^2 = 6 ALPHA X = 16 EXE
 Graph 3 ALPHA X = 11 EXE



※この例のような場合、トレース機能により接点を求め、正確な接点の座標を求めて確認した方が、より正確な接線となります。

3-4 1 変数統計グラフ

- 1変数統計グラフは **[SHIFT] [MODE] [X]** と押す SD 2 モードで描きます。
 - 1変数統計グラフとして、棒グラフ、折れ線グラフ、正規分布曲線グラフを描くことができます。
 - SD 2 モードでも一般の関数グラフを描くことができますので、予測線と実測線等の重ね描きができます。
- ※SD 2 モードでは **Abs**、 $\sqrt{\quad}$ は使えません。
- 1変数統計グラフは設定されたレンジの範囲内を増設されたメモリー (**[MODE] [□]** : Defm) 数で分割して、データ数を数えます。
 - グラフは x 軸をデータの範囲とし、 y 軸を各々のデータの個数(度数)として描かれます。
 - 棒グラフでは、棒の幅を以上～未満で表わします。
 - データの入力は **[DT]** キー (**[$\sqrt{\quad}$]**) を使い、データの個数(度数)は $Z[1]$ ~ 増設個数の中に累計されます。
 - **[CL]** キー (**[\rightarrow]**) によるデータの訂正ができます。

■ 1 変数統計グラフの描き方

● 手順

- ① SD 2 モードを指定する。 (**[SHIFT] [MODE] [X]**)
- ② レンジ内容を設定する。 (**[Range]**)
- ③ グラフの本数(分割数)に合わせて、メモリーを増設する。 (**[MODE] [□] n [EXE]**)
- ④ 集計用メモリークリアー。 (**[SHIFT] [Sci] [EXE]**)
- ⑤ データ入力(データ **[SHIFT] [↓]** 度数 **[DT]**)
- ⑥ グラフを描く。
 - 棒グラフ…………… **[Graph] [EXE]**
 - 折れ線グラフ…………… **[Graph] [SHIFT] [Line] [EXE]**
 - 正規分布曲線…………… **[Graph] [SHIFT] [Line] 1 [EXE]**

※⑤のデータ入力の仕方は、標準偏差計算(48ページ参照)と同じです。

例 次のデータにより、階級別のグラフを描く。

階級 No.	階級	度数
1	0	1
2	10	3
3	20	2
4	30	2
5	40	3
6	50	5
7	60	6
8	70	8
9	80	15
10	90	9
11	100	2

手順に従い、順番に操作します。

①レンジ内容を設定する。

レンジ内容は、 x 軸方向が0～100ですが、100の値のグラフを描くために、
最大値(Xmax)は110とします。(以上～未満ですので、0～110となります)
 y 軸方向は、最大値(度数)が15ですので、Ymaxを20とします。

```
X Range
min: 0.
max: 110.
scl: 10.
```

```
Y Range
min: 0.
max: 20.
scl: 2.
```

②グラフの本数は0～9、10～19、20～24、……100～109ですので11本となり、
メモリーを11個増設します。

MODE \square 11 EXE

```
**Defm**
Program : 64
Memory : 37
398 Bytes Free
```

③集計用メモリーをクリアーします。

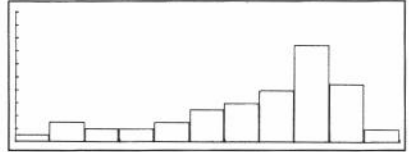
SHIFT Scl EXE

④データを入力します。

0 [DT] 10 [DT] [DT] [DT] 20 [DT] [DT] 30 [DT] [DT] 40 [DT] [DT] [DT]
50 [SHIFT] [↓] 5 [DT] 60 [SHIFT] [↓] 6 [DT] 70 [SHIFT] [↓] 8 [DT]
80 [SHIFT] [↓] 15 [DT] 90 [SHIFT] [↓] 9 [DT] 100 [DT] [DT]

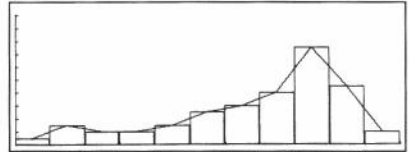
⑤まず棒グラフを描いてみます。

[Graph] [EXE]



次に折れ線グラフを重ね描きしてみましょう。

[Graph] [SHIFT] [Line] [EXE]



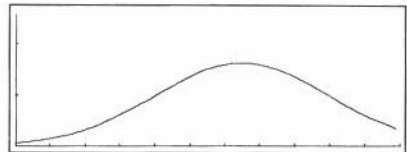
- 最後に正規分布曲線を描きますが、正規分布曲線の y 値は棒グラフや折れ線グラフに比べて、かなり小さいため、同じレンジ範囲では表示しきれません。レンジ内容を次のように変更してください。

X Range
min: 0.
max: 110.
scl: 10.

Y Range
min: 0.
max: 0.025
scl: 0.01

[Graph] [SHIFT] [Line] 1 [EXE]

↑
数字の1を入れると正規分布曲線を描きます。



〈まとめ〉

- データの分割数(棒グラフの本数に相当)に応じたメモリーの増設(**MODE** □ n **EXE**)は必ず行なってください。増設せずにグラフを描こうとするとエラー(Mem ERROR)になります。
- データ入力中にメモリーの増設数を変更すると、データの分割が変化して、正しいグラフが描けませんので、データ入力中はメモリーの増設数を変えないでください。
- レンジの範囲をこえた値が入力されたときは、統計用メモリーは入力されますが、グラフ用のメモリーには入力されません。
- レンジの y 軸方向の範囲より多くのデータが入力された場合、棒グラフでは表示画面上限まで描き、折れ線グラフでは表示画面外となる点を結びません。
- 正規分布曲線の式は、 $y = \frac{1}{\sqrt{2\pi} x \sigma n} e^{-\frac{(x-\mu)^2}{2x\sigma n^2}}$ を使っています。
- 棒グラフや折れ線グラフ実行後、テキストエリアでは、“done”を表示します。

3-5 2変数統計グラフ

- 2変数統計グラフは **SHIFT** **MODE** **+** と押す LR 2 モードで描きます。
- 2変数統計グラフとして、回帰直線を描くことができます。
- LR 2 モードでも一般の関数グラフを描くことができますので、理論線とデータ分布、回帰直線の重ね描きができます。
- LR 2 モードでは、データ入力後、即時に点を表示させ、統計用メモリーにデータを入力します。
- レンジ範囲をこえるデータが入力された場合は、統計用メモリーにはデータ入力されますが、点は表示されません。
- データの入力は **DT** キー (**☐**) を使い、 x データ **SHIFT** **☐** y データ **SHIFT** **☐** 度数 **DT** と入力します。
- データ入力後、**CL** キー (**☐**) によるデータ訂正ができますが、表示された点は消えません。(データ訂正に関係なく、**CL** キーでデータ入力しても点を打ちます)
- 画面クリアー (**SHIFT** **Cls** **EXE**) 後は、再び点を表示させることはできません。

■ 2変数統計グラフの描き方

● 手順

- ① LR 2 モードを指定する。 (**SHIFT** **MODE** **+**)
- ② レンジを設定する。 (**Range**)
- ③ 統計用メモリーをクリアーする。 (**SHIFT** **Sci** **EXE**)
- ④ データ入力。 (x データ **SHIFT** **☐** y データ **SHIFT** **☐** 度数 **DT**)
- ⑤ グラフを描く。 (**Graph** **SHIFT** **Line** 1 **EXE**)

※④のデータ入力の仕方は、回帰計算(50ページ参照)と同じです。

例 次のデータを直線回帰して、回帰直線を描く。

x_i	y_i
-9	-2
-5	-1
-3	2
1	3
4	5
7	8

①レンジ内容を次のように設定する。

X Range
min: -10.
max: 10.
scl: 2.

Y Range
min: -5.
max: 15.
scl: 5.

※レンジ内容は、 x 軸方向について「以上～未満」となりますので、-10以上、10未満となります。

②統計用メモリーをクリアーします。

SHIFT **ScI** **EXE**

④データを入力します。

(-) 9 **SHIFT** **□** **(-)** 2 **DT**

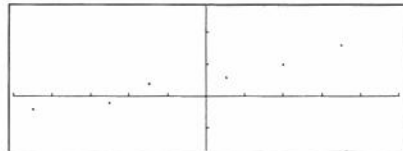
(-) 5 **SHIFT** **□** **(-)** 1 **DT**

(-) 3 **SHIFT** **□** 2 **DT**

1 **SHIFT** **□** 3 **DT**

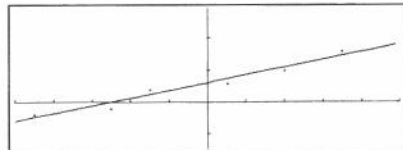
4 **SHIFT** **□** 5 **DT**

7 **SHIFT** **□** 8 **DT**



⑤回帰直線を描きます。

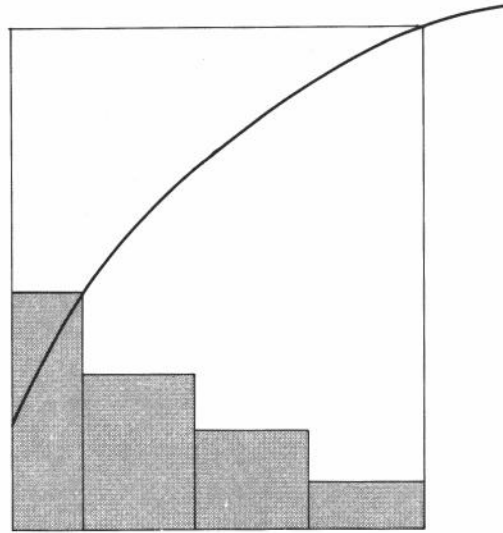
Graph **SHIFT** **Line** 1 **EXE**



※レンジ範囲外のデータを入力したときは、点は表示せず、グラフィック表示になりません。

※データ未入力時に **Graph** **SHIFT** **Line** 1 **EXE** と操作すると、Ma ERROR となります。

第4章 プログラム計算

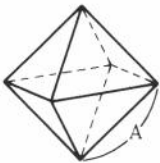


4-1 プログラムとは？

本機には繰り返し計算に便利なプログラム機能が内蔵されています。このプログラム機能はマニュアル計算における「マルチステートメント機能」と同様に、式を連続して実行する機能です。

まずは例題とともにプログラムを見てみましょう。

例 一辺の長さを与えて、図のような正8面体の表面積と体積を求める。



一辺の長さ(A)	表面積(S)	体積(V)
10cm	() cm ²	() cm ³
7	()	()
15	()	()

※カッコ内を求める。

①計算式を求める

表面積を S、体積を V、一辺の長さを A とすると、正8面体の S と V は…

$$S = 2\sqrt{3}A^2 \quad V = \frac{\sqrt{2}}{3}A^3$$

②プログラミング

計算式をもとにしてプログラムを作ることを「プログラミング」と呼びます。ここでは上の計算式をもとにプログラムを作ってみましょう。

プログラムの基本はマニュアル計算ですので、まず、マニュアルで計算するときの操作方法を考えてみましょう。

表面積：2 [√] 3 [A の数値] [x²] [EXE] …… S

体積：[√] 2 [÷] 3 [A の数値] [x³] 3 [EXE] …… V

ここで「A の数値」は2回使用していますので、A メモリーにあらかじめ記憶させておけば、合理的になります。

A の数値 → [ALPHA] [A] [EXE]

2 [√] 3 [ALPHA] [A] [x²] [EXE] …… S

[√] 2 [÷] 3 [ALPHA] [A] [x³] 3 [EXE] …… V

本機のプログラムは、このマニュアル計算で行なう操作をそのままプログラムにします。なお、プログラムはスタートさせると順番に命令を実行して止まりませんので、データを入力したり、答えを表示させる命令が必要です。このための命令が入力命令の「？」と出力命令の「▲」です。

「？」はプログラムの実行を一時停止させ、「？」を表示して入力を待ちます。この命令は単独では使わずに「 \rightarrow 」メモリー名」といっしょに使います。

たとえば入力された数値を A メモリーに記憶させるには、

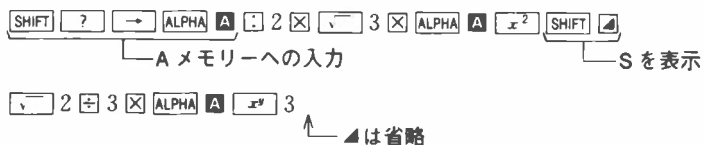
？ \rightarrow A

とします。

「？」が表示されているときは、111ステップまでの計算命令や数値が入力できます。

出力命令の「▲」はプログラムの実行を一時停止させ、直前の計算式の答えやアルファ文字(124ページ参照)を表示させます。この命令はマニュアル計算で [EXE] キーを押して答えを求める箇所に入れます。なお、プログラムの一番最後は自動的にプログラムが終了し、それまでの式の答えを表示させますので、省略してもかまいません。ただし、プログラム中で Base-n モードにして、進数を変換するようなときは、最後の「▲」は省略しないでください。

それでは前記の操作手順に入力命令と出力命令を加えてみましょう。



これでプログラムが完成しました。

本機のプログラミングはマニュアル計算の操作方法をもとにして、入力命令や出力命令を加えるという方法で行ないます。

③プログラムの記憶

プログラムの記憶(書き込みとも呼びます)は **MODE** ② と押して WRT モードで行ないます。

操 作

表 示

MODE ②

WRT	:	COMP
Deg	:	Norm
486	Bytes Free	
Prog	0	1 2 3 4 5 6 7 8 9

MODE ② と押すと動作モードが WRT にかわり、WRT モードであることを示します。そして3桁以内の数字は残りのステップ数(101ページ参照)を示します。この残りステップ数はプログラムを記憶させていったり、メモリーを増設したりしていきますと減ってきます。プログラムも記憶させずに、メモリーも初期状態の26メモリーのときは486となっているはずです。

最下段の数字はプログラムエリア(103ページ参照)を示しています。“P”に続いて0～9まで表示されていれば、P0からP9までのエリアにプログラムが記憶されていません。そして、点滅している“0”にプログラムエリアが指定されています。

もし、プログラムがいずれかのエリアに記憶されていれば、

WRT	:	COMP
Deg	:	Norm
312	Bytes Free	
Prog	0	1 _ 3 4 _ 6 7 8 9

のように“_”が表示されます。

ここでは点滅しているP0に前記のプログラムを記憶させてみましょう。

(P0以外のプログラムエリアに記憶させるときは **⇐** **⇒** キーにより点滅をプログラムエリア番号に合わせます)

操 作

表 示

EXE (記憶開始)

—

SHIFT **?** **→** **ALPHA** **A** **:**

?→A : 2×√3×A² ▽

2 **×** **√** **3** **×** **ALPHA** **A**

—

x² **SHIFT** **▽**

√ **2** **÷** **3** **×** **ALPHA** **A**

?→A : 2×√3×A² ▽

x^y **3**

√2÷3×A x^y 3_

これでプログラムの記憶は終了しました。

☆WRT モードでのキー操作はプログラムとしての書き込み(記憶)となります。

☆現在のプログラムエリアのステップ数や計算モードなどの状態は **M Dsp** キーを押している間だけ表示されます。

M Dsp (押している間、
表示)

```
*** MODE ***
WRT   :   COMP
Deg   :   Norm
Step  P0-20
```

☆プログラムの記憶が終了したら、**MODE** **1** と押して RUN モードに戻しておきます。

④プログラムの実行

プログラムの実行は **MODE** **1** と押す RUN モードで行ないます。

プログラムの実行は **Prog** キーを使い、プログラムエリアの指定を行ないます。

P 0 実行… **Prog** **0** **EXE**

P 3 実行… **Prog** **3** **EXE**

P 8 実行… **Prog** **8** **EXE**

前記の例題を実行してみましょう。

例題は正8面体の表面積(S)と体積(V)を求める計算でした。

一辺の長さ(A)	表面積(S)	体積(V)
10cm	(346.4101615)cm ²	(471.4045208)cm ³
7	(169.7409791)	(161.6917506)
15	(779.4228634)	(1590.990258)

操 作

表 示

MODE 1

```

*** MODE ***
  RUN   :   COMP
  Deg   :   Norm
  Step  :   0
    
```

RUNモードに
なっているとき
は必要なし

Prog 0 EXE

```

?→A:2X√3XA^2▲
√2÷3XA^y3
Prog 0
?
    
```

前回の操作内容
を表示

10 EXE (Aの値)

```

?
10
      346.4101615
      - Disp -
    
```

(A=10のS)
▲により答えを
表示中の意味

EXE

```

?
10
      346.4101615
      471.4045208
    
```

(A=10のV)

Prog 0 EXE

```

      346.4101615
      471.4045208
Prog 0
?
    
```

7 [EXE] (A の値)

```
?  
7  
169.7409791  
- Disp -
```

(A=7のS)

[EXE]

```
?  
7  
169.7409791  
161.6917506
```

(A=7のV)

[Prog] 0 [EXE]

```
169.7409791  
161.6917506  
Prog 0  
?
```

15 [EXE] (A の値)

```
?  
15  
779.4228634  
- Disp -
```

[EXE]

```
?  
15  
779.4228634  
1590.990258
```

☆プログラム計算は、《データ [EXE]》または《答え読み取り後 [EXE]》のように [EXE] キーを押すごとに自動的に計算します。

☆この例のように、“[Prog] 0 [EXE]”としてプログラムを実行した直後は、“Prog 0”の命令がリプレイ機能により記憶されていますので、2回目以後は [EXE] キーを押すだけでも、PO のプログラムが再び実行されます。

操 作

[Prog] 0 [EXE] (PO のプログラム実行)

10 [EXE] (A=10を入力)

[EXE] (A=10の V を表示)

[EXE] (再び実行)

7 [EXE] (A=7を入力)

[EXE] (A=7の V を表示)

⋮

4-2 プログラムのチェックおよび編集 (訂正・追加・削除)

プログラムのチェックとは、記憶されているプログラムを呼び戻して、内容を確認することです。

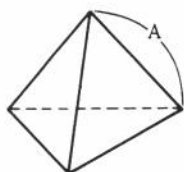
これはWRTモード(MODE 2と押す)にして、 $\square \leftarrow$ 、 $\square \rightarrow$ キーによりプログラムエリアを指定し、EXEキーを押します。プログラム内容が表示されますので、 $\square \rightarrow$ キー(または $\square \leftarrow$ 、 $\square \uparrow$ 、 $\square \downarrow$)を押して順に1ステップずつ送って確認します。

プログラムの編集とは、記憶されているプログラムの内容を訂正したり(誤って記憶させてしまったときなど)、変更を加えるために追加したり削除したりすることです。

では、例題にそってチェックと編集を行なってみましょう。

例

一辺の長さを与えて、図のような正4面体の表面積と体積を求める。



一辺の長さ(A)	表面積(S)	体積(V)
10 cm	() cm ²	() cm ³
7.5	()	()
20	()	()

①計算式を求める

表面積をS、体積をV、一辺の長さをAとすると、正4面体のSとVは…

$$S = \sqrt{3} A^2 \quad V = \frac{\sqrt{2}}{12} A^3$$

②プログラミング

前例と同じように一辺の長さをAメモリーに記憶させてプログラムを組んでみましょう。プログラムの方法は、マニュアル計算の操作を基本とします。

Aの数値→ ALPHA A EXE

$\sqrt{\square}$ 3 \times ALPHA A $\square x^2$ EXE …………… S

$\sqrt{\square}$ 2 \div 12 \times ALPHA A $\square x^3$ 3 EXE …………… V

これをプログラムにすると次のようになります。

SHIFT ? → ALPHA A : √ 3 X ALPHA A x² SHIFT ▾
 √ 2 + 12 X ALPHA A x^y 3

③プログラムの記憶(編集)

このプログラムを記憶させますが、プログラムエリアをかえて新たに記憶させても良いのですが、このプログラムは前例の正8面体と似ていますので、変更を加えながらプログラムを記憶させてみましょう。

まず、2つのプログラムを比較してみましょう。

正8面体: SHIFT ? → ALPHA A : 2 X √ 3 X ALPHA A x² SHIFT ▾
 √ 2 + 3 X ALPHA A x^y 3

正4面体: SHIFT ? → ALPHA A : √ 3 X ALPHA A x² SHIFT ▾
 √ 2 + 12 X ALPHA A x^y 3

比較してみると正8面体のプログラムから正4面体のプログラムへの変更は~~~~の部分削除して、___の部分訂正すればできます。

では、実際に行なってみましょう。

操作

表示

MODE ②

```
WRT   :   COMP
Deg   :   Norm
466   Bytes Free
Prog  _123456789
```

EXE

```
?→A: 2X√3XA2 ▾
√2÷3XAy3
```

カーソルが先頭に位置します。

⇐ ⇐ ⇐ ⇐

DEL DEL

```
?→A: √3XA2 ▾
√2÷3XAy3
```

削除する位置にカーソルを合わせて、2文字分削除します。

⇓ ⇐ SHIFT

INS 12

```
?→A: √3XA2 ▾
√2÷12[3]AXy3
```

2文字分挿入します。

DEL

```
?→A: √3XA2 ▾
√2÷12[X]Ay3
```

不必要な“3”を削除します。

MODE 1

```
*** MODE ***
RUN   :   COMP
Deg   :   Norm
Step  :   0
```

④プログラムの実行

では、この例題を実行してみましょう。

一辺の長さ(A)	表面積(S)	体積(V)
10 cm	(173.2050808)cm ²	(117.8511302)cm ³
7.5	(97.42785793)	(49.71844555)
20	(692.820323)	(942.8090416)

操 作

表 示

MODE 1

```
*** MODE ***
RUN   :   COMP
Deg   :   Norm
Step  :   0
```

Prog 0 EXE

```
?→A:√3×A2
√2÷12×A3
Prog 0
?
```

10 EXE

```
?
10
173.2050808
- Disp -
```

EXE

```
?
10
173.2050808
117.8511302
```


Prog 0 EXE

```
173.2050808
117.8511302
Prog 0
?
```

7.5 EXE

```
?
7.5
97.42785793
- Disp -
```

EXE

```
?
7.5
97.42785793
49.71844555
```

Prog 0 EXE

```
97.42785793
49.71844555
Prog 0
?
```

20 EXE

```
?
20
692.820323
- Disp -
```

EXE

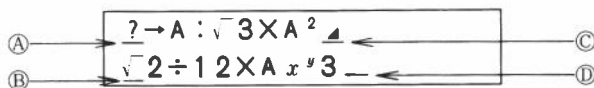
```
?
20
692.820323
942.8090416
```

〈まとめ〉

	操 作 内 容	使 う キ ー
プログラム	<ul style="list-style-type: none"> • WRT モードの指定 • プログラムエリア指定 (P0の場合は省略) • チェック開始 • 内容のチェック 	MODE 2 ← → EXE ← → ↑ ↓
訂 正	<ul style="list-style-type: none"> • 訂正する箇所にカーソルを合わせる • 正しくキー操作をする 	← → ↑ ↓
削 除	<ul style="list-style-type: none"> • 削除する箇所にカーソルを合わせる • 削除する 	← → ↑ ↓ DEL
挿 入	<ul style="list-style-type: none"> • 挿入する箇所の右にカーソルを合わせる • 挿入モードにする • 挿入したいキーを押す 	← → ↑ ↓ SHIFT INS

〈参 考〉 カーソルの動かし方

カーソルはカーソルキー (← → ↑ ↓) を押すことにより次のように動きます。



カーソルの位置	←	→	↑	↓
①の場合	無効	一つ右へ	無効	一つ下へ(②)
②の場合	一つ左へ(③)	一つ右へ	一つ上へ(①)	行末へ(④)
③の場合	一つ左へ	一つ右へ(②)	行頭へ	一つ下へ
④の場合	一つ左へ	無効	一つ上へ(③)	無効

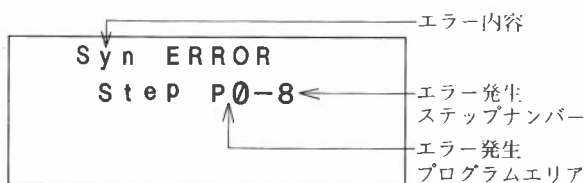
4-3 プログラムのデバッグ(まちがいを直す)

プログラムを作り上げ、記憶させた後に実行しても、エラーが表示されたり結果(答え)が思うように得られないことがあります。このようなときはプログラム中にまちがいがあるはずですので、そのまちがいを直して正しいプログラムとすることが必要です。

プログラム中にある「まちがい」のことを「バグ」(虫の意味)と呼び、この虫を取り除くという意味でプログラムの訂正を「デバッグ」といいます。

■エラー表示によるデバッグ

エラー表示とは、エラーメッセージとも呼ばれ、次のように表示されます。



エラー発生プログラムエリアは、P0～P9のエリアの内どこでエラーがおきたのかを知らせてくれます。

エラー内容はどのようなエラーがおきたのか知らせてくれます。このエラー内容によりどう対処すればよいかわかります。

エラー発生ステップナンバーはエラーの発生したプログラムエリア内の何ステップ目でエラーがおきたのかを知らせてくれます。

エラーメッセージは全部で7つあります。

■エラーメッセージ

① Syn ERROR：文法エラー

計算式にあやまりがある場合や、プログラム用の命令の使い方をまちがえた場合に表示されます。

② Ma ERROR：数学的エラー

数式の演算結果が 10^{100} 以上であったり、演算が不能であるような場合(例：0

による除算)。また、関数の入力範囲をこえる引数を入力した場合に表示されます。

③ Go ERROR：飛び先エラー

Goto 命令(108ページ参照)に対する Lbl がない場合や Prog 命令(114ページ参照)に対するプログラムエリア(103ページ参照)にプログラムが記憶されていない場合に表示されます。

④ Ne ERROR：ネスティングエラー

Prog 命令によるサブルーチンのネスティングがオーバーしたとき(115ページ参照)に表示されます。

⑤ Stk ERROR：スタックエラー

数値用スタックや命令用スタック(16ページ参照)をこえて演算が行なわれる場合に表示されます。

⑥ Mem ERROR：メモリーエラー

メモリーの増設を行っていないのに Z [5] のようなメモリーを使おうとした場合に表示されます。

⑦ Arg ERROR：引数エラー

プログラム用の命令や指定を行なうときの引数が入力範囲外の場合、表示されます。

(例：Sci10、Goto11の場合など)

エラーメッセージが表示されるとキー操作ができなくなります。この状態を解除するには **AC** キーまたは **←**・**→** キーを押します。

AC キーを押すとエラー状態が解除され、新たなキー入力ができます。このときのモードは RUN モードのままです。

← キーまたは **→** キーを押すと、エラー状態が解除されるとともに、動作モードを WRT モードに切り替え、エラーの発生した箇所でカーソルが点滅しますので、訂正をすることができます。

■エラーメッセージのチェックポイント

エラーメッセージの内容は以上の通りですが、エラーメッセージに合わせたチェックポイントをあげてみましょう。

① Syn ERROR

記憶させたプログラムにまちがいがいないか、プログラムのチェックに従いもう一度見なおします。

② Ma ERROR

メモリーを使って計算を行なう場合に、メモリー内の数値が関数の引数をこえていないかをチェックします。特に、0による除算や負数の平方根を求めようとしているような場合が多い。

③ Go ERROR

Goto n としたときの n に対する Lbl n があるかをチェックします。また、Prog n としたときの P_n 内にプログラムが正しく記憶されているかをチェックします。

④ Ne ERROR

Prog 命令により分岐したプログラムエリアから再び Prog 命令により、もとのプログラムエリアに分岐させていないかをチェックします。

⑤ Stk ERROR

計算式が長すぎて、スタックオーバーしていないかをチェックし、計算式を分割するようにします。

⑥ Mem ERROR

MODE **□** n (Defm)により正しくメモリーを増設しているかをチェックします。配列的使い方(121ページ参照)をしているときに、添字になるメモリーの数値が正しいかをチェックします。

⑦ Arg ERROR

MODE **7** (Sic)、**MODE** **8** (Fix)の指定値が0～9以内かチェックする。

Goto、Lbl、Prog の指定値が0～9以内かチェックする。

MODE **□** (Defm)によるメモリーの増設数が残りステップ数以内か、負数でないかをチェックする。

4-4 ステップ数のかぞえ方

本機の持っているプログラム容量は、全部で486ステップあります。このステップ数とはプログラムを記憶できる許容量を示す単位で、プログラムを記憶させていくと残りステップ数が減っていきます。

また、メモリーの増設により、ステップ数をメモリーに変換したときは486ステップより減っています。

プログラムを記憶していき、残りステップが5ステップ以下になりますとカーソルが“_”から“■”になります。5ステップ以上のプログラムを記憶させたい場合は他の不要なプログラムを消去するか、メモリーを減らしてプログラムステップを増やしてください。

現在の残りステップ数の確認には2つの方法があります。

- ① RUN モードで **MODE** と **EXE** と操作してメモリー数を確認すれば、同時に残りステップ数を表示します。

MODE と **EXE**

```
**Defm**
Program : 19
Memory  : 26
467 Bytes Free
```

← プログラム使用済ステップ数
← メモリー数
← 残りステップ数 (Base-nで表示)

- ② WRT モード (**MODE** と **2** と押す) にして、残りステップ数を確認する。このときはプログラムエリアの使用状況も確認できます。

MODE と **2**

```
WRT : COMP
Deg : Norm
467 Bytes Free
Prog _123456789
```

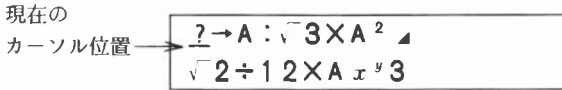
← 残りステップ数

ステップ数は原則として「1機能1ステップ」ですが、「1機能2ステップ」の命令もあります。

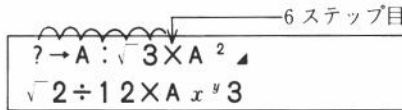
- 1機能1ステップ：sin、cos、tan、log、(、)、:、A、B、1、2、3等
- 1機能2ステップ：Lbl 1、Goto 2、Prog 8等

このステップ数はカーソルを動かすことにより、1ステップずつの確認ができます。

例

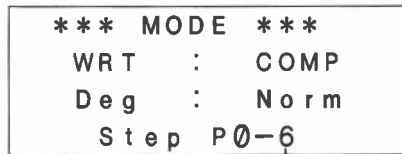


このようなときにカーソルキー(←) (→)を1回押して移動する範囲が1ステップです。← キーを押すとカーソルは次のように動きます。



現在のカーソル位置が何ステップ目であるかは、 [Disp] キーを押している間だけ表示されます。

[Disp] ~



↑
6ステップ目を示す

4-5 プログラムエリアと計算モード

本機にはプログラムを記憶させるプログラムエリアが10個あり、P0、P1、P2……P9と呼ばれています。

このプログラムエリアは全て同様な使い方ができ、独立したプログラムを10個記憶させたり、1個の主プログラム(メインルーチン)に対していくつかの副プログラム(サブルーチン)を組んで記憶させることができます。

なお、プログラムエリアに記憶できるステップ数はP0～P9までの合計で最大422ステップです。

このプログラムエリアの指定は次のように行ないます。

RUN モード：**Prog** キーに続き **0**～**9** のキーを押し、**EXE** キーを押す。

例 P0 **Prog** **0** **EXE**

P8 **Prog** **8** **EXE**

※RUN モードでは **EXE** キーを押すと同時にプログラムがスタートします。

WRT モード：WRT モードでは“P0 1 2 3 4 5 6 7 8 9”と表示されているときに、カーソルを **←**、**→** キーを移動させて指定したいプログラムエリアに合わせ、**EXE** キーを押します。

WRT モードでは **MODE** **2** と押したときにプログラムエリアの番号が表示されます。このとき0から9までの数字が表示されますが、数字の表示されているエリアにはまだプログラムが記憶されていません。もし“_”が表示されていれば、その位置のプログラムエリアにプログラムが記憶されています。

例

WRT	:	COMP
Deg	:	Norm
403	Bytes	Free
Prog	_123_	_67_9

すでにプログラムが記憶されています。

■ WRT モードでのプログラムエリア指定と計算モード(Comp、Base-n、SD、LR) 指定

通常の間数計算だけでなく、2進・8進・10進・16進数の変換や計算、標準偏差計算、回帰計算を行なうには計算モードを切り替えなければなりません。この計算モードの指定はプログラムエリアの指定と同時に進めます。

操作手順としては WRT モードにした後 (MODE 2) と押す)、計算モードを指定します。次に、プログラムエリアを指定し、[EXE] キーを押せば、プログラムエリアに計算モードも記憶されます。以後、記憶されるプログラムはこの計算モードに従います。

例 Base-n モードを P2 に記憶する

MODE 2

```

WRT   :   COMP
Deg   :   Norm
486   Bytes Free
Prog  0123456789
    
```

何も記憶していないとします。計算モードは COMP モードです。

⇐ ⇐

```

WRT   :   COMP
Deg   :   Norm
486   Bytes Free
Prog  0123456789
    
```

MODE 4

```

WRT   :   Base-n
          Dec
486   Bytes Free
Prog  0123456789
    
```

Base-n モードに指定

EXE

```

-
    
```

以上のように、プログラムエリアの指定後、**EXE** キーを押す前に計算モードを指定すれば、そのプログラムエリアに計算モードが記憶されます。

■計算モードにおける注意点

各計算モードにおけるキー操作は全てプログラムとして記憶できますが、計算モードによってはプログラム用の命令が使えないモードもあります。

Base-n モード

- 関数計算は行なえません。
- 角度単位の指定はできません。
- プログラム用の命令は全て使えます。
- プログラム終了後、元の計算モードに戻るため、最後の結果出力には「▲」を必ずつけてください。つけない場合は、10進表示になったり、エラーになることがあります。

SD1、SD2モード

- 関数計算の内、Abs、 $\sqrt{\quad}$ は使えません。
- プログラム用の命令で Dsz、>、< は使えません。

LR1、LR2モード、

- 関数計算の内、Abs、 $\sqrt{\quad}$ は使えません。
- プログラム用の命令で、 \Rightarrow 、=、キ、Isz、 \geq 、 \leq 、Dsz、>、< は使えません。

4-6 プログラムの消し方

プログラムの消去はPCLモード(**MODE** ③ と押す)で行ないます。

プログラムの消去方法は2つあり、単一のプログラムエリア内のプログラムを消す場合と、P0からP9までの全部のプログラムを同時に消す場合により異なります。

■単一プログラムの消去

単一プログラムエリア内のプログラムを消すにはPCLモードでプログラムエリアを指定後、**AC** キーを押します。

例 P3のプログラムだけを消去する

操 作	表 示	
MODE ③	<pre>PCL : COMP Deg : Norm 324 Bytes Free Prog 12_45678_</pre>	P0、P3、P9にプログラムが記憶されています。
⇨ ⇨ ⇨	<pre>PCL : COMP Deg : Norm 324 Bytes Free Prog _12_45678_</pre>	P3にカーソルを合わせます。
AC	<pre>PCL : COMP Deg : Norm 367 Bytes Free Prog _12345678_</pre>	“3”の数字が表示されます。
MODE ①	<pre>*** MODE *** RUN : COMP Deg : Norm Step 0</pre>	RUNモードに戻しておきます。

■全プログラムの消去

P0からP9までの全プログラムエリア内のプログラムを消すには、PCL モードで **SHIFT** **DEL** と操作します。

例 P0、P4、P8、P9に入っている全プログラムを消去する

操 作

表 示

MODE **3**

```
PCL : COMP
Deg : Norm
295 Bytes Free
Prog 123_567__
```

SHIFT **DEL**

```
PCL : COMP
Deg : Norm
486 Bytes Free
Prog 0123456789
```

MODE **0**

```
*** MODE ***
RUN : COMP
Deg : Norm
Step 0
```

4-7 便利なプログラム命令

本機のプログラム方式はマニュアル計算を基本として作りますが、単一の計算式だけでなく、判断によりいくつかの計算式を使い分けたり、繰り返し実行させたりするために特別なプログラム命令があります。

ここではこのプログラム命令を使って、もっと便利なプログラムを作ってみましょう。

■ジャンプ命令

ジャンプ命令とはプログラムの実行の流れを変える命令です。プログラムの実行は入力した順(ステップ番号の小さい順)に行なわれ、プログラムの最後まで実行して終了します。このままでは同じ計算を繰り返したいときや、別の計算式に実行を移したいときなど不便です。そこでジャンプ命令が威力を発揮します。

ジャンプ命令は3種類あり、単純に分岐先にジャンプする無条件ジャンプ、条件を与えて正しいか正しくないかを判断して分岐先を決める条件ジャンプ、特定のメモリー内を1ずつ加えるか減じて0になったかならないかで分岐先を決めるカウントジャンプがあります。

◆無条件ジャンプ

無条件ジャンプとは“Goto”(ゴートゥー)と“Lbl”(ラベル)とにより構成され、プログラムの実行を Goto n (n は0~9)に対する Lbl n (Gotoと同じ番号)に移します。

無条件ジャンプは単純にプログラムの先頭に分岐して実行を繰り返したり、途中から繰り返すときによく使われます。また、条件ジャンプやカウントジャンプで組み合わせて使います。

例 前出の正4面体の表面積と体積を求めるプログラムを繰り返し実行できるように“Goto 1”と“Lbl 1”を加える。

前出のプログラムは

?, →, A, :, √, 3, X, A, x², ▲,
√, 2, ÷, 1, 2, X, A, x³, 3

19ステップ

※以後、本書ではプログラムを表わすときには1ステップずつ“、”を入れて上記のように記します。

このプログラムの最後に“Goto 1”を加え、分岐先であるプログラムの先頭に“Lbl 1”を加えます。

ただし、このまま単に加えただけでは体積を表示せずに最初の一边の入力に進んでしまいますから、“Goto 1”の前に表示命令の“▲”を入れてください。

無条件ジャンプを加えたプログラム

Lbl, 1, :, ?, →, A, :, √, 3, X, A, x², ▲,

√, 2, ÷, 1, 2, X, A, x³, 3, ▲, Goto, 1 25ステップ

では、このプログラムを実行してみましょう。

※プログラムの記憶および編集については、4-1または4-2をご覧ください。

※以後、表示窓は結果出力のみ記載します。

操 作

表 示

Prog [0] EXE	?	P0に入っています。
10 EXE	173.2050808	一边の長さ10とします。
EXE	117.8511302	
EXE	?	
7.5 EXE	97.42785793	一边の長さ7.5とします。
EXE	49.71844555	
EXE	?	

このままではプログラムが終了しませんので、終わらせたいときは [MODE] [1] と押します。

[MODE] [1]

***	MODE	***
RUN	:	COMP
Deg	:	Norm
Step	:	0

この他にもプログラムの先頭だけでなく途中に分岐することもできます。

例 $y = ax + b$ を計算するとき、 x の値はその都度変化し、 a 、 b は計算によっては変化することもある場合。

プログラム

?, →, A, :, ?, →, B, :, Lbl, 1, :, ?, →, X, :,

A, X, X, +, B, ▲, Goto, 1 23ステップ

このプログラムでは、実行開始後メモリー A と B に a、b の値を入力し、以後、x の値のみを入力します。

このように、無条件ジャンプは“Goto”(ゴトウー)と“Lbl”(ラベル)が対になり、プログラムの実行の流れを変えます。なお、Goto n に対する Lbl n がない場合にはエラー(Go ERROR)となります。

◆条件ジャンプ

条件ジャンプとは、メモリー内の数値と定数の比較や、メモリー内の数値どうしを比較して条件に合えば(正しければ)“⇒”の次の文を実行し、条件に合わなければ(正しくなければ)次の文を飛ばして“:”または“▲”、“◀”以降を続けて実行します。

条件ジャンプは次のような形よりできています。

左辺 関係演算子 右辺 ⇒文 { : } 文

※◀はキャリッジリターン機能(117ページ参照)です。

左辺と右辺はメモリー名(アルファベットの A ~ Z)を1つずつまたは数値定数や計算式(A×2、D-E 等)が入ります。

関係演算式は比較記号で、=、キ、≥、≤、>、<の6種類があります。

左辺=右辺…左辺と右辺が等しい

左辺キ右辺…左辺と右辺が等しくない

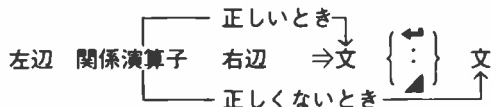
左辺≥右辺…左辺より右辺が小さいか等しい

左辺≤右辺…左辺より右辺が大きいか等しい

左辺>右辺…左辺より右辺が小さい

左辺<右辺…左辺より右辺大きい

“⇒”は **SHIFT** **7** と押すと表示される命令で、条件が正しければ“⇒”の次の文へ進みます。もし、条件が正しくなければ“⇒”の次の文を飛びこして、その次の“:”または“▲”、“◀”に続く文へ進みます。



文は計算式(sinA×5 等)やプログラム命令(Goto、Prog 等)で、次の“:”または“▲”までを1つの文とします。

例 入力した数値が0または0より大きければ、その値の平方根を表示し、0より小さければ再び入力を繰り返す。

プログラム

Lbl、1、:、?、→、A、:、A、≧、0、⇒、√、A、▲、Goto、1
16ステップ

※本書ではプログラム中の0(ゼロ)はアルファベットのO(オー)と区別するために0と記します。

このプログラムは入力された数値をメモリーのAに記憶させ、次にAメモリーの内容が0より大きいかわり等しいかわり(小さくなれば)“⇒”と“▲”の間の文(計算式)を実行し、続けて“Goto 1”により“Lbl 1”に戻ります。もし、Aメモリーの内容が0より小さければ、“▲”以降に飛び、“Goto 1”により“Lbl 1”に戻ります。

例 いくつかの数値の合計を計算する。ただし0を入力したときに、合計を表示するようにする。

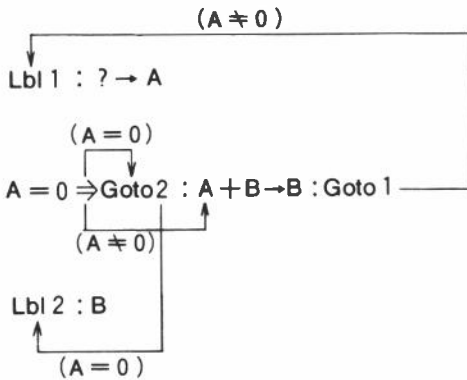
プログラム

0、→、B、:、
Lbl、1、:、?、→、A、:、A、=、0、⇒、Goto、2、:、
A、+、B、→、B、:、Goto、1、:、
Lbl、2、:、B
31ステップ

このプログラムは、まず合計を計算するBメモリーに0を入れてクリアしておきます。そして、“?→A”で入力された値をAメモリーに記憶させ、“A=0⇒”でAメモリーの内容が0か0でないかを判断し、0のときは“Goto 2”により“Lbl 2”へジャンプします。もし、Aメモリーが0でないときは、“Goto 2”を飛ばして“:”の次の命令“A+B→B”から実行を続け、“Goto 1”により再び“Lbl 1”に戻り、入力を繰り返します。

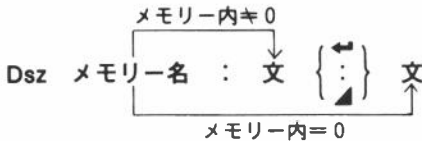
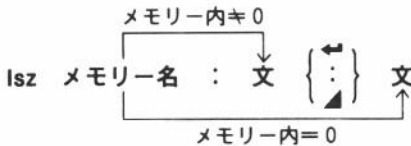
“Lbl 2”からは合計の入っているBメモリーを表示させます。本来“B”の後に表示命令の“▲”を入れますが、ここでは省略しています。

このプログラムの流れを追ってみますと、次のようになります。



◆カウントジャンプ

カウントジャンプとは、指定されたメモリー内の値を1つずつ加算したり、減算したりして、値が0でなければ次の文を実行し、値が0であれば次の文を飛ばして“:”または“▲”の次の文を実行します。メモリー内の値を1つずつ加算して判断するのが“Isz”(Increment and Skip on Zero)命令で、1つずつ減算して判断するのが“Dsz”(Decrement and Skip on Zero)命令です。



ここでいうメモリー名にはAからZのようなメモリーの名前(アルファベット1文字やA []等)を書きます。

例 Aメモリーに1つずつ加える…… Isz A

Bメモリーから1つずつ引く…… Dsz B

例 10個の数値を入力して平均を求める。

プログラム

1、0、→、A、:、0、→、C、:、
Lbl、1、:、?、→、B、:、B、+、C、→、C、:、
Dsz、A、:、Goto、1、:、C、÷、1、0 32ステップ

このプログラムは最初に A メモリーに10を、C メモリーに0を入れます。これは A メモリーをカウンター(数をかぞえる器)として使い、“Dsz”を使ってカウントダウンさせます。C メモリーは合計を取るメモリーですので、0を入れてクリアしておきます。

“?”により入力された数値は B メモリーに入れられ、続く“B+C → C”により、C メモリーに合計を取っていきます。

“Dsz A”は A メモリー内の数値から1を引いて、結果が0でなければ直後の文、ここでは“Goto 1”に進み、もし0であれば“Goto 1”の次の“:”を飛びこして、“C ÷ 10”に進みます。

例 地上から初速 V m/sec、角度 S° で打ち上げたボールの高さを1秒おきに求める。式は $h = V \sin \theta t - \frac{1}{2}gt^2$ で、 $g = 9.8$ として空気抵抗は考えないものとする。

プログラム

Deg、:、0、→、T、:、?、→、V、:、?、→、S、:、
Lbl、1、:、Isz、T、:、V、X、sin、S、X、T、-、
9、.、8、X、T、x²、÷、2、▲、Goto、1 38ステップ

このプログラムは、最初に角度単位の設定と初期値化(T メモリーをクリアする)をし、続いて V メモリーと S メモリーに初速と角度を入力します。

“Lbl 1”は繰り返しの最初の部分に入れますので入力の後に入れ、続けて“Isz T”で T メモリー内の数値をカウントアップ(1加える)しています。ここでの“Isz”の使い方は、T メモリー内の数値をカウントアップするために使い、判断によりジャンプさせてはしません。判断によるジャンプの仕方は前出の“Dsz”と同じですが、このようにカウンターとして1ずつ増やすだけの使い方もあります。もし、“Isz”を使わない場合は、“T + 1 → T”と5ステップ分必要としますが、このような使い方では“Isz T”の2ステップですみます。応用的な使い方ですが、覚えておくと便利です。

Tメモリーをカウントアップした後は、式に従い計算式通りに計算をし、高さを表示していきます。

なお、このプログラムでは終わりがないので、必要な高さまでを求めましたら、**MODE 1** と押して終了させてください。

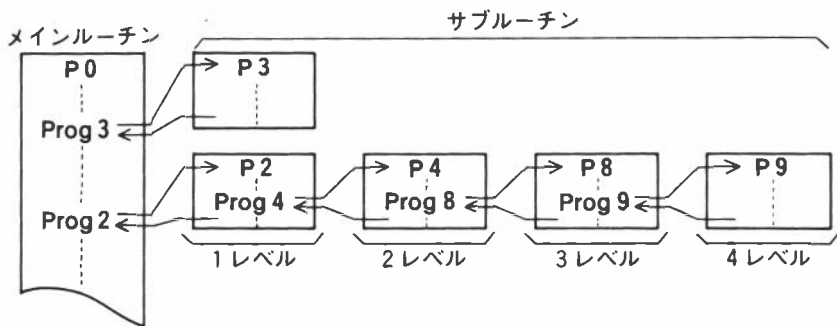
〈まとめ〉

命 令	書 式	内 容
無条件ジャンプ	Lbl n Goto n (n は 0 ~ 9 の自然数)	Goto n に対する Lbl n に無条件でジャンプする。
条件ジャンプ	左辺 関係演算子 右辺⇒ 文 $\left\{ \begin{array}{c} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\}$ 文 (関係演算子：=、キ、>、<、≥、≤)	左辺と右辺を比較して、条件式に合えば(正しければ)、“⇒”以降を実行し、正しくなければ：または▲以降にジャンプする。 文は数式や Goto 命令等が入りません。
カウント ジャンプ	Isz メモリー名：文 $\left\{ \begin{array}{c} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\}$ 文 Dsz メモリー名：文 $\left\{ \begin{array}{c} \leftarrow \\ \vdots \\ \blacktriangle \end{array} \right\}$ 文 (メモリー名：A ~ Z の 1 文字 または A [] 等)	メモリー内の数値を 1 ずつ加算 (Isz) または減算 (Dsz) して、0 になれば、：または▲以後にジャンプする。 文は数式や Goto 命令等が入りません。

■サブルーチン

今まで出てきたように 1 つのプログラムエリア内だけのプログラムを「メインルーチン」と呼び、このメインルーチンに対して他のプログラムエリアによく使うプログラムの一部を入れて使うことを「サブルーチン」と呼びます。

このサブルーチンとしての使い方は、繰り返し使う計算式を 1 つのブロックとして組み、その都度ジャンプして使ったり、よく使う計算式や操作を独立させて色々なプログラムからジャンプさせて使うのに便利です。



サブルーチン命令は“Prog”を使い、続く 0～9 の番号でプログラムエリアを指定します。

例 Prog 0…プログラムエリア P 0へジャンプする

Prog 2…プログラムエリア P 2へジャンプする

Prog命令によりジャンプした後は、指定のプログラムエリア内のプログラムを先頭から実行し、最後まで終ると再びもとのプログラムエリアの“Prog n”の次の命令に戻ります。サブルーチンとして組まれているプログラムの内からさらにサブルーチンへジャンプすることもできます。このサブルーチンへジャンプすることを「ネスティング」と呼び、全部で10レベル(10段)できます。このレベルをこえようとエラー(Ne ERROR)となります。また、“Prog n”に対するプログラムエリア Pnにプログラムが記憶されていないときはエラー(Go ERROR)となります。ただし、“?”による入力バッファを含みますので、サブルーチン中で入力命令があるときは、9レベルまでとなります。

サブルーチン内の Goto nは、そのプログラムエリア内のLbnにジャンプします。

例 前出の正 8 面体と正 4 面体の表面積と体積を求めるプログラムを同時に組む。なお、答えは小数点以下 3 桁まで求める。

この例では、すでに説明したプログラムを使い、最初に小数点以下指定(**MODE** 7)を組み合わせます。

では、2つのプログラムを見てみましょう

正8面体

P0 Fix、3、 \div 、 $\sqrt{\quad}$ 、 \rightarrow 、A、 \div 、2、 \times 、 $\sqrt{\quad}$ 、3、 \times 、A、 x^2 、 \blacktriangleleft 、
 $\sqrt{\quad}$ 、2、 \div 、3、 \times 、A、 x^3 、3 23ステップ

正4面体

P1 Fix、3、 \div 、 $\sqrt{\quad}$ 、 \rightarrow 、A、 \div 、 $\sqrt{\quad}$ 、3、 \times 、A、 x^2 、 \blacktriangleleft 、
 $\sqrt{\quad}$ 、2、 \div 、1、2、 \times 、A、 x^3 、3 22ステップ
計 45ステップ

この2つのプログラムを比較してみると を引いた部分は同じ内容ですので、共通して使えればプログラムが簡単になり、ステップ数も少なくて済みます。また、 を引いた部分はそのままでは共通になりませんが、P1の正4面体の体積部分を“ $\sqrt{\quad}$ 、2、 \div 、3、 \times 、A、 x^3 、3、 \div 、4”と考えれば共通になります。では を引いた部分をP9に、 を引いた部分をP8に独立させてみましょう。

P9 Fix、3、 \div 、 $\sqrt{\quad}$ 、 \rightarrow 、A、 \div 、 $\sqrt{\quad}$ 、3、 \times 、A、 x^2 12ステップ

P8 $\sqrt{\quad}$ 、2、 \div 、3、 \times 、A、 x^3 、3 8ステップ

次に、共通部分を取り除いた残りを、正8面体はP0に、正4面体はP1に組んでみましょう。共通部分のP9、P8のサブルーチンへのジャンプ“Prog 9”、“Prog 8”を忘れずに追加します。

P0 Prog、9、 \div 、Ans、 \times 、2、 \blacktriangleleft 、Prog、8 9ステップ

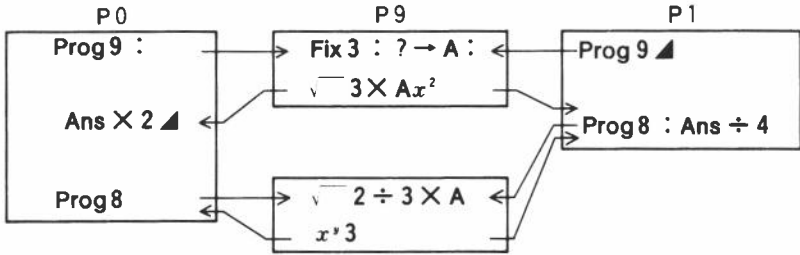
P1 Prog、9、 \blacktriangleleft 、Prog、8、 \div 、Ans、 \div 、4 9ステップ

計 38ステップ

このように組むと、P0またはP1のプログラムをスタートした後はP9のプログラムにジャンプし、小数点以下3桁指定を行ない、1辺の値を入力して表面積の計算をします。P0の正8面体では“ $2 \times$ ”が抜けていますので、P0に戻った後“Ans \times 2”で答えを求めます。P1の正4面体ではそのまま答えが求められていますので、P1に戻り表示させます。体積の計算も同様に、P8へジャンプして計算を行なった後メインルーチンのP0またはP1に戻り、P0ではそのままプログラムを終了させて、答えを表示します。P1ではP8で求められた答えをさらに4で割り、答えを求めます。

このようにサブルーチンを使ってプログラムを組んだ方が、ステップ数が短くなり、見たためにもスッキリしてきます。

ではこのプログラムの流れを見てみましょう。



サブルーチンはプログラムの共通部分を取り出して、別のプログラムエリアに組むことにより、ステップ数を短くしたり、プログラムをブロックごとに別けてプログラムを構造化するとき便利な命令です。

■キャリッジリターン機能

キャリッジリターン機能とは、命令の区切りである“:”のかわりに **EXE** キーを押すことにより、プログラム内容の表示を見やすくすることができます。

```

Deg : 0 -> T : ? -> V : ? -> S :
Lb l 1 : l s z T : V X s i
n S X T - 9 . 8 X T ² ÷ 2 ▲
G o t o 1
  
```

このプログラムをキャリッジリターン機能により区切ってみますと、次のようになります。

```

Deg
0 -> T : ? -> V : ? -> S
Lb l 1 : l s z T : V X s i
n S X T - 9 . 8 X T ² ÷ 2 ▲
  
```

この2カ所で **EXE** キーを押す。何も表示されず、次の行に改行する。

このようにすれば、角度単位の指定や Lbl の位置によるループの動きなどがわかりやすくなります。

操作手順

MODE 4 EXE (⏏ のかわりに押す)

⏏ → ALPHA T ⏏ SHIFT ? → ALPHA V ⏏ SHIFT ? → ALPHA S EXE

SHIFT Lbl 1 ⏏ ……

☆一度入力したプログラムにキャリッジリターンを組むには、SHIFT INS と押し
て挿入モードとしてから EXE キーを押し、その後“:”を削除します。

```

Deg : 0 → T : ? → V : ? → S :
L b l 1 : l s z T : V X s i
n S X T - 9 . 8 X T ^ 2 ÷ 2 ↓
G o t o 1
    
```

カーソルを“Deg”の後の“:”に合わせ SHIFT INS と押してから EXE を挿入します。

⇨ SHIFT INS EXE

```

Deg
[:] 0 → T : ? → V : ? → S : L b l
1 : l s z T : V X s i n S
X T - 9 . 8 X T ^ 2 ÷ 2 ↓
    
```

DEL

```

Deg
[0] → T : ? → V : ? → S : L b l
1 : l s z T : V X s i n S X
T - 9 . 8 X T ^ 2 ÷ 2 ↓
    
```

次の場所である“? → S”の次の“:”にカーソルを合わせ、同様に EXE を挿入
してから、“:”を削除します。

⇨ ~ ⇨ SHIFT

INS EXE DEL

```

Deg
0 → T : ? → V : ? → S
[ ] b l 1 : l s z T : V X s i
n S X T - 9 . 8 X T ^ 2 ÷ 2 ↓
    
```

以上で、キャリッジリターン機能が組み込みました。

☆マニュアルでも SHIFT EXE とすることにより、キャリッジリターン機能が使え
ます。

4-8 メモリーの配列的使用

■配列的使用とは？

今まで出てきたメモリーは A とか B、X、Y のようにアルファベット 1 文字を名前として使われてきました。

ここで説明する配列的使用とは、メモリー名(アルファベット 1 文字 A ~ Z) に [1]、[2] のような添字を付けて使う方法です。(※[] は `ALPHA` □、`ALPHA` `EXP` と押す)

今までのメモリー	配列的メモリー	
A	A[0]	C[-2]
B	A[1]	C[-1]
C	A[2]	C[0]
D	A[3]	C[1]
E	A[4]	C[2]

この添字(番号)によりメモリーを使いわけて、プログラムを短く簡単にすることができます。この添字には負数も使え、0 番目のメモリーを基準にして計算します。

例 10個のメモリー A ~ J に 1 ~ 10の数値を入れる

今までのメモリー使用

1、→、A、:、2、→、B、:、3、→、C、:、4、→、D、:、
5、→、E、:、6、→、F、:、7、→、G、:、8、→、H、:、
9、→、I、:、1、0、→、J 40ステップ

配列的メモリー使用

0、→、Z、:、Lbl、1、:、Z、+、1、→、A、[、Z、]、:、
lsz、Z、:、Z、<、1、0、⇒、Goto、1 26ステップ

このようにいちいち「→A」、「→B」と入力してはめんどろです。この例のような場合にはこれ位ですみましたが、もし何番目かの指定をして、任意のメモリー内容を見たいときはどうでしょう。

今までのメモリー使用

Lbl、1、:、?、→、Z、:、
Z、=、1、⇒、A、▲、Z、=、2、⇒、B、▲、
Z、=、3、⇒、C、▲、Z、=、4、⇒、D、▲、
Z、=、5、⇒、E、▲、Z、=、6、⇒、F、▲、
Z、=、7、⇒、G、▲、Z、=、8、⇒、H、▲、
Z、=、9、⇒、I、▲、Z、=、1、0、⇒、J、▲、
Goto、1

70ステップ

配列的メモリー使用

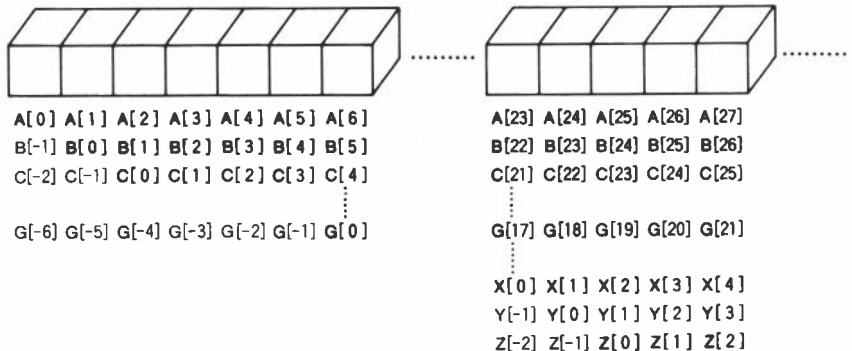
Lbl、1、:、?、→、Z、:、A、[、Z、-、1、]、▲、
Goto、1

16ステップ

このようにしてみれば、だいぶ違います。今までのメモリーの使い方では、1～10を判断して表示させますが、配列的使い方ではZメモリーの内容(数値)を“[Z-1]”により変えるだけで簡単に選ぶことができます。添字の中は計算式(Z-1、A+10等)が使えますので、とても便利です。

■配列的使い方の注意点

メモリーを配列として使う場合、メモリー名をA～Zから選び、添字により使い分けても、今までのA～Zとして使っていたメモリーと同じメモリーを使っているので、重複して使わないように気を付けなければなりません。



Zメモリー以降のA [26]やZ [1]などは、配列的な使い方使います。
 使い方が異っても、使うメモリーは基本のA～Zの26個とMODE □ (Defm)により
 増設されたメモリーですので、アルファベット1文字のメモリーとして使って
 も、添字を付けて配列的に使っても、同じメモリーを使うことがありますので注
 意してください。

次のような使い方は配列的メモリーと今までのメモリーとを重複して使っていま
 すので、このような使い方はしないように注意してください。

例 A [1]～A [5]に1～5を記憶させる。

5、→、C、∴、Lbl、1、∴、C、→、A、[、C、]、∴、

Dsz、C、∴、Goto、1、∴、

A、[、1、]、▲、A、[、2、]、▲、A、[、3、]、▲、

A、[、4、]、▲、A、[、5、] 44ステップ

このプログラムでは配列としてA [1]～A [5]に1～5を記憶しますが、カウ
 ンター用のメモリーとしてCメモリーを使っています。

このプログラムを実行すると次のようになります。

操 作	表 示
Prog 0 EXE	1.
EXE	0.
EXE	3.
EXE	4.
EXE	5.

2番目のA [2]の値が正しくありません。正しくは2ですので、プログラムに
 誤りがあります。実はA [2]というメモリーはCメモリーと同じだからです。

A	B	C	D	E	F
A [1]	A [2]	A [3]	A [4]	A [5]	

このため、Cメモリーの内容が5から1ずつ減っていき、最後に0になって表
 示に移りますので、CメモリーすなわちA [2]は0が表示されます。

配列的な使い方をするときには、このようなメモリーの重複をさけて使ってくだ
 さい。

■配列的使用の応用

たくさんのデータを使うときに、必ず2種類のデータを1組として使う場合があります。このようなときにはメモリーをデータ数分だけずらして使います。

例 2つのデータ x 、 y を記憶させ、 x の値を入力して y の値を表示させる。

なお、データ数は15個とする。

プログラム例1

データの制御用メモリーにAメモリーを、 x データの一時入力用にBメモリーを使い、C[1](D)からC[15](R)メモリーに x データを、C[16](S)からC[30](Z(7))メモリーに y データを記憶させます。

```
1、→、A、：、Defm、7、：、  
Lbl、1、：、？、→、C、[、A、]、：、  
？、→、C、[、A、+、1、5、]、：、  
Isz、A、：、A、=、1、6、⇒、Goto、2、：、Goto、1、：、  
Lbl、2、：、1、5、→、A、：、？、→、B、：、  
B、=、0、⇒、Goto、5、：、  
Lbl、3、：、B、=、C、[、A、]、⇒、Goto、4、：、  
Dsz、A、：、Goto、3、：、Goto、2、：、  
Lbl、4、：、C、[、A、+、1、5、]、▲、Goto、2、：、  
Lbl、5
```

98ステップ

このプログラムではメモリーを次のようにして使っています。

x データ

```
C[1] C[2] C[3] C[4] C[5] C[6] C[7] C[8] C[9] C[10]  
D E F G H I J K L M  
C[11] C[12] C[13] C[14] C[15]  
N O P Q R
```

y データ

```
C[16] C[17] C[18] C[19] C[20] C[21] C[22] C[23] C[24] C[25]  
S T U V W X Y Z Z(1) Z(2)  
C[26] C[27] C[28] C[29] C[30]  
Z(3) Z(4) Z(5) Z(6) Z(7)
```

プログラム例 2

使うメモリーは同じですが、メモリーの名前を 2 種類にして x 、 y データを区別します。

```
1、→、A、：、Defm、7、：、  
Lbl、1、：、？、→、C、[、A、]、：、  
？、→、R、[、A、]、：、  
Isz、A、：、A、=、1、6、⇒、Goto、2、：、Goto、1、：、  
Lbl、2、：、1、5、→、A、：、？、→、B、：、  
B、=、0、⇒、Goto、5、：、  
Lbl、3、：、B、=、C、[、A、]、⇒、Goto、4、：、  
Dsz、A、：、Goto、3、：、Goto、2、：、  
Lbl、4、：、R、[、A、]、▲、Goto、2、：、  
Lbl、5
```

92ステップ

メモリーの使い方

x データ

```
C [1] C [2] C [3] C [4] C [5] C [6] C [7] C [8] C [9] C [10]  
D   E   F   G   H   I   J   K   L   M  
C [11] C [12] C [13] C [14] C [15]  
N   O   P   Q   R
```

y データ

```
R [1] R [2] R [3] R [4] R [5] R [6] R [7] R [8] R [9] R [10]  
S   T   U   V   W   X   Y   Z   Z (1) Z (2)  
R [11] R [12] R [13] R [14] R [15]  
Z (3) Z (4) Z (5) Z (6) Z (7)
```

このように、使うメモリーが同じでもメモリー名を変えて使うこともできます。ただしメモリーの増設(**MODE** □ : Defm)により増えた部分については、メモリー名が A ~ Z までのため、配列としての使い方だけとなります。

※メモリー数を標準の26メモリー以上使うときは、増設命令もプログラム中に組み込んでおいてください。

例 メモリーを14個増やして40個を使う場合

```
Defm、1、4、：、……
```

4-9 アルファ文字の表示

アルファ文字とは、**ALPHA** キーに続けて押すアルファベット文字や数字、演算命令のことで“ ”(引用符：**ALPHA Prog**)に囲まれた中の文字を表示させて、メッセージなどに使います。

■アルファ文字

● **ALPHA** キー後に押して表示する文字

[、]、k、m、μ、n、p、f、スペース、
A、B、C、D、E、F、G、H、I、J、K、L、M、N、
O、P、Q、R、S、T、U、V、W、X、Y、Z

● その他の数字、記号、演算命令、プログラム命令

0、1、2、3、4、5、6、7、8、9、
(、)、√、ε、+、-、×、÷、……
sin、cos、tan、log、ln、……
=、≠、≥、≤、>、<、……
A、B、C、D、E、F、d、h、b、o、
Neg、Not、and、or、xor
x、y、xon、xon-1、……
°(**SHIFT** **MODE** [4])、°(**SHIFT** **MODE** [5])、°(**SHIFT** **MODE** [6])

※これらのように、表示される文字は全てアルファ文字として使うことができます。

例えば、前出の例のようにxデータとyデータの2種類を入力するような場合、“?”命令だけでは何を入力していいかわかりません。この入力の確認のために“?”の前にメッセージを入れてみましょう。

Lbl、1、:、?、→、X、:、?、→、Y、:、……

このプログラムに“X=”と“Y=”のメッセージを入れてみます。

Lbl、1、:、“X、=”、?、→、X、:、

“、Y、=”、?、→、Y、:、……

このようにメッセージを加えれば、次のように表示されます。

P1に記憶しているとします。

Prog 1	EXE	X = ?
10	EXE	Y = ?
⋮		⋮

プログラム計算で計算結果の内容を表示させるときのメッセージにも便利です。

Lbl、0、:、”、N、=、”、?、→、B、～、C、:、

0、→、A、:、

Lbl、1、:、C、÷、2、→、C、:、Frac、C、キ、0、⇒、Goto、3、

:、lsz、A、:、C、=、1、⇒、Goto、2、:、Goto、1、:、

Lbl、2、:、”、X、=、”、▲、A、▲、Goto、0、:、

Lbl、3、:、”、N、O、”、▲、Goto、0

70 ステップ

このプログラムは2のx乗を計算するプログラムで、データの入力時は“N = ?”と表示されます。答えは“X =”と表示後、EXE キーを押すと次の行に表示されます。また、データが2のx乗でないときは“NO”を表示して再入力になります。

P2に記憶してあるとします。

Prog 2	EXE	N = ?
4096	EXE	X =
	EXE	12.
	EXE	N = ?
3124	EXE	NO
	EXE	N = ?
512	EXE	X =
	EXE	9.

※アルファ文字の後に計算式の結果を表示する場合は、アルファ文字の後を▲（表示命令）で区切ります。

アルファ文字を17文字以上表示させますと、次の行の先頭から続けて表示します。
また、アルファ文が表示窓の最下行の最後に表示された場合は一定時間後、順に
上方に送られ、最上行の表示内容は表示窓から消えます。

Prog 0

```
968+125-65
                                     1028.
Prog 0_
```

EXE

```
968+125-65
                                     1028.
Prog 0
ABCDEFGHIJKLMN
```

↓ 一定時間後

```
                                     1028.
Prog 0
ABCDEFGHIJKLMN
PQRSTUVWXYZ
```

4-10 グラフィック機能をプログラムに組む

便利なグラフィック機能をプログラムに組むことにより、長くて複雑な方程式や複数のグラフの重ね描きを繰り返し描かせることができます。

グラフィック命令はトレース機能以外は全てプログラム中に組み込むことができ、レンジ内容などを組み込んでおけば、自動的に設定されてグラフが描けます。グラフを描くプログラムも、基本的にはマニュアルで行なう操作をそのまま記憶させます。

例1) 次の2本のグラフが実根となる解の数を求める。

$$y = x^4 - x^3 - 24x^2 + 4x + 80$$

$$y = 10x - 30$$

レンジは次の内容とする。

X Range

min: -10.

max: 10.

scl: 2.

Y Range

min: -120.

max: 150.

scl: 50.

まず、レンジの設定をプログラムにします。

各値は“,”で区切ります。

Range、(-)、1、0、,、1、0、,、2、,、(-)、1、2、0、,、1、5、0、,、5、0

次に、1本目のグラフをプログラムにします。

Graph、X、x⁴、4、-、X、x³、3、-、2、4、X、x²、+、4、X、+、8、0

2本目のグラフも同様にプログラムにします。

Graph、1、0、X、-、3、0

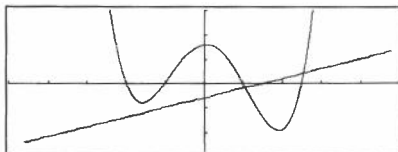
計49ステップ

これでプログラムができました。このプログラムを入力しますが、レンジと1本目のグラフの終りに **EXE** キーを押して続けて入力してみます。

```
120,150,50
Graph Y=Xx^4-Xx^3
3-24X^2+4X+80
Graph Y=10X-30_
```

では、実行してみましょう。

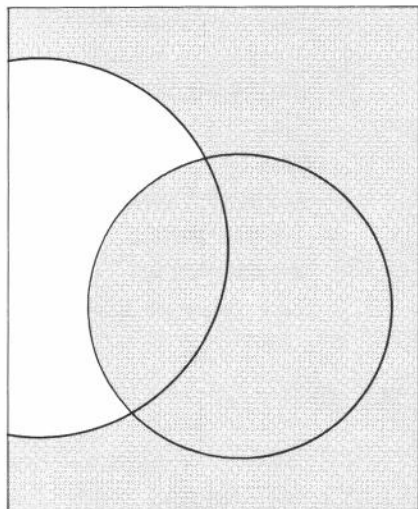
MODE **T**
Prog **0** **EXE**



もし、1本目のグラフ表示後、画面を停止させるときは、**EXE** で区切るかわりに“**▲**”を入れてください。一度停止し、次の **EXE** キーで続けて2本目のグラフを描きます。

このように、プログラムに組み込むことにより、いろいろなグラフが描けます。このようなプログラムによるグラフィックについては巻末のライブラリー編を参考にしてください。

ライブラリー編



〈お使いの前に〉

※プログラムを記憶させる前に、残りのステップ数を確認してください。

※本ライブラリーは、数値計算編とグラフィック編の2種類に分れています。

数値計算編は、答えのみ記載し、グラフィック編は表示窓全部を記載しています。

また、数値計算編のプログラムは計算中の“×”を省略していませんが、グラフィック編のプログラムは省略しています。

※グラフィック編のプログラムでは、表示を見やすくするためにキャリッジリターン機能(↵)を使っています。“↵”の場所では [EXE] キーを押してください。(↵は表示されません)

※プログラム中の「Graph」は、[Graph] キーを押してください。(Graph Y=と表示されます)

※計算モードの指定は、[MODE] [2] と押して WRT モードにした後、[SHIFT] [MODE] [田] や [MODE] [田] と押してモード指定後に [EXE] キーを押します。

CASIO PROGRAM SHEET

プログラム名 <b style="text-align: center;">素 因 数 分 解	No. <b style="text-align: center;">1
--	---

内容計算式等

任意の正の整数 m の素因数を選び出す。

但し $1 < m < 10^{10}$ で、

素数は小さな方から選び出し、“END”を表示したら終了とする。

〈考え方〉

m を順次 2 および $d=3, 5, 7, 9, 11, 13, \dots$ (奇数のすべて) の数列で割り、割り切れるかどうかを調べる。

d が素数だった場合 $m_i = m_{i-1} / d$ とし、

$\sqrt{m_i} + 1 \leq d$ まで割算を繰り返す。

〈例 1〉

例 題 $119 = 7 \times 17$

〈例 2〉

$1234567890 = 2 \times 3 \times 3 \times 5 \times 3607 \times 3803$

〈例 3〉

$987654321 = 3 \times 3 \times 17 \times 17 \times 379721$

準備および操作 ●右のプログラムを計算機に覚えさせます。
 ●**[MODE]** **[1]** (RUN 状態) で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	[Prog] 0 [EXE]	M ?	11	[EXE]	3803.
2	119 [EXE]	7.	12	[EXE]	END
3	[EXE]	17.	13	[EXE]	M ?
4	[EXE]	END	14	987654321 [EXE]	3.
5	[EXE]	M ?	15	[EXE]	3.
6	1234567890 [EXE]	2.	16	[EXE]	17.
7	[EXE]	3.	17	[EXE]	17.
8	[EXE]	3.	18	[EXE]	(約12秒後) 379721.
9	[EXE]	5.	19	[EXE]	END
10	[EXE]	(約1分14秒後) 3607.	20		

行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ	
1	Mcl :			2	
2	Lbl 0 :	" M " ? → A : Goto 2 :		15	
3	Lbl 1 :	2 ▲ A ÷ 2 → A : A = 1 ⇒		30	
4	Goto 9 :			33	
5	Lbl 2 :	Frac (A ÷ 2) = 0 ⇒ Goto 1 :		48	
6	3 → B :			52	
7	Lbl 3 :	$\sqrt{\quad}$ A + 1 → C :		62	
8	Lbl 4 :	B ≥ C ⇒ Goto 8 : Frac (A ÷ B		77	
9) = 0 ⇒ Goto 6 :			84	
10	Lbl 5 :	B + 2 → B : Goto 4 :		96	
11	Lbl 6 :	A ÷ B × B - A = 0 ⇒ Goto 7		111	
12	: Goto 5 :			115	
13	Lbl 7 :	B ▲ A ÷ B → A : Goto 3 :		129	
14	Lbl 8 :	A ▲		134	
15	Lbl 9 :	" E N D " ▲ Goto 0		145	
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
メモリー内容	A	m_i	H	O	V
	B	d	I	P	W
	C	$\sqrt{m_i+1}$	J	Q	X
	D		K	R	Y
	E		L	S	Z
	F		M	T	
	G		N	U	

CASIO PROGRAM SHEET

プログラム名 <b style="text-align: center;">最 大 公 約 数	No. <b style="text-align: center;">2
--	---

内容計算式等

ユークリッドの互除法で、 a 、 b 2つの整数の最大公約数を求める。

但し、 $|a|$ 、 $|b| < 10^9$ 正の場合は $< 10^{10}$ とする。

〈考え方〉

$$n_0 = \max(|a|, |b|)$$

$$n_1 = \min(|a|, |b|)$$

$$n_k = n_{k-2} - \left\lfloor \frac{n_{k-2}}{n_{k-1}} \right\rfloor n_{k-1}$$

$$k = 2, 3, \dots$$

$n_k = 0$ ならば、最大公約数(c)は、 n_{k-1} となる。

例 題

〈例 1〉

$$a = 238$$

$$b = 374$$

のとき

$$c = 34$$

〈例 2〉

$$a = 23345$$

$$b = 9135$$

のとき

$$c = 1015$$

〈例 3〉

$$a = 522952$$

$$b = 3208137866$$

のとき

$$c = 998$$

準備および操作

●右のプログラムを計算機に覚えさせます。

●**MODE** **1** (RUN 状態)で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手 順	キ ー 操 作	表 示
1	Prog 0 EXE	A ?	11		
2	238 EXE	B ?	12		
3	374 EXE	34.	13		
4	EXE	A ?	14		
5	23345 EXE	B ?	15		
6	9135 EXE	1015.	16		
7	EXE	A ?	17		
8	522952 EXE	B ?	18		
9	3208137866 EXE	998.	19		
10			20		

行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	Lbl 1 :	" A " ? → A :	" B " ? →	15
2	B :			17
3	Abs A → A :	Abs B → B :		27
4	B < A ⇒ Goto 2 :			34
5	A → C :	B → A :	C → B :	46
6	Lbl 2 :	(-)(Int (A + B)) × B - A		61
7) → C :			65
8	C = 0 ⇒ Goto 3 :			72
9	B → A :	C → B :	Goto 2 :	83
10	Lbl 3 :	B ▲ Goto 1		90
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				

メモリー内容	A	a, n_0	H		O		V
	B	b, n_1	I		P		W
	C	n_k	J		Q		X
	D		K		R		Y
	E		L		S		Z
	F		M		T		
	G		N		U		

CASIO PROGRAM SHEET

プログラム名 <b style="font-size: 1.2em;">シンプソン法による定積分	No. <b style="font-size: 1.2em;">3
---	---

内容計算式等

$$I = \int_a^b f(x) dx = \frac{h}{3} |y_0 + 4(y_1 + y_3 + \dots + y_{2m-1}) + 2(y_2 + y_4 + \dots + y_{2m-2}) + y_{2m}|$$

$$h = \frac{b-a}{2m}$$

式変形

$$I = \frac{h}{3} |y_0 + \sum_{i=1}^m (4y_{2i-1} + 2y_{2i}) - y_{2m}|$$

$$f(x) = \frac{1}{x^2+1} \text{ の場合}$$

例題

<1> $a = 0, b = 1, 2m = 10$

$$I = \int_0^1 \frac{1}{x^2+1} dx = 0.7853981537$$

<2> $a = 2, b = 5, 2m = 20$

$$I = \int_2^5 \frac{1}{x^2+1} dx = 0.2662526769$$

準備および操作

- 右のプログラムを計算機に覚えさせます。
- **MODE** **1** (RUN 状態) で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	Prog 0 EXE	A ?	11		
2	0 EXE	B ?	12		
3	1 EXE	2 M ?	13		
4	10 EXE	0.7853981535	14		
5	EXE	A ?	15		
6	2 EXE	B ?	16		
7	5 EXE	2 M ?	17		
8	20 EXE	0.2662526769	18		
9			19		
10			20		

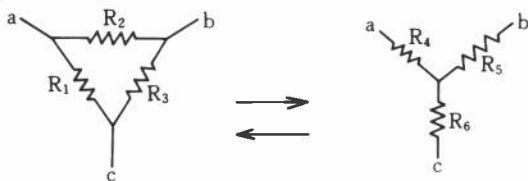
行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	P0			
2	Lbl 1 : Mcl :			5
3	" A " ? → A : " B " ? → B : "			20
4	2 M " ? → M :			27
5	A → G : Prog: 1 : P → I : (B - A			42
6) + M → D : M ÷ 2 → O :			54
7	Lbl 2 : G + D → G : Prog: 1 : I + P			69
8	X 4 → I :			74
9	G + D → G : Prog: 1 : I + P X 2 →			89
10	I : O - 1 → O :			97
11	O ≠ 0 ⇒ Goto: 2 :			104
12	B → G : Prog: 1 : I - P → I :			117
13	D X I ÷ 3 ↙			123
14	Goto: 1			125
15				
16	P1			
17	I ÷ (G X G + 1) → P			11
18				
19			計 136	
20				
21				
22				
23				
24				
25				
26				
27				
28				

メモリー内容	A	a	H		O	m (ループ回数)	V
	B	b	I	I	P		W
	C		J		Q		X
	D	$h = \frac{b-a}{2m}$	K		R		Y
	E		L		S		Z
	F		M	$2m$	T		
	G	x	N		U		

CASIO PROGRAM SHEET

プログラム名 △ ↔ Y 変換	No. 4
--	--

内容計算式等



1) $\Delta \rightarrow Y$

$$R_4 = \frac{R_1 \cdot R_2}{R_1 + R_2 + R_3}$$

$$R_5 = \frac{R_2 \cdot R_3}{R_1 + R_2 + R_3}$$

$$R_6 = \frac{R_3 \cdot R_1}{R_1 + R_2 + R_3}$$

2) $Y \rightarrow \Delta$

$$R_1 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_5}$$

$$R_2 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_6}$$

$$R_3 = \frac{R_4 R_5 + R_5 R_6 + R_6 R_4}{R_4}$$

例題

<1>

$$R_1 = 12(\Omega)$$

$$R_2 = 47(\Omega)$$

$$R_3 = 82(\Omega)$$

<2>

$$R_4 = 100(\Omega)$$

$$R_5 = 150(\Omega)$$

$$R_6 = 220(\Omega)$$

準備および操作

- 右のプログラムを計算機に覚えさせます。
- **MODE** **1** (RUN 状態)で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	Prog 0 EXE	D→Y:1,Y→D:2?	11	EXE	D→Y:1,Y→D:2?
2	1 EXE	R 1 = ?	12	2 EXE	R 4 = ?
3	12 EXE	R 2 = ?	13	100 EXE	R 5 = ?
4	47 EXE	R 3 = ?	14	150 EXE	R 6 = ?
5	82 EXE	R 4 =	15	220 EXE	R 1 =
6	EXE	4.	16	EXE	466.6666667
7	EXE	R 5 =	17	EXE	R 2 =
8	EXE	27.33333333	18	EXE	318.1818182
9	EXE	R 6 =	19	EXE	R 3 =
10	EXE	6.978723404	20	EXE	700.

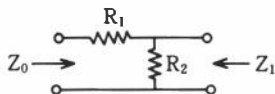
行	(MODE [2] に続いて下の命令の順にキーを押す)						プログラム	実行内容	ステップ						
1	Lbl	1	:	"	D → Y	: 1 , Y → D	: 2		15						
2	"	?	→	N	:				20						
3	N	=	2	⇒	Goto: 2	:	N キ 1 ⇒ Goto 1	:	34						
4	"	R	1	=	"	? → A	:		43						
5	"	R	2	=	"	? → B	:		52						
6	"	R	3	=	"	? → C	:		61						
7	A	+	B	+	C	→	D	:	69						
8	"	R	4	=	"	▲ A × B ÷ D	▲		81						
9	"	R	5	=	"	▲ B × C ÷ D	▲		93						
10	"	R	6	=	"	▲ A × C ÷ D	▲		105						
11	Goto:	1	:						108						
12	Lbl	2	:						111						
13	"	R	4	=	"	? → E	:		120						
14	"	R	5	=	"	? → F	:		129						
15	"	R	6	=	"	? → G	:		138						
16	E	×	F	+	F	×	G	+	G	×	E	→	H	:	152
17	"	R	1	=	"	▲ H ÷ F	▲		162						
18	"	R	2	=	"	▲ H ÷ G	▲		172						
19	"	R	3	=	"	▲ H ÷ E	▲		182						
20	Goto:	1							184						
21															
22															
23															
24															
25															
26															
27															
28															
メモリー内容	A		R ₁		H		R ₄ R ₅ + R ₃ R ₆ + R ₆ R ₄		O				V		
	B		R ₂		I				P				W		
	C		R ₃		J				Q				X		
	D		R ₁ + R ₂ + R ₃		K				R				Y		
	E		R ₄		L				S				Z		
	F		R ₅		M				T						
	G		R ₆		N			判断用	U						

CASIO PROGRAM SHEET

プログラム名 <b style="font-size: 1.2em;">最 小 損 失 整 合	No. <b style="font-size: 1.2em;">5
--	--

内容計算式等

Z_0 と Z_1 を最小損失でマッチングする R_1 、 R_2 を求める。



$$R_1 = Z_0 \sqrt{1 - \frac{Z_1}{Z_0}} \qquad R_2 = \frac{Z_1}{\sqrt{1 - \frac{Z_1}{Z_0}}}$$

$$\text{最小損失 } L_{\min} = 20 \cdot \log \left(\sqrt{\frac{Z_0}{Z_1}} + \sqrt{\frac{Z_0}{Z_1} - 1} \right) [\text{dB}]$$

例 題

$$\left(\begin{array}{l} Z_0 = 500 \Omega \\ Z_1 = 200 \Omega \end{array} \right. \text{のときの} \left. \begin{array}{l} R_1 \\ R_2 \\ L_{\min} \end{array} \right. \text{を求めよ}$$

- 準備および操作
- 右のプログラムを計算機に覚えさせます。
 - MODE** **1** (RUN 状態)で下のキー操作の順にキーを押します。

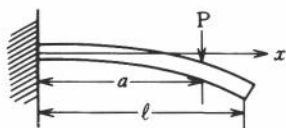
手順	キ ー 操 作	表 示	手 順	キ ー 操 作	表 示
1	Prog 0 EXE	$Z_0 = ?$	11		
2	500 EXE	$Z_1 = ?$	12		
3	200 EXE	$R_1 =$	13		
4	EXE	387.2983346	14		
5	EXE	$R_2 =$	15		
6	EXE	258.1988897	16		
7	EXE	$L_{\min} =$	17		
8	EXE	8.961393328	18		
9			19		
10			20		

行	(MODE) に続いて下の命令の順にキーを押す)						プログラム	実行内容	ステップ
1	"	Z	0	=	"	? → Y :		9	
2	"	Z	1	=	"	? → Z :		18	
3	√	(1	-	Z	+ Y) → A :		29	
4	Y	×	A	→	R	: Z ÷ A → S :	Y ÷ Z	44	
	→	B	:	2	0	×	log: (√ B + √ (B -	59	
6	1))	→	T	:		65	
7	"	R	1	=	"	▲ R ▲		73	
8	"	R	2	=	"	▲ S ▲		81	
9	"	L	M	I	N	=	" ▲ T	90	
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
メモリー内容	A	$\sqrt{1 - \frac{Z_1}{Z_0}}$	H		O		V		
	B	$\frac{Z_0}{Z_1}$	I		P		W		
	C		J		Q		X		
	D		K		R	R 1	Y	Z ₀	
	E		L		S	R 2	Z	Z ₁	
	F		M		T	Lmin			
	G		N		U				

CASIO PROGRAM SHEET

プログラム名 <b style="font-size: 1.2em;">集中荷重の片持梁	No. <b style="font-size: 1.2em;">6
---	---

内容計算式等



E : ヤング率 (kg/mm²)
 I : 断面2次モーメント (mm⁴)
 a : 集中荷重の位置 (mm)
 P : 荷重 (kg)
 x : たわみ位置 (mm)

たわみ y (mm)、たわみ角 s (°)、曲げモーメント M (kg · mm)

① $l > x > a$

$$y = \frac{Pa^3}{6EI} - \frac{Pa^2}{2EI}x$$

$$s = \tan^{-1}\left[-\frac{Pa^2}{2EI}\right]$$

$$M = 0$$

(せん断荷重 $W_s = 0$)

② $x \leq a$

$$y = \frac{P}{6EI}x^3 - \frac{Pa}{2EI}x^2$$

$$s = \tan^{-1}\left[\frac{Px}{2EI}(x-2a)\right]$$

$$M = P(x-a)$$

(せん断荷重 $W_s = P$)

例題

$E = 4000 \text{ kg/mm}^2$
 $I = 5 \text{ mm}^4$
 $a = 30 \text{ mm}$
 $P = 2 \text{ kg}$

で $x=25 \text{ mm}$ さらに、 $x=32 \text{ mm}$ の位置では?

準備および操作 ●右のプログラムを計算機に覚えさせます。

●MODE [1] (RUN 状態)で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	Prog 0 [EXE]	E = ?	11	[EXE]	-10.
2	4000 [EXE]	I = ?	12	[EXE]	X = ?
3	5 [EXE]	A = ?	13	32 [EXE]	Y =
4	30 [EXE]	P = ?	14	[EXE]	-0.99
5	2 [EXE]	X = ?	15	[EXE]	S =
6	25 [EXE]	Y =	16	[EXE]	-2.57657183
7	[EXE]	-0.6770833333	17	[EXE]	M =
8	[EXE]	S =	18	[EXE]	0.
9	[EXE]	-2.505092867	19	以下手順5より繰り返し	
10	[EXE]	M =	20		

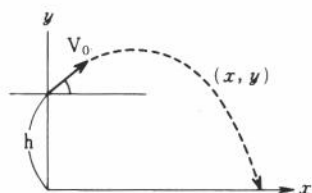
行	(MODE) [2] に続いて下の命令の順にキーを押す	プログラム	実行内容	ステップ
1	Deg :	" E = " ? → E :	" I = " ?	15
2	→ I :	" A = " ? → A :	" P = "	30
3	? → P :			34
4	Lbl 1 :	" X = " ? → X :		45
5	X ≤ A ⇒	Goto: 2 :		52
6	" Y = "	▲ P X A x ² ÷ (2 X E X		67
7	I) X (A ÷ 3 - X)	▲		78
8	" S = "	▲ tan ⁻¹ ((-) P X A x ² ÷ (2		93
9	X E X I))	▲ " M = " ▲ 0 ▲		107
10	Goto: 1 :			110
11	Lbl 2 :			113
12	" Y = "	▲ P X X x ² ÷ (2 X E X		129
13	I) X (X ÷ 3 - A)	▲		139
14	" S = "	▲ tan ⁻¹ (P X X ÷ (2 X E		154
15	X I) X (X - 2 X A))) ▲		167
16	" M = "	▲ P X (X - A) ▲		180
17	Goto: 1			182
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				

メモリー内容	A	a	H	O	V	
	B		I	P	W	
	C		J	Q	X	x
	D		K	R	Y	
	E	E	L	S	Z	
	F		M	T		
	G		N	U		

CASIO PROGRAM SHEET

プログラム名 <b style="font-size: 1.2em;">放 物 線 運 動	No. 7
--	--

内容計算式等



$$x = (V_0 \cos a) \cdot t$$

$$y = (V_0 \sin a) \cdot t - \frac{1}{2}gt^2 + h$$

$g = 9.8 \text{ (m/s}^2\text{)}$

$V_0 \text{ (m/s)}$
 $a \text{ (}^\circ\text{)}$
 $\Delta t \text{ 時間間隔 (sec)}$
 $h \text{ (m)}$

例 題

初速 $V_0 = 130 \text{ (m/sec)}$
 角度 $a = 25^\circ$
 高さ $h = 0 \text{ (m)}$
 $\Delta t = 0.5 \text{ (sec)}$
 の場合の放物線の描く座標を Δt 間隔で求めよ。

準備および操作

- 右のプログラムを計算機に覚えさせます。
- **MODE** **1** (RUN 状態)で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	Prog 0 EXE	$V \theta = ?$	11	EXE	T =
2	130 EXE	A = ?	12	EXE	0.5
3	25 EXE	H = ?	13	EXE	X =
4	0 EXE	T = ?	14	EXE	58.91000616
5	0.5 EXE	T =	15	EXE	Y =
6	EXE	0.	16	EXE	26.24518701
7	EXE	X =	17		
8	EXE	0.	18	以下手順11より繰り返し	
9	EXE	Y =	19		
10	EXE	0.	20		

行	(MODE 2) に続いて下の命令の順にキーを押す	プログラム	実行内容	ステップ
1	Deg: : 0 → S :			6
2	" V 0 = " ? → V : " A = " ? →			21
3	A : " H = " ? → H : " T = " ?			36
4	→ T :			39
5	Lbl 1 : V X cos A X S → X : V X sin			54
6	A X S - 9 . 8 X S x^2 ÷ 2 + H →			69
7	Y :			71
8	" T = " ▲ S ▲ S + T → S :			84
9	" X = " ▲ X ▲ " Y = " ▲ Y ▲			98
10	Y ≥ 0 ⇒ Goto: 1			104
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				

メモリー内容	A	a	H	h	O	V	V_0
	B		I		P	W	
	C		J		Q	X	
	D		K		R	Y	
	E		L		S	Z	
	F		M		T	Δt	
	G		N		U		

CASIO PROGRAM SHEET

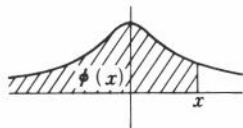
プログラム名 <h2 style="text-align: center;">正 規 分 布</h2>	No. 8
--	-----------------

内容計算式等

正規分布関数 $\phi(x)$ を求める (Hastings の最良近似式より)

$$\phi(x) = \int_{-x}^x \phi(t) dt$$

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$



$$t = \frac{1}{1+Px} \text{ とすると}$$

$$\phi(x) \doteq 1 - \phi(t)(c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5)$$

$$P = 0.2316419$$

$$C_3 = 1.78147937$$

$$C_1 = 0.31938153$$

$$C_4 = -1.821255978$$

$$C_2 = -0.356563782$$

$$C_5 = 1.330274429$$

例 題

$x = 1.18$ 、 $x = 0.7$ のとき $\phi(x)$ は?

- 準備および操作**
- 右のプログラムを計算機に覚えさせます。
 - **[MODE]** **[1]** (RUN 状態) で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	[Prog] 0 [EXE]	X = ?	11		
2	1.18 [EXE]	PX =	12		
3	[EXE]	0.880999696	13		
4	[Prog] 0 [EXE]	X = ?	14		
5	0.7 [EXE]	PX =	15		
6	[EXE]	0.7580361367	16		
7			17		
8			18		
9			19		
10			20		

行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ				
1	" X = " ? → X :			8				
2	1 ÷ (1 + 0 . 2 3 1 6 4 1 9 ×			23				
3	X) → T : 1 ÷ √ (2 × π) × e ^r			38				
4	((-) X x ² ÷ 2) → Q :			48				
5	" P X = " ▲ 1 - Q × (0 . 3 1			63				
6	9 3 8 1 5 3 × T + (-) 0 . 3 5 6			78				
7	5 6 3 7 8 2 × T x ² + 1 . 7 8 1			93				
8	4 7 9 3 7 × T x ^r 3 + (-) 1 . 8 2			108				
9	1 2 5 5 9 7 8 × T x ^r 4 + 1 . 3			123				
10	3 0 2 7 4 4 2 9 × T x ^r 5)			136				
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
メモリー内容	A		H		O		V	
	B		I		P		W	
	C		J		Q	ϕt	X	x
	D		K		R		Y	
	E		L		S		Z	
	F		M		T	t		
	G		N		U			

CASIO PROGRAM SHEET

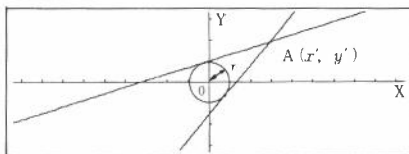
プログラム名

円 と 接 線

No.

9

内容計算式等



円の式

$$x^2 + y^2 = r^2$$

点 A (x' , y') を通る接線の式

$$y - y' = m(x - x')$$

※ m は接線の傾き

半径 r の円に点 A (x' , y') から接線を引き、傾き m と切片 $b(=y' - mx')$ を求める。

また、トレースにより接点の座標を読み、ファクターにより拡大する。

例 題

$$\left. \begin{array}{l} r = 1 \\ x' = 3 \\ y' = 2 \end{array} \right\} \text{より } m \text{ と } b \text{ を求める。}$$

〈注意〉

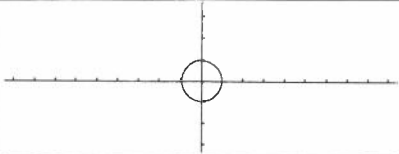
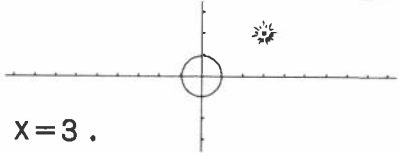
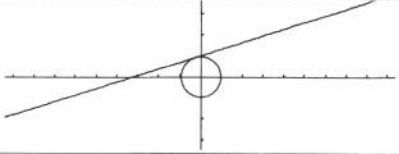
- A 点プロットの移動は不可(移動しても、はじめの点で実行される)
- $r=x'$ の場合は Ma ERROR となります。
- トレース実行を選択し、“TRACE”表示で停止中は、必ずトレースを実行してください。

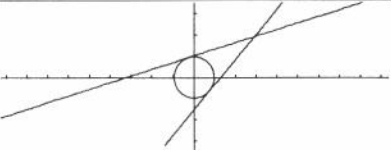
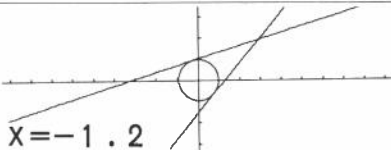
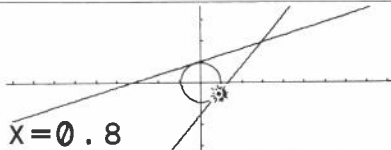
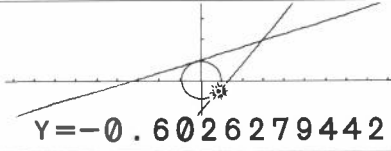
準備および操作 ●右のプログラムを計算機に覚えさせます。

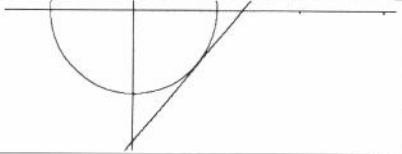
メモリー内容	A	H	O	V
	B	I	P	W
	C	J	Q	X
	D	K	R	Y
	E	L	S	Z
	F	M	T	
	G	N	U	

行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	P0			
2	Prog 1	←		3
3	" X	$x^2 + Y x^2 = R x^2$ ←		13
4	R = "	? → R ←		20
5	Prog 2	▲		23
6	" (X	, Y) ←		30
7	X = "	? → A ←		37
8	" Y = "	? → B ←		45
9	Plot A	, B ▲		50
10	R x^2 (A $x^2 + B x^2 - R x^2$) → P ←		65
11	($\sqrt{}$ P	- A B) (R $x^2 - A x^2$) x^{-1}		80
12	→ M	←		83
13	Lbl 6	←		86
14	Graph M	(X - A) + B ▲		96
15	" M = "	▲ M ▲		103
16	" B = "	▲ B - M A ▲		113
17	Lbl: 0	←		116
18	" T R A C E ?	←		124
19	Y E S ⇒	1 ←		130
20	N O ⇒	0 " : ? → Z ←		140
21	1 → S :	Z = 1 ⇒ Goto 1 ←		151
22	Z = 0 ⇒	Goto 2 : Goto 0 ←		161
23	Lbl 2	←		164
24	((-) A B	- $\sqrt{}$ P) (R $x^2 - A x^2$)		179
25	x^{-1} → N	←		183
26	Graph N	(X - A) + B ▲		193
27	" M = "	▲ N ▲		200
28	" B = "	▲ B - N A ▲		210
29	Lbl 5	←		213
30	" T R A C E ?	←		221
31	Y E S ⇒	1 ←		227
32	N O ⇒	0 " : ? → Z ←		237
33	2 → S :	Z = 1 ⇒ Goto 1 ←		248
34	Z = 0 ⇒	Goto 3 : Goto 5 ←		258
35	Lbl 1	←		261
36	" T R A C E "	▲		269

行	(MODE [2] に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	"	Factor : N : N = " ? → F : Factor : F ←		283
2	Prog : 2 :	S = 1 ⇒ Goto : 9 ←		293
3	S = 2 ⇒	Graph : M (X - A) + B ←		307
4	Graph : N (X - A) + B ▲			317
5	Goto : 3 ←			320
6	Lbl : 9 ←			323
7	Graph : M (X - A) + B ▲			333
8	Prog : 1 :	Prog : 2 : Goto : 6 ←		342
9	Lbl : 3 ←			345
10	"	E N D "		350
11				
12	P1			
13	Range : (-) :	4 . 7 , 4 . 7 , 1 , (-) 3 .		15
14	1 , 3 . 1 , 1			22
15				
16	P2			
17	Graph : √ (: R x ² - X x ²) ←			10
18	Graph : (-) : √ (: R x ² - X x ²)			20
19				
20			計 392	
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

プログラム名 円と接線		No. 9
手順	操作内容	表示
1	Prog 0 EXE	Prog 0 $X^2 + Y^2 = R^2$ R=?
2	1 EXE	
3	EXE	1 done (X, Y) X=?
4	3 EXE 2 EXE	 X=3.
5	EXE	
6	EXE EXE	done M= 0.3169872981 - Disp -
7	EXE EXE	0.3169872981 B= 1.049038106 - Disp -
8	EXE	TRACE? YES⇒1 NO⇒0 ?

手順	操作内容	表示
9	0 <input type="button" value="EXE"/>	
10	<input type="button" value="EXE"/> <input type="button" value="EXE"/>	done M= 1.183012702 - Disp -
11	<input type="button" value="EXE"/> <input type="button" value="EXE"/>	B= 1.183012702 -1.549038106 - Disp -
12	<input type="button" value="EXE"/>	TRACE? YES⇒1 NO⇒0 ?
13	1 <input type="button" value="EXE"/>	? 1 TRACE - Disp -
14	<input type="button" value="SHIFT"/> <input type="button" value="Trace"/>	 X=-1.2
15	<input type="button" value="⇐"/> ~	 X=0.8
16	<input type="button" value="SHIFT"/> <input type="button" value="X→Y"/>	 Y=-0.6026279442

プログラム名		No.
円と接線		9
手順	操作内容	表示
17	EXE	? 1 TRACE Factor N:N=?
18	4 EXE	
19	EXE	Factor N:N=? 4 done END
20		
21		
22		
23		
24		

CASIO PROGRAM SHEET

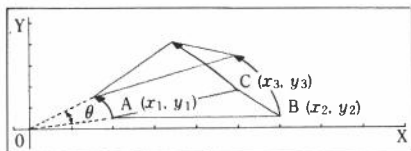
プログラム名

図形の回転

No.

10

内容計算式等



任意の三角形を θ 度回転した図を描く。

座標変換式

$$(x, y) \rightarrow (x', y')$$

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

例題

A(2, 0.5)、B(6, 0.5)、C(5, 1.8)の3点により表わされる三角形を 30° 回転した図形を描く。



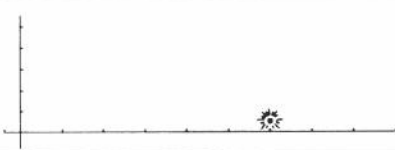
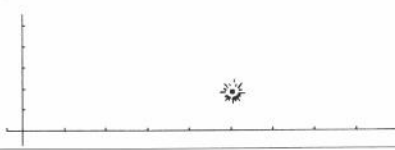
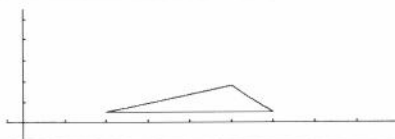
〈注意〉

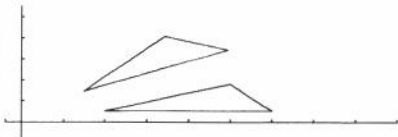
- 入力して点滅している点は、カーソルキーで移動できます。
- プログラムを中止するときは、グラフィック表示中に **[AC]** キーを押します。
- 変換した座標がレンジをこえた場合は、三角形は描きません。

準備および操作 ●右のプログラムを計算機に覚えさせます。

メモリー内容	A	x_1	H	y'_1	O		V
	B	y_1	I	x'_2	P		W
	C	x_2	J	y'_2	Q	θ	X
	D	y_2	K	x'_3	R		Y
	E	x_3	L	y'_3	S		Z
	F	y_3	M		T		
	G	x'_1	N		U		

行	(MODEL 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	Range	(-) 0 . 4 , 9 , 1 , (-) 0 . 8 ,		15
2		5 . 4 , 1 : Deg: ←		23
3	"	(X 1 , Y 1) ←		32
4	X	1 = " ? → A ←		40
5	"	Y 1 = " ? → B ←		49
6	Plot	A , B ▲		54
7	X	→ A : Y → B ←		62
8	"	(X 2 , Y 2) ←		71
9	X	2 = " ? → C ←		79
10	"	Y 2 = " ? → D ←		88
11	Plot	C , D ▲		93
12	X	→ C : Y → B ←		101
13	"	(X 3 , Y 3) ←		110
14	X	3 = " ? → E ←		118
15	"	Y 3 = " ? → F ←		127
16	Plot	E , F ▲		132
17	X	→ E : Y → F ←		140
18	Lbl	1 ←		143
19	Line	: Plot A , B : Line : Plot C , D : Line		158
20	▲			159
21	"	K A K U D O : Deg: " ? → Q ←		173
22	A	cos Q - B sin Q → G ←		183
23	A	sin Q + B cos Q → H ←		193
24	Plot	G , H ←		198
25	C	cos Q - D sin Q → I ←		208
26	C	sin Q + D cos Q → J ←		218
27	Plot	I , J : Line ←		225
28	E	cos Q - F sin Q → K ←		235
29	E	sin Q + F cos Q → L ←		245
30	Plot	K , L : Line ←		252
31	Plot	G , H : Line ▲		259
32	Cls	: Plot C , D : Plot E , F : Goto 1		273
33				
34			計 273	
35				
36				

プログラム名		No.
図形の回転		10
手順	操作内容	表示
1	Prog 0 [EXE]	Prog 0 (X1, Y1) X1=?
2	2 [EXE] 0.5 [EXE]	 X = 2 .
3	[SHIFT] X↔Y [SHIFT] X↔Y (座標表示なし)	
4	[EXE]	0.5 done (X2, Y2) X2=?
5	6 [EXE] 0.5 [EXE]	
6	[EXE]	0.5 done (X3, X3) X3=?
7	5 [EXE] 1.8 [EXE]	
8	[EXE]	

手順	操 作 内 容	表 示
9	EXE	1.8 done done KAKUDO: Deg?
10	30 EXE	
11	以下手順8より繰り返す	
12		
13		
14		
15		
16		

CASIO PROGRAM SHEET

プログラム名

パラメータによるグラフの変化を見る

No.

11

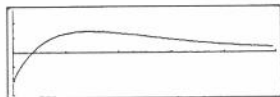
内容計算式等

減衰振動

(i) $\epsilon > n$ (過減衰)

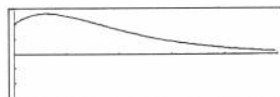
$$P_1 = -\epsilon + \sqrt{\epsilon^2 - n^2}, \quad P_2 = -\epsilon - \sqrt{\epsilon^2 - n^2}$$

$$x = \frac{v_0 - x_0 P_2}{P_1 - P_2} e^{P_1 t} - \frac{v_0 - x_0 P_1}{P_1 - P_2} e^{P_2 t}$$



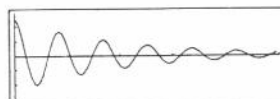
(ii) $\epsilon = n$ (臨界減衰)

$$x = \{x_0 + (v_0 + \epsilon x_0)t\} e^{-\epsilon t}$$



(iii) $\epsilon < n$ (減衰振動)

$$x = e^{-\epsilon t} \left\{ x_0 \cos \sqrt{n^2 - \epsilon^2} t + \frac{v_0 + \epsilon x_0}{\sqrt{n^2 - \epsilon^2}} \sin \sqrt{n^2 - \epsilon^2} t \right\}$$



例題


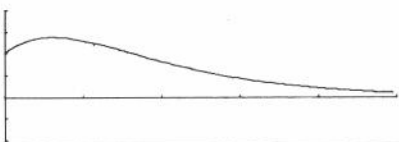
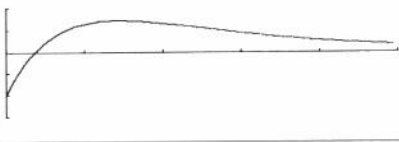
下記のパラメータを持つ減衰振動のグラフを描く。

- | | | |
|----------------------|----------------------|----------------------|
| (1) $\epsilon = 0.1$ | (2) $\epsilon = 0.2$ | (3) $\epsilon = 0.2$ |
| $n = 1.5$ | $n = 0.2$ | $n = 0.18$ |
| $x_0 = 2.5$ | $x_0 = 2$ | $x_0 = -2$ |
| $v_0 = 1$ | $v_0 = 0.6$ | $v_0 = 1.5$ |

準備および操作 ●右のプログラムを計算機に覚えさせます。

メモリー内容	A	x_0	H		O		V
	B	v_0	I		P	$P_1 = -\epsilon + \sqrt{\epsilon^2 - n^2}$	W
	C	$\sqrt{n^2 - \epsilon^2}$	J		Q	$P_2 = -\epsilon - \sqrt{\epsilon^2 - n^2}$	X
	D		K		R		Y
	E	ϵ	L		S		Z
	F		M		T		
	G		N	n	U		

行	(MODE 2 に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	Rad: ←			2
2	Range: 0 , 2 5 , 5 , (-) 3 , 3 , 1 ←			17
3	" E P S I L O N = " ? → E ←			31
4	" N = " ? → N ←			39
5	" X 0 = " ? → A ←			48
6	" V 0 = " ? → B ←			57
7	E > N ⇒ Goto 1 ←			64
8	E = N ⇒ Goto 2 ←			71
9	$\sqrt{\quad}$ (N x^2 - E x^2) → C ←			82
10	Graph: e^x ((-) : E X) (A cos (C X)) +			97
11	(B + E A) C x^{-1} sin (C X)) ←			112
12	Goto: 0 ←			115
13	Lbl: 1 ←			118
14	(-) E + $\sqrt{\quad}$ (E x^2 - N x^2) → P ←			132
15	(-) E - $\sqrt{\quad}$ (E x^2 - N x^2) → Q ←			146
16	Graph: (B - A Q) (P - Q) x^{-1} e^x (161
17	P X) - (B - A P) (P - Q)			176
18	x^{-1} e^x (Q X) ←			183
19	Goto: 0 ←			186
20	Lbl: 2 ←			189
21	Graph: (A + (B + E A) X) e^x ((-)			204
22	E X) ←			208
23	Lbl: 0			210
24				
25			計 210	
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

プログラム名 パラメータによるグラフの変化を見る		No. 11
手順	操作内容	表示
1	Prog 0 [EXE] 0.1 [EXE] 1.5 [EXE] 2.5 [EXE]	1.5 $X0 = ?$ 2.5 $V0 = ?$
2	1 [EXE]	
3	Prog 0 [EXE] 0.2 [EXE] 0.2 [EXE] 2 [EXE]	0.2 $X0 = ?$ 2 $V0 = ?$
4	0.6 [EXE]	
5	Prog 0 [EXE] 0.2 [EXE] 0.18 [EXE] (-) 2 [EXE]	0.18 $X0 = ?$ -2 $V0 = ?$
6	1.5 [EXE]	
7		
8		

手順	操 作 内 容	表 示
9		
10		
11		
12		
13		
14		
15		
16		

CASIO PROGRAM SHEET

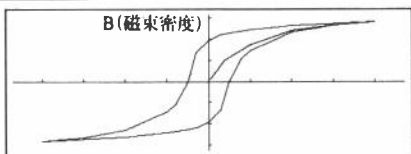
プログラム名

ヒステリシスループ

No.

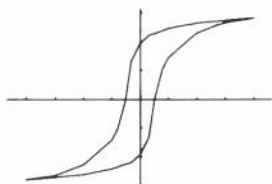
12

内容計算式等

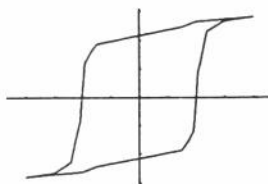


強磁性体試料を磁界中に保持すると、試料は磁化され、 $B-H$ の関係はヒステリシス曲線として表わすことができる。

軟磁性体



硬磁性体



例題

軟磁性体のヒステリシス曲線の作成

	1	2	3	4	5	6	7	8	9
H	0.4	1.0	2.0	3.0	4.0	2.0	1.0	0.5	0.3
B	0.5	0.86	1.2	1.32	1.4	1.31	1.22	1.13	1.1

- データ数：17
- メインループデータ数：12



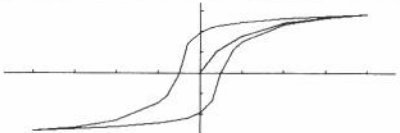
	10	11	12	13	14	15	16	17
H	0	-0.3	-0.5	-0.8	-1.0	-2.0	-3.0	-4.0
B	0.96	0.66	0	-0.53	-0.72	-1.15	-1.33	-1.4

*データ数は20個以内。

準備および操作 ●右のプログラムを計算機に覚えさせます。

メモリー内容	A	データ数	H		O		V
	B	メインループデータ数	I		P		W
	C		J		Q		X
	D		K		R		Y
	E		L		S		Z
	F		M		T		Z[1]~Z[20]B
	G	F[1]~F[20]H	N		U		

行	(MODE [2] に続いて下の命令の順にキーを押す)	プログラム	実行内容	ステップ
1	Range: (-)	4 . 7 , 4 . 7 , 1 , (-) 1 .		15
2	5 5 , 1 . 5 5 , 0 . 5 ←			27
3	Defm: 2 0 ←			31
4	" D A T A SPACE S U U " ? → A ←			45
5	Lbl 9 ←			48
6	" M A I N SPACE L O O P ←			59
7	SPACE: SPACE: SPACE: D A T A SPACE: S U U " ? → B			74
8	←			75
9	B > 2 0 ⇒ Goto 9 ←			83
10	1 → C : Plot 0 , 0 ←			92
11	Lbl 0 : " H = " ? → F [C] ←			106
12	" B = " ? → Z [C] ←			117
13	Plot: F [C] , Z [C] : Line ▲			130
14	C + 1 → C ←			136
15	C ≠ A + 1 ⇒ Goto 0 ←			145
16	A - B + 1 → D ←			153
17	Lbl 1 : Plot: (-) F [D] , (-) Z [D]			168
18	: Line ←			171
19	D + 1 → D ←			177
20	D ≠ A + 1 ⇒ Goto 1 ←			186
21	" E N D "			191
22				
23				
24			メモリ-20×8=160	
25				
26			計 351	
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

プログラム名	ヒステリシスループ	No. 12
手順	操 作 内 容	表 示
1	Prog 0 [EXE]	Prog 0 DATA SUU?
2	17 [EXE]	DATA SUU? 17 MAIN LOOP DATA SUU?
3	12 [EXE]	MAIN LOOP DATA SUU? 12 H=?
4	0.4 [EXE] 0.5 [EXE]	
5	[EXE] 1.0 [EXE] 0.86 [EXE]	
6	以下、データを順に入力する。 ⋮	
7	[EXE]	B=? -1.4 done END
8	G→T	

手順	操 作 内 容	表 示
9		
10		
11		
12		
13		
14		
15		
16		

CASIO PROGRAM SHEET

プログラム名	回 帰 曲 線	No. 13
--------	----------------	---

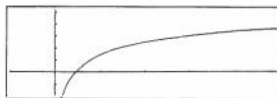
内容計算式等

i 対数回帰曲線

回帰式 $y = A + B \ln x$

$$B = \frac{n \cdot \sum(y \cdot \ln x) - \sum \ln x \cdot \sum y}{n \sum (\ln x)^2 - (\sum \ln x)^2}$$

$$A = \frac{\sum y - B \cdot \sum \ln x}{n}$$

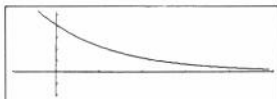


ii 指数回帰曲線

回帰式 $y = A \cdot e^{Bx}$

$$B = \frac{n \sum (x \ln y) - \sum x \cdot \sum \ln y}{n \cdot \sum x^2 - (\sum x)^2}$$

$$A = e^{\left(\frac{\sum \ln y - B \cdot \sum x}{n} \right)}$$

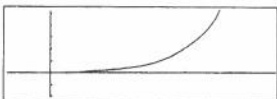


iii べき乗回帰曲線

回帰式 $y = A \cdot x^B$

$$B = \frac{n \sum (\ln x \cdot \ln y) - \sum \ln x \cdot \sum \ln y}{n \cdot \sum (\ln x)^2 - (\sum \ln x)^2}$$

$$A = \frac{\sum \ln y - B \cdot \sum \ln x}{n}$$



※例題は172ページ

準備および操作 ●右のプログラムを計算機に覚えさせます。

メモリー内容	A	A or ln A	H	$\sum (\ln x)^2$	O		V	$\sum x$
	B	B	I		P	$\sum y^2$	W	n
	C	$\sum \ln x$	J		Q	$\sum y$	X	x データ
	D	$\sum \ln y$	K		R	$\sum xy$	Y	y データ
	E	$X \sum \ln y$	L		S	1 ~ 3 選択用	Z	
	F	$Y \sum \ln x$	M		T			
	G	$\sum (\ln x \cdot \ln y)$	N		U	$\sum x^2$		

行	(MODE) [2] に続いて下の命令の順にキーを押す	プログラム	実行内容	ステップ
1	P0 [SHIFT] [MODE] [⇩] →	LR 2		
2	Sci : Cls : 0 →	C ~ H ←		10
3	" Range : O K ? "	▲		17
4	" D : A T A SPACE	I N ~ E N D → ←		31
5	A C → Prog 1	SPACE E X E " ←		42
6	Lbl 1 ←			45
7	" X : " ? →	X ←		53
8	" Y : " ? →	Y ←		61
9	In X + C →	C : In Y + D →	D : X	76
10	In Y + E →	E : Y In X + F →	F :	91
11	In X X In Y +	G → G : (In X)	x^2	106
12	+ H →	H ←		111
13	X , Y DT	▲		116
14	Goto 1			118
15				
16	P1 [MODE] [⇩] →	COMP:		
17	" Y = A + B In X	SPACE → 1 ←		12
18	Y = A X e' (B X)	SPACE → 2 ←		25
19	Y = A X X x' B	SPACE SPACE → 3 ←		37
20	1 ~ 3 : " ? →	S ←		46
21	S = 1 ⇒	Prog 7 ←		53
22	S = 2 ⇒	Prog 8 ←		60
23	S = 3 ⇒	Prog 9 ←		67
24	" E N D "			72
25				
26	P7 [SHIFT] [MODE] [⇩] →	LR 2		
27	(W F - C Q) (W H - C	x^2) x^{-1}		15
28	→ B : (Q - B C) W	x^{-1} → A ←		29
29	Graph A + B In X	▲		36
30	" A : " ▲	A ▲		43
31	" B : " ▲	B ▲		50
32				
33				
34				
35				
36				

CASIO PROGRAM SHEET

プログラム名

回 帰 曲 線

No.

13

例 題

次のデータを指数回帰する。

x_i	2.2	5.6	9.5	13.8	18.0	23.2	29.9	37.8
y_i	35.6	28.1	23.0	17.9	12.9	10.2	6.2	4.0

指数回帰曲線を描き、トレースにより、 $X=20$ のときの Y の値を推定し、回帰式の A および B の値を求める。

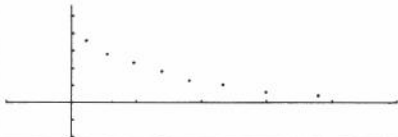
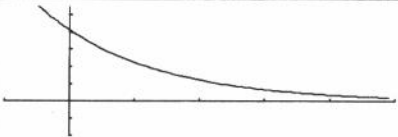
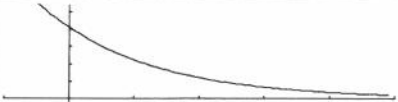
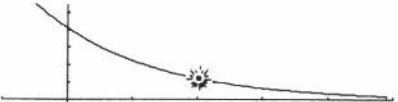
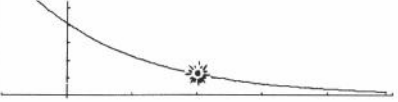
レンジ内容

$X \text{ min} : -10$	$Y \text{ min} : -20$
$X \text{ max} : 50$	$Y \text{ max} : 55$
$X \text{ scl} : 10$	$Y \text{ scl} : 10$

メモリー内容	A	H	O	V
	B	I	P	W
	C	J	Q	X
	D	K	R	Y
	E	L	S	Z
	F	M	T	
	G	N	U	

行	(MODE 2) に続いて下の命令の順にキーを押す	プログラム	実行内容	ステップ
1	P8	SHIFT MODE + → LR 2		
2	(W E - V D) (W U - V x ²) x ⁻¹		15
3	→	B : (D - B V) W x ⁻¹ → A ←		29
4	Graph	e ^r A X e ^r B X ▲		37
5	"	A : " ▲ e ^r A ▲		45
6	"	B : " ▲ B ▲		52
7				
8	P9	SHIFT MODE + → LR 2		
9	(W G - C D) (W H - C x ²) x ⁻¹		15
10	→	B : (D - B C) W x ⁻¹ → A ←		29
11	Graph	e ^r A X X x ^r B ▲		37
12	"	A : " ▲ e ^r A ▲		45
13	"	B : " ▲ B ▲		52
14				
15			計 344	
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

プログラム名		No.
回 帰 曲 線		13
手順	操 作 内 容	表 示
1	Prog 0 EXE (レンジ設定チェック)	Prog 0 Range OK? - Disp -
2	Range (ここではレンジを設定します) (-) 10 EXE 50 EXE 10 EXE	Range Xmin : -10 max : 50 scl : 10
3	(-) 20 EXE 55 EXE 10	Y Range min : -20 max : 55 scl : 10
4	EXE EXE (データ入力終了後は AC キーを押し、 Prog 1 のプログラムを実行する)	Range OK? DATA IN ~END→ AC → Prog 1 EXE X : ?
5	2.2 EXE 35.6 EXE	Y : ? 35.6 2.2 - Disp -
6	EXE	Y : ? 35.6 2.2 X : ?
7	以下、データを順に入力する ⋮	
8	4.0 EXE	Y : ? 4.0 37.8 - Disp -

手順	操作内容	表示
9	$\boxed{G \rightarrow T}$	
10	\boxed{AC} $\boxed{Prog} 1 \boxed{EXE}$	$Y=A+B \ln X \rightarrow 1$ $Y=A X e^{(B X)} \rightarrow 2$ $Y=A X X x^y B \rightarrow 3$ $1 \sim 3 : ?$
11	2 \boxed{EXE} (指数回帰を選択)	
12	$\boxed{SHIFT} \boxed{Trace}$	 $X = -4.893617021$
13	$\boxed{\Rightarrow} \sim$ ($X = 20$ となるまで、ポインタを移動させます)	 $X = 20.$
14	$\boxed{SHIFT} \boxed{X \rightarrow Y}$	 $Y = 11.86149086$
15	\boxed{EXE} \boxed{EXE}	<p style="text-align: right;">done</p> A : 40.68214077 - Disp -
16	\boxed{EXE} \boxed{EXE}	B : 40.68214077 -0.06162460519 - Disp -

プログラム名		No.
回 帰 曲 線		13
手順	操 作 内 容	表 示
17	EXE	40.68214077 B : -0.06162460519 END
18		
19		
20		
21		
22		
23		
24		

手順	操作内容	表示
25		
26		
27		
28		
29		
30		
31		
32		

CASIO PROGRAM SHEET

プログラム名 <div style="text-align: center; font-size: 1.2em; font-weight: bold;">パレード図</div>	No. <div style="font-size: 1.2em; font-weight: bold;">14</div>
---	---

内容計算式等

パレード図とは、たとえば、QC活動でとり上げる問題で、その問題の悪さ加減を事実データにもとづいて、定量的に分析し、手を打たなければならない重点項目を決めることをねらいとして用いたりします。

ヨコ軸：問題の分類

(この例では6項目)

タテ軸：(左側)各分類における

悪さの加減

(右側)占有率



分類(不良場所)

例題

右のデータにより、パレード図を作成する。

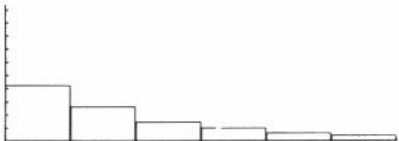
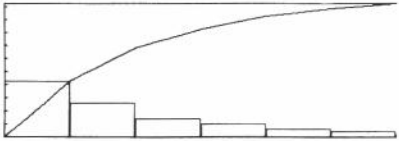
不良場所	件数
A	105
B	65
C	35
D	25
E	15
その他	10

準備および操作 ●右のプログラムを計算機に覚えさせます。

●**MODE** **1** (RUN 状態)で下のキー操作の順にキーを押します。

メモリー内容	A	入力データ	H		O		V	
	B		I		P		W	n
	C		J		Q		X	データ入力数カウント
	D		K		R		Y	
	E		L		S	表示時のカウント	Z	データの和
	F		M		T			Z[1]~Z[6]
	G		N		U			

行	(MODE 2)に続いて下の命令の順にキーを押す	プログラム	実行内容	ステップ
1	P0	SHIFT:MODE: [X] → SD2		
2	Sci	: Mcl : Defm: 6 ←		7
3	Range	: 0 , 6 , 1 , 0 , 5 0 0 , 5 0		22
4	←			23
5	Lbl	1 ←		26
6	"	D A T A " ? → A ←		36
7	X	: A DT ←		41
8	X	+ 1 → X : X ≤ 5 ⇒ Goto 1 ←		54
9	Range	: , , , W , W ÷ 1 0 ←		66
10	Graph	▲		68
11	Plot	0 , 0 ←		73
12	1	→ S ←		77
13	Lbl	2 ←		80
14	Z	[S] + Z → Z ←		89
15	Plot	S , Z : Line ←		96
16	S	+ 1 → S : S ≤ 6 ⇒ Goto 2 ←		109
17	Graph	W		111
18				
19			メモリー 6 × 8 = 48	
20				
21			計 159	
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

プログラム名		No.
パレード図		14
手順	操作内容	表示
1	Prog 0 [EXE]	Prog 0 DATA?
2	105 [EXE]	Prog 0 DATA? 105 DATA?
3	65 [EXE]	105 DATA? 65 DATA?
4	以下、データを順に入力します ⋮	
5	10 [EXE] (棒グラフの表示)	
6	[EXE] (パレード図の表示)	
7		
8		

手順	操 作 内 容	表 示
9		
10		
11		
12		
13		
14		
15		
16		

CASIO PROGRAM SHEET

プログラム名	No.
--------	-----

内容計算式等

例題

- 準備および操作
- 右のプログラムを計算機に覚えさせます。
 - MODE 1 (RUN 状態)で下のキー操作の順にキーを押します。

手順	キ ー 操 作	表 示	手順	キ ー 操 作	表 示
1	<input type="checkbox"/>		11	<input type="checkbox"/>	
2	<input type="checkbox"/>		12	<input type="checkbox"/>	
3	<input type="checkbox"/>		13	<input type="checkbox"/>	
4	<input type="checkbox"/>		14	<input type="checkbox"/>	
5	<input type="checkbox"/>		15	<input type="checkbox"/>	
6	<input type="checkbox"/>		16	<input type="checkbox"/>	
7	<input type="checkbox"/>		17	<input type="checkbox"/>	
8	<input type="checkbox"/>		18	<input type="checkbox"/>	
9	<input type="checkbox"/>		19	<input type="checkbox"/>	
10	<input type="checkbox"/>		20	<input type="checkbox"/>	

No.

行	(MODE 2) に続いて下の 命令の順にキーを押す				プログラム	実行内容	ステップ
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
メモ リ ー 内 容	A		H		O		V
	B		I		P		W
	C		J		Q		X
	D		K		R		Y
	E		L		S		Z
	F		M		T		
	G		N		U		

CASIO PROGRAM SHEET

プログラム名	No.
--------	-----

例 題

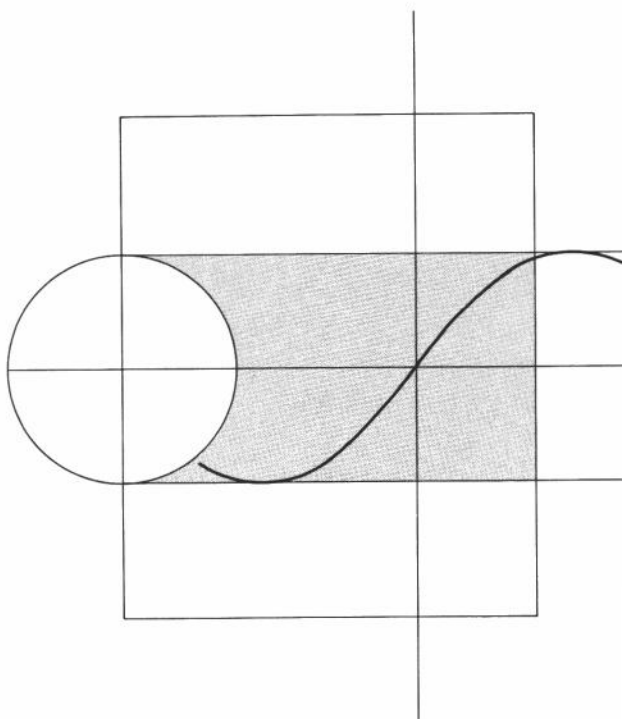
メモリー内容	A	H	O	V
	B	I	P	W
	C	J	Q	X
	D	K	R	Y
	E	L	S	Z
	F	M	T	
	G	N	U	

No.

行	(MODEI ② に続いて下の 命令の順にキーを押す)	プ	ロ	グ	ラ	ム	実行内容	ステップ
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								

プログラム名		No.
手順	操 作 内 容	表 示
1		
2		
3		
4		
5		
6		
7		
8		

付 録



■マニュアル計算

モードの確認	Comp モード (MODE) (田)	四則計算および一般関数計算を行なうモードです。
	Base-n モード (MODE) (口)	2進・8進・10進・16進数の変換および計算、論理演算を行なうモードです。
	SD1 モード (MODE) (x)	標準偏差計算(1変数統計計算)を行なうモードです。
	LR1 モード (MODE) (田)	回帰計算(2変数統計計算)を行なうモードです。
	SD2 モード (SHIFT) (MODE) (x)	1変数統計グラフを描くモードです。 (棒グラフ、折れ線グラフ、正規分布曲線)
	LR2 モード (SHIFT) (MODE) (田)	2変数統計グラフを描くモードです。 (回帰直線)
一般関数	前置関数	数値の直前に関数命令を入力します。 $\left(\begin{array}{l} \sin, \cos, \tan, \sin^{-1}, \cos^{-1}, \tan^{-1}, \sinh, \\ \cosh, \tanh, \sinh^{-1}, \cosh^{-1}, \tanh^{-1}, \log, \\ \ln, e^x, 10^x, \sqrt{\quad}, \sqrt[3]{\quad}, \text{Abs}, \text{Int}, \text{Frac}, \end{array} \right)$ ※sinh, cosh, tanh, sinh ⁻¹ , cosh ⁻¹ , tanh ⁻¹ は fx-6500G のみです。
	後置関数	数値の直後に関数命令を入力します。 $[x^2, x^{-1}, x!]$
	2変数関数	2つの数値の間に関数命令を入力するものと、関数命令の直後に数値を()で囲んで入力するものがあります。 $\left(\begin{array}{l} A x^B (A \text{ の } B \text{ 乗}) \text{ や } B \sqrt[A]{\quad} (A \text{ の } \frac{1}{B} \text{ 乗}), \\ \text{Pol}(A, B), \text{Rec}(A, B) \\ \text{※} A, B \text{ は数値} \end{array} \right)$
	即時実行関数	キー入力と同時に表示数値を変換します。 ENG、 $\overleftarrow{\text{ENG}}$ 、 $\overrightarrow{\text{ENG}}$
2進・8進・10進・16進計算	基本進数の設定	10進数……(Dec) (EXE) ((Dec) = $\sqrt{\quad}$) 16進数……(Hex) (EXE) ((Hex) = x^2) 2進数……(Bin) (EXE) ((Bin) = log) 8進数……(Oct) (EXE) ((Oct) = ln)

	進数指定	基本進数にかかわらず、直後の数値の進数を指定します。 10進指定…… $\text{SHIFT} \text{ [d]} (\text{[d]} = \sqrt{\quad})$ 16進指定…… $\text{SHIFT} \text{ [h]} (\text{[h]} = x^2)$ 2進指定…… $\text{SHIFT} \text{ [b]} (\text{[b]} = \log)$ 8進指定…… $\text{SHIFT} \text{ [o]} (\text{[o]} = \ln)$
	論理演算	数値を2進数に変換し、ビットごとに計算します。 N o t ……ビットごとの反転 A n d ……ビットごとの論理積 o r ……ビットごとの論理和 x o r ……ビットごとの排他的論理和
標準偏差計算	データのクリアー	$\text{SHIFT} \text{ [Sci]} \text{ [EXE]} (\text{[Sci]} = \text{[AC]})$
	データの入力	データ[:度数] $\text{[DT]} (\text{[DT]} = x^i)$ ※度数は省略可
	データの削除	データ[:度数] $\text{[CL]} (\text{[CL]} = x^j)$ ※度数は省略可
	結果の表示	データ数 n …………… $\text{[ALPHA]} \text{ [n]} \text{ [EXE]} (\text{[n]} = \text{[3]})$ 総和 Σx …………… $\text{[ALPHA]} \text{ [Σx]} \text{ [EXE]} (\text{[Σx]} = \text{[2]})$ 2乗和 Σx^2 …………… $\text{[ALPHA]} \text{ [Σx^2]} \text{ [EXE]} (\text{[Σx^2]} = \text{[1]})$ 平均 \bar{x} …………… $\text{[SHIFT]} \text{ [x̄]} \text{ [EXE]} (\text{[x̄]} = \text{[1]})$ 母標準偏差 $x\sigma n$ …… $\text{[SHIFT]} \text{ [xσn]} \text{ [EXE]} (\text{[xσn]} = \text{[2]})$ 標本標準偏差 $x\sigma n-1$ …………… $\text{[SHIFT]} \text{ [xσn-1]} \text{ [EXE]} (\text{[xσn-1]} = \text{[3]})$
回帰計算	データのクリアー	$\text{SHIFT} \text{ [Sci]} \text{ [EXE]} (\text{[Sci]} = \text{[AC]})$
	データの入力	x データ, y データ[:度数] $\text{[DT]} (\text{[DT]} = x^i)$ ※度数は省略可
	データの削除	x データ, y データ[:度数] $\text{[CL]} (\text{[CL]} = x^j)$ ※度数は省略可
	結果の表示	データ数 n …………… $\text{[ALPHA]} \text{ [n]} \text{ [EXE]} (\text{[n]} = \text{[3]})$ x の総和 Σx …………… $\text{[ALPHA]} \text{ [Σx]} \text{ [EXE]} (\text{[Σx]} = \text{[2]})$ y の総和 Σy …………… $\text{[ALPHA]} \text{ [Σy]} \text{ [EXE]} (\text{[Σy]} = \text{[5]})$

		x の 2 乗和 $\sum x^2$ …… ALPHA Σx^2 EXE ($\Sigma x^2 = 11$) y の 2 乗和 $\sum y^2$ …… ALPHA Σy^2 EXE ($\Sigma y^2 = 4$) データの積和 $\sum xy$ …… ALPHA Σxy EXE ($\Sigma xy = 6$) x の平均 \bar{x} …… SHIFT \bar{x} EXE ($\bar{x} = 11$) y の平均 \bar{y} …… SHIFT \bar{y} EXE ($\bar{y} = 4$) x の母標準偏差 $x\sigma n$ SHIFT $x\sigma n$ EXE ($x\sigma n = 2$) y の母標準偏差 $y\sigma n$ SHIFT $y\sigma n$ EXE ($y\sigma n = 5$) x の標本標準偏差 $x\sigma n-1$ …… SHIFT $x\sigma n-1$ EXE ($x\sigma n-1 = 3$) y の標本標準偏差 $y\sigma n-1$ …… SHIFT $y\sigma n-1$ EXE ($y\sigma n-1 = 6$) 回帰式の定数項 A …… SHIFT A EXE (A = 7) 回帰係数 B …… SHIFT B EXE (B = 8) 相関係数 r …… SHIFT r EXE ($r = 9$) x の推定値 \hat{x} …… y データ SHIFT \hat{x} EXE ($\hat{x} = \boxtimes$) y の推定値 \hat{y} …… x データ SHIFT \hat{y} EXE ($\hat{y} = \boxplus$)
特別機能	Ans 機能	マニュアル計算およびプログラム計算において、最後に行なわれた計算の答えを記憶し、Ans キーにより呼び出します。※数値は仮数部10桁です。
	リプレイ機能	<ul style="list-style-type: none"> 演算結果後、\leftarrow キーまたは \rightarrow キーにより計算式を呼び戻します。 エラー発生後に \leftarrow キーまたは \rightarrow キーを押すことにより、エラーが解除され、エラー箇所を表示します。
	マルチステートメント機能	: (コロン) により文や計算式をつなげて実行します。▲によりそれまでの計算結果を表示します。
	メモリー増設	メモリーの数を標準の26個から増やします。増やす個数は1個単位で最大60個(合計86個)までです。メモリー1個につきプログラムエリア8ステップ必要とします。 MODE \square 増設個数 EXE






グラフ機能	レンジ機能	<p>グラフを描く範囲を設定します。</p> <ul style="list-style-type: none"> • Xmax …… x の最大値 • Xmin …… x の最小値 • Xscl …… x 軸上の目盛の間隔 • Ymax …… y の最大値 • Ymin …… y の最小値 • Yscl …… y 軸上の目盛の間隔
	トレース機能	<p>グラフ上をポインター(点滅している点)が移動し、x 座標、y 座標が読み取れます。</p>
	プロット機能	<p>表示画面の任意の座標にポインター(点滅している点)を打ちます。</p>
	ライン機能	<p>プロット機能により打たれた2点間を直線で結びます。</p>
	ファクター機能	<p>ポインター(点滅している点)を中心として、レンジの範囲を拡大・縮小します。</p>

■プログラム計算

プログラムの入力	入力モード	WRT モード(MODE ②)で行なう。
	計算モード	MODE ④、 MODE ⑤、 MODE ⑥、 MODE ⑦により、プログラム計算に合ったモードを指定する。
	プログラムエリア指定	⇐ キーおよび ⇒ キーにより P0～P9 (0～9)のプログラムエリア番号にカーソルを合わせ、 EXE キーを押す。
プログラムの実行	実行モード	RUN モード(MODE ①)で行なう。
	プログラムエリア指定	Prog プログラムエリア番号 EXE で実行開始。 ※プログラムエリア番号は0～9の1文字
プログラムの編集	入力モード	WRT モード(MODE ②)で行なう。
	プログラムエリア指定	⇐ キーおよび ⇒ キーにより P0～P9 (0～9)のプログラムエリア番号にカーソルを合わせ、 EXE と押す。
	編 集	⇐ 、 ⇒ 、 ⇐ 、 ⇒ キーでカーソルを合わせる。 <ul style="list-style-type: none"> 訂正は正しいキーを押す。 削除は DEL キーを押す。 挿入は SHIFT INS (⇐)と押し挿入モードとし、挿入したいキーを押す。
プログラムのクリアー	クリアーモード	PCL モード(MODE ③)で行なう。
	単一プログラムのクリアー	⇐ キーおよび ⇒ キーによりプログラムエリア(_)になっている)にカーソルを合わせ、 AC キーを押す。
	全プログラムのクリアー	SHIFT Mcl (Mcl = DEL)と押す。

<p>プログラム命令</p>	<p>無条件ジャンプ</p>	<p>プログラムの流れを Goto n に対する Lbl n にジャンプします。*nは0~9の1文字</p>
	<p>条件ジャンプ</p>	<p>関係演算子による条件式が正しければ“⇒”以降の文を実行し、正しくなければ⇐、または▲以降にジャンプします。</p> <div style="text-align: center;"> </div> <p>*関係演算子は=、キ、>、<、≥、≤です。</p>
	<p>カウントジャンプ</p>	<p>メモリーの値を1ずつ加算または減算し、0でなければ次の文を実行し、0であれば次の文の後の⇐、または▲以降にジャンプします。</p> <p style="text-align: center;">メモリー内が0でないとき</p> <p>加算 Isz メモリー名:文 { : } 文</p> <div style="text-align: center;"> </div> <p style="text-align: center;">メモリー内が0のとき</p> <p style="text-align: center;">メモリー内が0でないとき</p> <p>減算 Dsz メモリー名:文 { : } 文</p> <div style="text-align: center;"> </div> <p style="text-align: center;">メモリー内が0のとき</p>
	<p>サブルーチン</p>	<p>メインルーチンからのプログラムの流れを、Prog n (nは0~9の1文字)で示されるサブルーチンに移します。サブルーチンでの実行終了後、元のプログラムの Prog n の次の文に戻ります。</p>

■エラーメッセージ一覧表

メッセージ	エラー内容	対策
Syn ERROR	<p>①計算式の書式に誤りがある。</p> <p>②プログラム中に書式上の誤りがある。</p>	<p>①  キーまたは  キーを押してエラー箇所を表示させ、訂正する。</p> <p>②  キーまたは  キーを押してエラー箇所を表示させ、  後 WRT モードで正しく訂正する。</p>
Ma ERROR	<p>①計算の結果が演算範囲をこえている。</p> <p>②関数桁容量の被演算数をこえて計算が行なわれた。</p> <p>③数値的な誤り(0による除算等)が行なわれた。</p>	<p>①②③ 入力した数値を確認して、範囲内になおす。特に、メモリーを使っている場合はメモリー内の数値をチェックして正しくする。</p>
Go ERROR	<p>① Goton に対する Lbln が無い。</p> <p>② Prog n に対する P n のプログラムエリアにプログラムが記憶されていない。</p>	<p>① Goton に対する Lbln を正しく入れるか、不要であれば Goton を削除する。</p> <p>② Prog n に対する P n のプログラムエリアにプログラムを記憶させるか、不要であれば Prog n を削除する。</p>
Ne ERROR	<p>• Prog n によるサブルーチンのネスティングが10段をこえてオーバーしている。</p>	<p>• サブルーチンからの戻りに Prog n を使っていないかをチェックし、不要な Prog n を削除する。</p> <p>• サブルーチンのジャンプ先をたどり、元のプログラムエリアにジャンプしていないかをチェックし、正しく戻るようにする。</p>

Stk ERROR	<ul style="list-style-type: none"> • 数値スタックおよび演算スタックをこえる計算式が実行された。 	<ul style="list-style-type: none"> • 計算式を簡略化して数値スタックは8段、演算スタックは20段以内に納める。 • 計算式を2つ以上に分けてスタック以内に納める。
Mem ERROR	<ul style="list-style-type: none"> • メモリーに関するエラーで、増設していないメモリー(Z [5] 等)を使った。 	<ul style="list-style-type: none"> • MODE □ (Defm)により、正しくメモリー数を増やす。 • メモリーの使い方を現在設定されているメモリー数以内に納める。
Arg ERROR	<p>引数を必要とする命令で、引数の指定が間違っている。</p>	<p>引数を正しく訂正する。</p> <ul style="list-style-type: none"> • Sci n、Fix n の n を 0～9 の自然数とする。 • Goto n、Lbl n、Prog n の n を 0～9 の自然数とする。 • Defm n の n が 0～残りステップ数以内の自然数とする。

■関数桁容量(原則として)

関数名	被演算数
$\sin x, \cos x, \tan x$	$ x \leq 9 \times 10^9$ 度 $ x \leq 5 \times 10^7 \pi$ rad $ x < 10^{10}$ gra
$\sin^{-1} x, \cos^{-1} x$	$ x \leq 1$
$\tan^{-1} x$	$ x < 10^{100}$
e^x	$-10^{100} < x \leq 230.2585092$
$\sinh x, \cosh x$	$ x \leq 230.2585092$
$\tanh x$	$ x < 10^{100}$
$\sinh^{-1} x$	$ x < 5 \times 10^{99}$
$\cosh^{-1} x$	$1 \leq x < 5 \times 10^{99}$
$\tanh^{-1} x$	$ x < 1$
$\log x, \ln x$	$0 < x < 10^{100}$
10^x	$-10^{100} < x < 100$
\sqrt{x}	$0 \leq x < 10^{100}$
x^2	$ x < 10^{50}$
$x^{-1} (\frac{1}{x})$	$ x < 10^{100}, x \neq 0$
$\sqrt[3]{x}$	$ x < 10^{100}$
$x!$	$0 \leq x \leq 69$ (x は整数)
x^y	$x < 0$ のとき、 y は自然数(但し、 $x = 0$ のとき $y > 0$) $x > 0$: $-1 \times 10^{100} < y \log x < 100$
$\sqrt[y]{x} (x^{\frac{1}{y}})$	$x \geq 0, y \neq 0$
$\text{Pol}(x, y)$	$ x < 10^{100}, y < 10^{100}$ 、ただし $\sqrt{x^2 + y^2} < 10^{100}$
$\text{Rec}(r, \theta)$	$ r < 10^{100}$ $ \theta \leq 9 \times 10^9$ 度 $ \theta \leq 5 \times 10^7 \pi$ rad $ \theta < 10^{10}$ gra

2 進 数	(正)1111111111111111 $\cong x \cong 0$ (負)1111111111111111 $\cong x \cong 1000000000000000$
8 進 数	(正)17777777777 $\cong x \cong 0$ (負)3777777777 $\cong x \cong 20000000000$
16 進 数	(正)7 FFFFFFF $\cong x \cong 0$ (負) FFFFFFF $\cong x \cong 80000000$
10進→60進数	$ x \leq 9999999.999$ でかつ度・分・秒の合計が11桁をこえた場合は上位(度・分)を優先して11桁分表示します。
統 計 計 算	$ x < 10^{30}$ 、 $ y < 10^{50}$ 、 $ n < 10^{100}$

※sinh、cosh、tanh、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} はfx-6500Gのみです。

※答えの精度は原則として10桁目±1。

※ x^i 、 $x^{\frac{1}{i}}$ 、 $x!$ 、 $\sqrt[i]{x}$ 、など、内部連続演算をするものは、誤差が累積し、精度が出ないことがあります。

※ $\tan x$ においては $|x| \neq 90^\circ \times (2n+1)$ 、 $|x| \neq \frac{\pi}{2} \text{ rad} \times (2n+1)$ 、 $|x| \neq 100 \text{ gra}(2n+1)$ ； n は整数。

※ $\sinh x$ 、 $\tanh x$ では $x=0$ のとき特異点となり、この近傍では誤差が累積されて、精度が悪くなります。

規 格

型 式：fx-6500G/fx-6000G

演 算 部

基本計算機能：負数・指数・カッコを含む四則計算(加減・乗除の優先順位判別機能つき)

組込関数機能：三角・逆三角関数(角度単位は度・ラジアン・グラッド)、双曲・逆双曲線関数、対数・指数関数、逆数、階乗、開平、立方根、べき乗、べき乗根、2乗、10進 \leftrightarrow 60進、2進・8進・16進計算、座標変換、 π 、乱数、絶対値、整数部除去、小数部除去
※sinh、cosh、tanh、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} はfx-6500Gのみです。

統計計算機能：標準偏差＝データ数、総和、2乗和、平均、標準偏差(2種類)
直線回帰＝データ数、 x の総和、 y の総和、 x の2乗和、 y の2乗和、 x の平均、 y の平均、 x の標準偏差(2種類)、 y の標準偏差(2種類)、定数項、回帰係数、相関係数、 x の推定値、 y の推定値

メモリー：標準26メモリー(最大86メモリー)

計算範囲： $\pm 1 \times 10^{-99}$ ～ $\pm 9.999999999 \times 10^{99}$ 、および0、内部演算は仮数部13桁を使用

小数点方式：浮動方式(工学浮動小数、指定小数、有効桁数表示切替可)

四捨五入：有効桁数指定または小数点以下桁数指定による四捨五入

プログラム部

ステップ数：最大486ステップ(～6ステップ)

ジャンプ機能：無条件ジャンプ(Goto)最大10対

条件ジャンプ(=、 \neq 、 $>$ 、 $<$ 、 \geq 、 \leq)

カウントジャンプ(Isz、Dsz)

サブルーチン(Prog)最大9組、深さ(レベル)10段(入力用バッファを含む)

組込プログラム数：最大10組(P0～P9)

チェック機能：プログラムのチェック、デバック、削除、追加等

グラフィック機能

組込関数グラフ: (20種類) \sin 、 \cos 、 \tan 、 \sin^{-1} 、 \cos^{-1} 、 \tan^{-1} 、 \sinh 、 \cosh 、 \tanh 、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} 、 \log 、 \ln 、 10^x 、 e^x 、 x^2 、 $\sqrt{\quad}$ 、 $\sqrt[3]{\quad}$ 、 x^{-1}

※ \sinh 、 \cosh 、 \tanh 、 \sinh^{-1} 、 \cosh^{-1} 、 \tanh^{-1} は fx-6500G のみです。

グラフ命令: Graph、Range、Plot、Trace、Factor、Line、 $X \leftrightarrow Y$

グラフの種類: 任意の関数グラフ、統計グラフ(棒グラフ、折れ線グラフ、正規分布曲線、回帰直線)

共通部

表示方式および内容: 液晶表示、仮数部10桁、指数部2桁、2進・8進・16進数表示可、60進数表示可

文字表示機能: 関数命令表示、プログラム命令表示、アルファベット表示等を最大64文字表示

エラーチェック機能: 10^{100} 以上および計算不能、ジャンプ等不能をチェック、“ERROR”表示

電源: リチウム電池(CR-2032C) 3個使用

消費電力: 0.03W

電池寿命: CR-2032Cで約130時間

オートパワーオフ: 操作完了後約6分で自動電源オフ

使用温度: 0°C~40°C

大きさ・重さ: <fx-6500G>

幅81奥行148.5厚さ11.5mm、127g (電池込み)

<fx-6000G>

幅81奥行150厚さ21mm、124g (電池込み)

付属品: fx-6500G 手帳型ケース

fx-6000G ソフトケース

カシオ計算機株式会社 本社 営業本部

東京都新宿区西新宿2-6 新宿住友ビル
(〒163) Tel 03-347-4811(代表)

カシオ計算機お問合せ先

旭川	☎0166-23-8580	旭川市七条通り 8 丁目
札幌	☎011-231-2343	札幌市中央区南一条西12丁目
釧路	☎0154-24-8575	釧路市光陽町 6-1
青森	☎0177-22-7466	青森市勝田 2-1-12
秋田	☎0188-33-6211	秋田市中通り 6-1-55
盛岡	☎0196-24-2502	盛岡市本町通り 3-19-6
仙台	☎0222-27-1404	仙台市国分町 2-8-14
山形	☎0236-42-8018	山形市あこや町 3-14-39
郡山	☎0249-33-5172	郡山市香久池 2-11-6
宇都宮	☎0286-34-0395	宇都宮市西大寛 2-1-3
前橋	☎0272-53-3000	前橋市元総社町 92-5
水戸	☎0292-25-6985	水戸市城南 3-10-17
埼玉	☎0486-66-8567	大宮市大成町 4-83
千葉	☎0472-43-1751	千葉市登戸町 2-276
東京	☎03-862-4141	千代田区神田佐久間町 2-23
城西	☎03-341-3321	新宿区新宿 2-4-3
多摩	☎0425-23-3531	立川市錦町 3-2-25
横浜	☎045-211-0821	横浜市中区弁天通り 6-85
新潟	☎0252-87-1155	新潟市弁天橋通 3-9-12
長野	☎0262-28-9360	長野市岡田町 30-20
甲府	☎0552-37-6371	甲府市城東 2-22-11
静岡	☎0542-81-8085	静岡市西中原 1-4-35
浜松	☎0534-64-1658	浜松市西塚町 324
豊橋	☎0532-53-2515	豊橋市魚町 5
名古屋	☎052-263-0454	名古屋市中区栄 4-6-15
岐阜	☎0582-62-0145	岐阜市鷹見町 8
三重	☎0592-27-5066	津市上浜町 1-251
富山	☎0764-22-2251	富山市白銀町 2-1
金沢	☎0762-37-8511	金沢市諸江町下丁 93-1
京都	☎075-351-1161	京都市下京区五条通り堀川東入ル
大阪	☎06-362-8181	大阪市北区南森町 2-1-20
神戸	☎078-392-4123	神戸市中央区伊藤町 119
岡山	☎0862-41-8471	岡山市西古松西町 8-21
広島	☎082-273-7111	広島市西区己斐本町 2-17-24
高松	☎0878-62-5240	高松市亀岡町 9-16
松山	☎0899-45-2234	松山市平和通り 1-1-5
福岡	☎092-411-2684	福岡市博多区博多駅南 1-2-15
長崎	☎0958-61-8084	長崎市宝栄町 2-26
熊本	☎096-367-0650	熊本市健軍 4-1-5
鹿児島	☎0992-56-3575	鹿児島市上荒田町 30-18

カシオ電卓保証書

This Warranty is valid only in Japan.

本書は、下記記載の内容により無料で修理および調整を行なうことをお約束するものです。

★ ご販売店様へ

この保証書はお客様へのアフターサービスの実施と責任を明確にするものです。贈答品、記念品の場合も含めて必ずご記入の上お客様にお渡しください。

機種名：fx-6500G/fx-6000G		保証期間：お買い上げ日より本体1ヵ年	
		お買い上げ日：昭和 年 月 日	
お客様	お名前		様
	ご住所		
	電話		
販売店	住所・店名		

日本橋本店
TEL (06) 643-2038
63.3.
保証書印

保証規定

1. 説明書の注意に従った正常な使用状態で故障した場合は、お買い上げ後1年間無料で修理いたします。
 2. 修理の必要が生じた場合は、製品に本証を添えて、お買い上げ店または最寄りのカシオ計算機サービスセンターへご持参またはご郵送ください。
 3. 修理ご依頼品のご持参およびお持ち帰りの場合の交通費など、またご郵送される場合の郵送料金および諸掛りはお客様の負担となります。
※ご郵送には適切な梱包の上、紛失などを防ぐため簡易書留をご利用ください。
 4. 保証期間内でも次の場合は有料修理になります。
イ. 誤用・乱用および取り扱い不注意による故障
ロ. 火災・地震・水害および盗難などの災害による故障
ハ. 不当な修理や改造および異常電圧に起因する故障
ニ. 使用中に生じたキズなどの外観上の変化
ホ. 消耗品および付属品の交換
ヘ. 電池の液もれによる故障・損傷
ト. 本証の提示がない場合および必要事項（お買い上げ日、販売店名など）の記入がない場合
 5. 記憶されているプログラムおよびデータの消失は保証対象外です。
 6. 修理内容などの記録は修理伝票にかえさせていただきます。
 7. 本証は日本国内においてのみ有効です。また本証の再発行はいたしませんので大切に保管ください。
- ※この保証書によってお客様の法律上の権利を制限するものではありません。

カシオ計算機株式会社

〒163 東京都新宿区西新宿2-6 新宿住友ビル
☎03-347-4811

fx-6500G/fx-6000G 操作マニュアル

昭和61年2月

初版発行

発行 カシオ計算機株式会社
東京都新宿区西新宿2-6-1
新宿住友ビル内 〒163
電話(03)347-4811(大代表)

印刷 株式会社新晃社
